Automated Visual Inspection/Detection of Railroad Track

Contract No: BD550, RPWO #8

Final Report

July 2010

DR. MUBARAK SHAH

COMPUTER VISION LAB

UNIVERSITY OF CENTRAL FLORIDA

Disclaimer

The opinions, findings, and conclusions expressed in this publication are those of the authors and not necessarily those of the State of Florida Department of Transportation.

Metric Conversion Table

Symbol	Converted From	Multiplier	Converted To	Symbol
in	inches	25.4	millimeters	mm
mm	millimeters	0.039	inches	in
m	meters	3.28	feet	ft
ft	feet	0.305	meters	m
m	meters	1.09	yards	yd
yd	yards	0.914	meters	m
mi	miles	1.61	kilometers	km
km	kilometers	0.621	miles	mi

Technical Report Documentation Page

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.
4. Title and Subtitle Automated Visual Inspection/Detection of Railroad Track		5. Report Date July 2010 6. Performing Organization Code
7. Author(s) Mubarak Shah		8. Performing Organization Report No.
9. Performing Organization Name and Address University of Central Florida 4000 Central Florida Blvd. Harris Engineering Center, Suite 346 Orlando, FL 32816-2362		10. Work Unit No. (TRAIS) 11. Contract or Grant No. BD550-08
12. Sponsoring Agency Name and Address Florida Department of Transportation		13. Type of Report and Period Covered Final Report – Dec '04 – Jul '10 14. Sponsoring Agency Code
15. Supplementary Notes		

16. Abstract

This report describes the technical details behind the implementation of a visual inspection system for railroad tracks for the Florida Department of Transportation (FDOT). It consists of the various algorithms used, and their results in finding defects in tracks including structural and geometric defects. Included is a system on the FDOT's Hi-Rail vehicle deployed for monitoring the gage and notifying the inspector when the gage is not within prescribed limits. This description includes, but is not limited to the hardware interfaces, software Graphical User Interfaces (GUIs) and the actual implementation details of the algorithms. The equipment on-board the Hi-Rail vehicle is currently in service and has been extensively tested over the last 6 months.

17. Key Word		18. Distribution Statement No restrictions.	
19. Security Classif. (of this report) Unclassified.	20. Security Classif. (of this page) Unclassified.	21. No. of Pages 45	22. Price

Form DOT F 1700.7 (8-72) Reproduction of completed page authorized

Executive Summary

Problem Statement

Defects in railroad tracks are responsible for several incidents every year resulting in injuries, fatalities, infrastructure cost, environmental damage, loss of use, etc. As a result, there is a continuous need for inspection and maintenance of these tracks, with human inspectors are performing track inspection. Although their efforts are very thorough, the process can be extremely tedious, demanding, and time-consuming. An automated inspection system to identify defects in railroad tracks may therefore be valuable in assisting the personnel involved during the inspection.

Objectives

The objective of the research is to develop a system that can automatically detect defects in railroad tracks including, but not limited to defects in the track structure and track geometry with notification to the inspector of the type and location of the track defects.

Findings

Researchers performed field-testing of the developed system over several miles (over 1000 GB of data). The system used to find the gage between the rails was extensively tested and found to be useful to the Department's track inspector warning him through audio cues whenever the gage was found to be outside the limits. The algorithms to find defects such as missing bolts, cracks in rails, engine burns, etc. was extensively tested in a lab environment and found to perform satisfactorily while field-level testing has not provided consistent results.

Conclusion

The system currently deployed is being used extensively in the field for gage detection. Researchers have developed the remaining algorithms for bolt detection, missing fasteners, detection of scarring of cross-ties, etc. in a controlled laboratory environment. Both these systems have been tested and found to be working with good results in the lab.

Projected / actual benefits of implementing research

The developed automated inspection system was found to be very useful in assisting the personnel in the inspection process. The system has the capabilities to log the exact location of a defect as well as the gage corresponding to the location accurately over long distances using the encoder-camera-laser interface. This is beneficial since the department can maintain a history of the inspection and use it for comparison as a measure of deterioration of track conditions over time.

Table of Contents

Disclaimer ii

Metric Conversion Table iii

Technical Report Documentation Page iv

Executive Summary v

Table of Contents vii

List of Figures viii

List of Tables ix

List of Schematics ix

List of Illustrations x

1. Introduction 1

2. Hardware Setup and Architecture 3

2.1 SICK LMS-400 Laser 3

2.2 Laser Setup for Gage Detection 3

2.3 Technical Specifications of Visual Sensors and Equipment 5

2.4 Encoder-Camera Interface for Image Acquisition 10

3. Software – Graphical User Interface (GUI) 13

4. Technical Details – Defects Being Inspected and Results 17

4.1 Measuring Gage using the Laser Scanners 17

4.2 Engine Burns 24

4.3 Cracks in Rails 26

4.4 Scarring across Cross-Ties 28

4.5 Missing Rail-Fasteners 29

4.6 Re-laying of Tracks 32

5. Conclusion 36

References 37

APPENDIX A – Additional System Modules and Features 38

Mapping Interface 38

Logging 39

Image Viewer Development 40

APPENDIX B - Mobile Simulation Platforms 42

APPENDIX C - Computer Vision Terminology 43

List of Figures

Figure 1: SICK LMS 400 Schematic 3

Figure 2: Schematic of the platform and photographs of the rail-gage measurement system. 4

Figure 3: Imagery acquired from the Piranha 2 Line Scan cameras. 8

Figure 4: Graphical Illustration of the proposed setup for image acquisition. 10

Figure 5: Interface Circuitry for Triggering. 12

Figure 6: Screenshot of Application Software for measuring gage. 14

Figure 7: Partial Screen-shot of the back-end implementation of the GUI in LabView. 15

Figure 8: Merged profile from both laser scanners. 18

Figure 9: Sample 3D Profile obtained from Laser Scanners. 19

Figure 10: Algorithm depicting potential edges found on the rail in an image, before filtering. 20

Figure 11: Measuring the Rail Gage using the laser scanner readings. 20

Figure 12: The profile of a rail gage measured in inches at a track segment of about 8000 feet long. 23

Figure 13: Flow diagram of the algorithmic steps in finding engine burns. 24

Figure 14: The main phases of the detection process. (a) The original panorama frame, (b) The blurred frame region, (c) The mean-shift segmentation results, (d) Detection results. 26

Figure 15: Flow diagram of the algorithmic steps to detect cracks in rails. 27

Figure 16: Stages of the detection process – (a) Original image, (b) Blurred image, (c) Edge detection algorithm, (d) Eliminating regions according to their major axis length, (e) Final result. 27

Figure 17: The defined orientation filters of different sizes used to find the defect. (a) Input image showing scarring of wood. (b) Segmented image detecting scarred region on the cross-tie. (c) Stages in the process. 28

Figure 18: Color based segmentation with K-means clustering. 29

Figure 19: Typical appearance of missing fasteners. 30

Figure 20: (a) Clockwise from the upper left: two steel fasteners, a fast clip, an e-clip on a wooden sleeper, and an e-clip on a concrete sleeper. (b) Confusion Matrix for missing clip detection. 31

Figure 21: Inspection setup for detecting shifted tracks during re-laying. 32

Figure 22: The two sets of images taken using the GigE Setup (Top 2 images – Row 1, Bottom 2 images, - Row 2). 33

Figure 23: Simulated results showing aligned and misaligned images. 34

Figure 24: County (Green), Interstate (Blue), and Rail (Red) for the State of Florida. 38

Figure 25: The LABINS Project and Web Interface through Ka-Maps. 39

Figure 26: Screenshot of Mega-size Image Viewer. 41

Figure 27: The Mobile Simulation Platform for acquiring data. 42

List of Tables

Table 1: Specifications of the Dalsa Piranha 2 Line Scan Camera. 6

Table 2: Specifications of the Prosilica GC1020 Gigabit Ethernet Area Scan Cameras. 7

Table 3: Specifications of the HB6M Hollow Bore Optical Encoder. 8

List of Schematics

Schematic 1: Flow Diagram rail edge detection and gage measurement 20

List of Illustrations

Illustration 1: Clustering 21

1. Introduction

According to Federal Railroad Administration's (FRA) Office of Safety Analysis [1], 9,764 of the 30,195 train accidents in the United States from January 2000 to January 2010 occurred due to defects in the tracks. In Florida, track defects accounted for 36% of the 527 train accidents during the same period. Track defects were responsible for causing 32% of the 3,185 railroad accidents in the United States during 2007 alone. With these figures becoming increasingly alarming, it is of prime importance to efficiently inspect the tracks on a regular basis and prevent untimely accidents that may result in injuries or death. Track defects can be primarily classified into problems that involve:

(a) Track Geometry: This includes deviation from the ideal gage measure, misalignment during re-laying of tracks, deviation of curved/tangent track, etc.

(b) Track Structure: These may include defects (including missing parts) in the cross-ties, tie-plates, fasteners, bolts, rail-joints, etc.

Specialized geometry cars usually inspect track geometry while inspectors visually inspect structural faults, either on foot, or by riding in an inspection vehicle called the Hi-Rail vehicle at a low speed over the tracks. Currently, two inspectors perform track inspection on a regular basis in the state of Florida and attempt to identify these defects. Although they are very thorough in their efforts, the process itself can be time-consuming and demanding. Maintenance of the expanding network of rail tracks requires a considerable investment of time and effort. In the last 7 years, inspectors performed 45,842 nation-wide inspections and 216,100 defects were recorded. At the University of Central Florida (UCF) Computer Vision Lab, researchers have developed an automated railroad track inspection system to assist the inspectors. The development cycle of the system includes:

1. Scientific research to develop algorithms for gage detection, automatic segmentation, feature extraction, object recognition and classification of defects.
2. Development of a user interface which integrates the various algorithms into an easy to use framework.
3. Hardware installation and integration onto a Florida Department of Transportation (FDOT) Hi-Rail vehicle.

4. Laboratory, field testing, and refinement.

This report includes a description of the track inspection system, along with a description of the defects that it can detect.

2. Hardware Setup and Architecture

In the following section, the hardware setup on the FDOT Hi-Rail vehicle is described, which includes an optical encoder, laser scanners and high-speed line-scan cameras.

2.1 SICK LMS-400 Laser

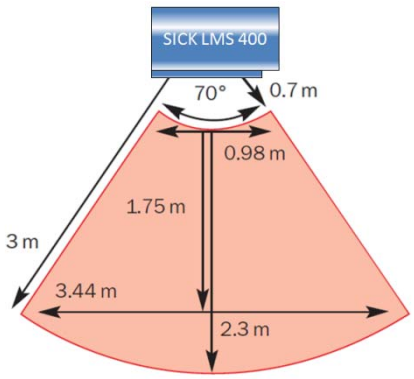


Figure 1: SICK LMS 400 Schematic

Researchers used two SICK LMS 400 Laser Scanners on the Hi-Rail vehicle to detect the gage with high precision. The laser scanners measure a distance to a surface by measuring the time a reflected ray of infrared laser takes to get back to scanner device. Both scanners operate in a 70-degree arc, with an angular precision of 0.1 degrees; each scanner produces 700 measurements per arc. The frequency of measurements is 30Hz (30 arc scans per minute for each scanner). The precision of distance measurements is about +/- 7.5 mm. Both scanners are manually aligned in the horizontal direction with bolts and screws. The distance between scanners is measured and deemed to be invariable due to the rigid setup.

2.2 Laser Setup for Gage Detection

Figure 2 shows a schematic of the system setup and photographs of the laser sensor and Ethernet hub. The features of this system are described below.

1. A server is mounted at the back of the Hi-Rail vehicle to isolate it from engine noise.
2. Researchers at the UCF Computer Vision Lab developed a Graphical User Interface running on a laptop for gage detection and monitoring. The operator monitors the server's operations through this laptop with a Wi-Fi / Local Area Network connection in the front of the Hi-Rail vehicle.
3. An Ethernet hub is placed in the front of the Hi-Railvehicle, in the mounting platform, to receive the Ethernet connections. This unit is connected to the server in the back of the Hi-Rail vehicle through a fiber optic cable.

4. The Encoder signal is converted to the robust Ethernet signal using a converter and the output is connected to the Ethernet hub.

5. Researchers added a set of folding blinds to the box that contains the laser scanner to shield direct sunlight.

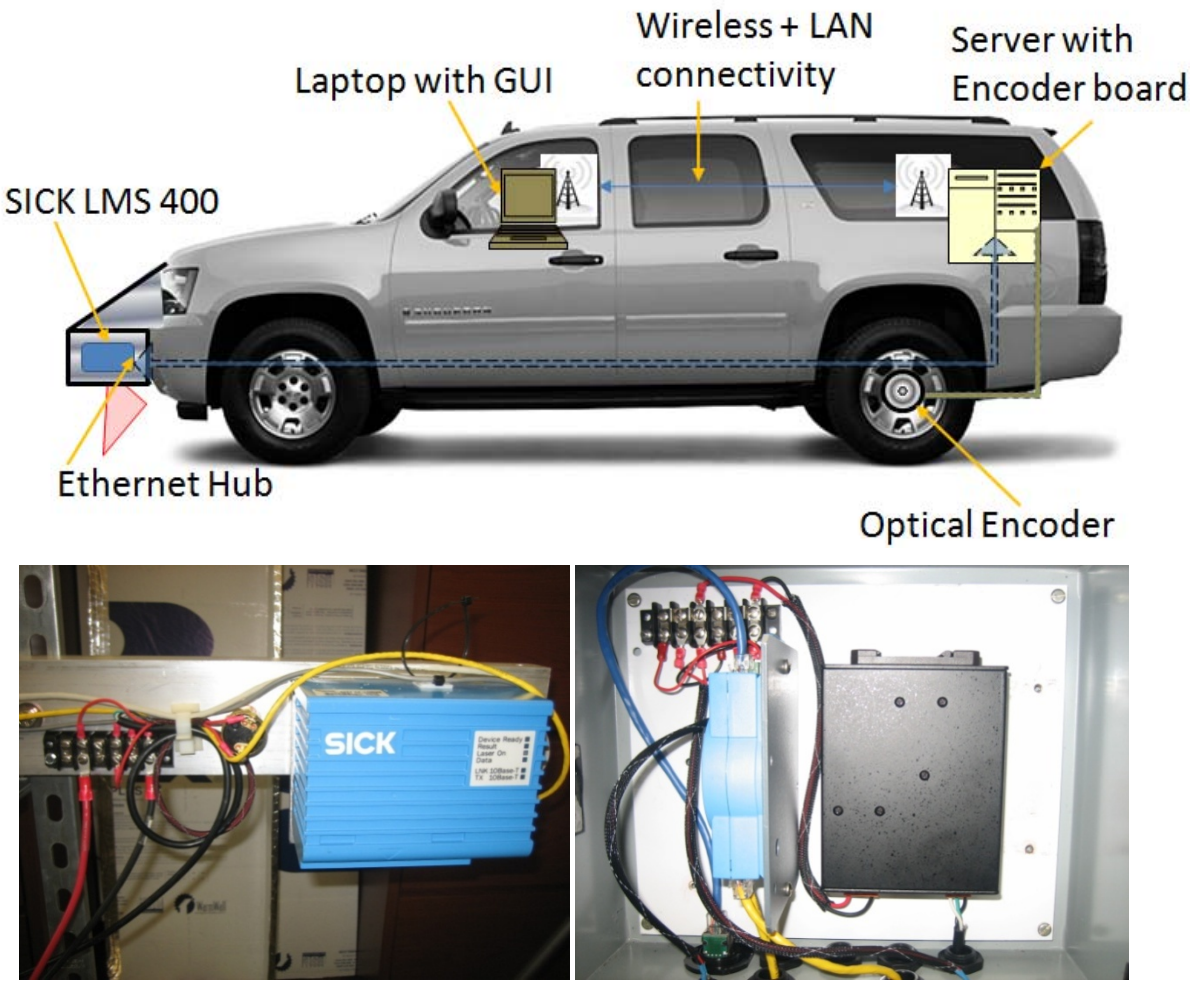


Figure 2: Schematic of the platform and photographs of the rail-gage measurement system.

2.3 Technical Specifications of Visual Sensors and Equipment

The following section details the researchers' use of high-speed cameras and encoder as a part of the image acquisition system. Following this, is an overview of the architecture interfacing the two components.

Image Acquisition: Dalsa Piranha 2 Line Scan Camera

In order to acquire high-resolution imagery, researchers at the UCF Computer Vision Lab decided to use a line scan camera, which captures an image using a single line of sensor pixels. Since a single-line of image is captured every time, a full 2D image is obtained from the relative motion between the object, and the camera itself. In this case, the movement of the Hi-Rail vehicle mounted with the line scan cameras over the railroad tracks will capture the required image. The use of a line scan camera has several benefits over regular area-scan cameras, particularly with respect to railroad inspection.

- Images of fast moving objects are sharp and crisp, and the need for controlling shutter speed and the use of expensive equipment such as strobe lights is eliminated.
- At high speeds, the pixel-fill factor is crucial to both imaging and analysis, and this value is known to reach 100% with line-scan cameras, resulting in superior sensitivity.
- Line scan cameras have a much higher signal-noise ratio when compared to other imaging technology.
- Redundant data can be eliminated using line scan cameras since there is no overlap between successive frames of imagery, thereby assisting in the development of efficient algorithms during the analysis phase.

Researchers therefore chose the Dalsa Piranha 2 line scan camera as the imaging sensor. The camera is capable of acquiring 36,000 lines per second with the resolution of each line being 2048 pixels. The specifications of the camera are outlined in Table 1.

SPECIFICATIONS

Feature	Spec
Part Number	P2-2x-02K40
Resolution	2048 x 1
Total Data Rate	80 MHz
Max. Line Rate	36 kHz
Pixel Size	10 µm
Number of Camera Taps	2
Output Format	Camera Link Base
Size	50 x 85 x 50 mm
Mass	< 450 g
Responsivity	up to 76 DN(nJ/cm2) @ 10 dB gain

`Table 1: Specifications of the Dalsa Piranha 2 Line Scan Camera.

Image Acquisition: Prosilica GC1020 Gigabit Ethernet Camera

In addition to the line scan camera setup, researchers at the UCF Computer Vision Lab developed and tested a second system that involves the Prosilica series of Gigabit Ethernet (GigE) area scan cameras.

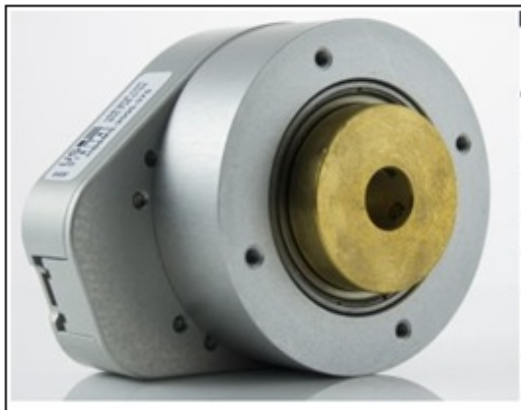
GigE is the first standard that allows fast transfers (1000 Mbps) of data using standard low-cost cables over long lengths of up to 100 meters. Researchers are primarily using this second system to address a new issue that they discussed with the Department's Track Inspector, Mr. Larry Jacobs that was not included in the original proposal. This issue involves attempting to verify if the tracks have shifted during the process of re-laying that is explained in detail in Section 4.6 of this report. The GC1020 is particularly suited to this application since it is required to obtain imagery at a high (near megapixel) resolution at the standard frame rate of 30 fps. The GigE protocol allows for the transfer of this data-stream in real-time. The specifications of this camera are listed in Table 2.

Resolution	1024x768
Sensor Type	1/3" CCD progressive scan Sony ICX204
Pixel Size (µm)	4.65 x 4.65
Maximum Frame Rate	33 fps at 1024x768
Lens Mount	C-mount with adjustable back focus
Digital Interface*	GigE Vision 1.0
Interface Type	IEEE 802.3 1000baseT
Exposure Range	10µs to 60s
Region of Interest (ROI)	Independent x and y control; 1 pixel resolution
Binning	Independent H and V control
Imaging Modes	External Trigger, Fixed frame rate, Software trigger
External Trigger Modes	Rising edge, Falling edge, Any edge, Level high, Level low
External Sync Modes	Trigger ready, Trigger input, Exposing, Readout, Imaging, Strobe, GPO
External Trigger/ Sync Connection	12-pin Hirose
Monochrome Modes	Mono8, Mono16**
Power Consumption	less than 2.9 W (12V)
Housing Size	33x46x38 mm
Weight	99 g

Table 2: Specifications of the Prosilica GC1020 Gigabit Ethernet Area Scan Cameras. (Mono16 mode is 16-bit and available only with specific camera models, found on the manufacturer's website.)**

Distance Measurement: HB6M Hollow Bore Optical Encoder

While acquiring high-resolution imagery is important, it is also critical to know the locations at which this imagery was acquired, since this information can help in localizing the position of the defect in the track from a known point such as a mile marker. This requires fitting a device such as an optical encoder onto the wheel of the Hi-Rail vehicle. The optical encoder provides a measure of the distance traveled along the tracks during the measurement. Researchers decided to synchronize the encoder output with the imagery acquired from the line scan cameras for improved information gathering. The specifications of the chosen optical encoder are shown in Table 3.



Parameter	Dimension
Shaft Speed	6,000 RPM max. continuous
Acceleration	$1x10^5$ rad/sec²
Starting Torque	.70 oz-in typical
Shaft Loading	5 lbs. max.
Weight	11.5 oz.
Shaft Runout	.030" axial .010" TIR
Moment of Inertia	$3.7x10^{-3}$ oz-in-sec²
Vibration	20 g. 5 to 2KHz
Shock	50 g. @ 11mS

Table 3: Specifications of the HB6M Hollow Bore Optical Encoder.

Synchronization: The optical encoder outputs a pulse for every count and these pulses are used as an input to the camera-link boards to trigger the line-scan cameras to acquire an image.

Illumination of the railroad tracks: Custom Built Super-Bright LED Lights

Researchers tested line scan cameras to acquire a few images of the railroad tracks. Personnel working on the project found that the quality of the images was very good, but subject to variations in lighting. A sample image from the two line scan cameras on either side is shown in Figure 3 below. The black band is the region outside of the field of view between both cameras.

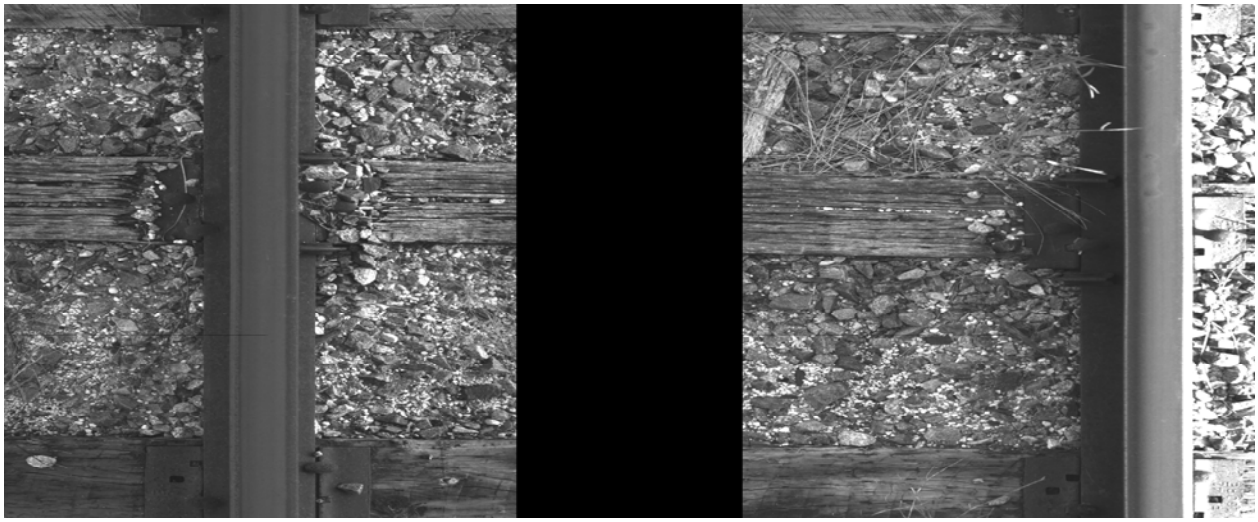


Figure 3: Imagery acquired from the Piranha 2 Line Scan cameras.

One can note in the image that the right edge of the image is over-exposed due to the sunlight being at an angle from the east in the picture. Consequently, a shadow is cast on the inside of the rail thereby causing a variation in illumination on either side of the rail. This makes it hard for a vision-based algorithm to cope adaptively with such changes since these cannot be modeled. To resolve this issue, researchers decided to use their own source of lighting to illuminate the railroad tracks. This required building a shade that would darken the area under the cameras, and prevent seepage of any external light. Following this, researchers could use a bright light source to illuminate the area to provide consistent lighting. This made the developed algorithms much more robust for the application. Sufficient amount of effort was involved in finding the appropriate light source to consistently illuminate the railroad tracks during image acquisition. Researchers tried several combinations of lights which are described below:

Plasma Light, High Pressure Sodium Vapor Lamps, Halogen Bulbs: A plasma lamp is characterized by the absence of electrodes and is energized using radio frequency power. A feedback amplifier circuit establishes an electric field that ionizes the gases and creates plasma at the bulb's center. This vaporizes the salts that join the plasma to give off an intensely bright white source of light that is rated at 140 lumens / watt. In comparison, standard bulbs give off about 15 lumens / watt. When testing this lamp for image acquisition, researchers encountered the following problems. The bulb flickered running on Alternating Current (A/C) causing artifacts in the image due to the high frequency line scan cameras. There was difficulty running the plasma light off Direct Current (D/C) due to its non-standard operating voltage of 28V. Similarly, a high pressure sodium lamp was tested, which flickered while running on a ballast and could not operate efficiently on D/C. Regular Halogen Lamps were simply not bright enough for the proposed application. Finally, the researchers decided to use a row of super bright Light Emitting Diode (LED) Lights. In the meantime, they used an overhead projector as a light source, since it is capable of producing a very bright light. The LED lights were expected to produce similar results. Researchers therefore chose Custom-built high intensity LED lamps as the light-source. The authors depict a graphical illustration of this platform in Figure 4.

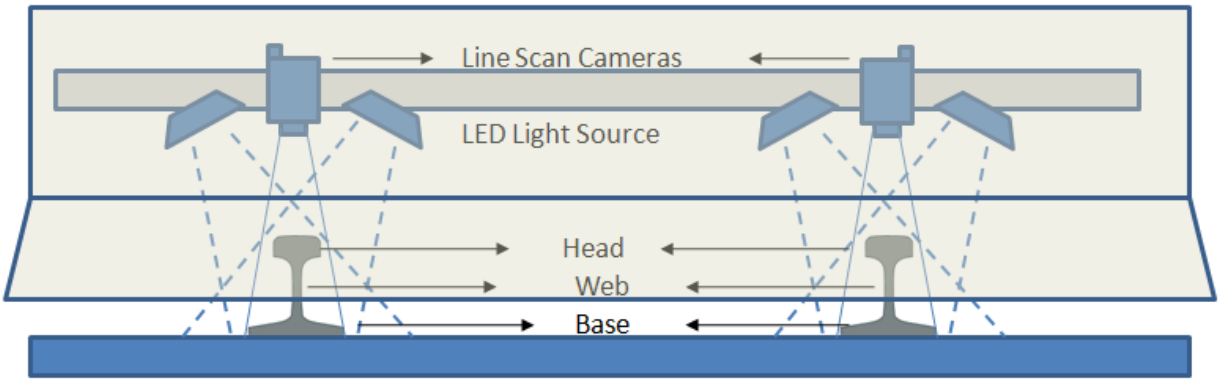


Figure 4: Graphical Illustration of the proposed setup for image acquisition.

As shown in Figure 4, researchers used a set of two LED Lights on each side of the rail, for each camera as the illumination source. They then worked with their configurations so as to illuminate the 'web' and 'base' of the rail to obtain a sharper image of the edge locations while at the same time enhancing the ability to detect defects in the ties such as missing bolts, spikes, scarring of the cross-tie, etc.

2.4 Encoder-Camera Interface for Image Acquisition

Researchers primarily use the laser setup described for high precision gage detection. Along with this setup, researchers have also developed a second setup consisting of high-speed line scan cameras. This setup helps obtain high-resolution images without any motion blur since the cameras have been setup to be triggered off an encoder pulse as the Hi-Rail vehicle moves. One can mount the platform in tandem with the lasers, or separately either at the front or back of the Hi-Rail vehicle for acquisition of high-resolution imagery. The authors briefly describe the construction and working of this setup in the section below.

By default, most cameras power up in *free-run* mode, acquiring and sending images continuously. This means that the camera acquires images at regular intervals. When one mounts these cameras on the Hi-Rail vehicle, several important issues need to be resolved:

(a) If the Hi-Rail vehicle is not moving, it results in the camera acquiring several duplicate images of the same location. This increases computational overhead during the processing of these images to identify defects.

(b) Acquiring images at a constant rate independent of the rate of movement of the Hi-Rail vehicle results in an overlap between successive images. A defect found in one image may also be in the next image but at a different location because of the overlap. The algorithm would therefore need to keep track of every image processed, the location of the defect, and differentiate between a new defect and one that has already been found. To do this, it will need to know the exact speed at which the Hi-Rail vehicle is traveling at the time of acquisition. This may be impractical.

(c) The faster the Hi-Rail vehicle moves, the lesser the resolution of the area under the image (i.e. the image will appear stretched out or contracted depending on the speed at which the Hi-Rail vehicle moves). This makes it very challenging to apply vision algorithms that use shape-measures such as the Hough transform to detect missing bolts, cracks in ties, etc.

Researchers addressed these issues by using a pulse from every encoder count to trigger the camera to acquire a single line of an image that is 2048 pixels wide and one pixel in height. As a result, the image resolution is independent of the speed at which the Hi-Rail vehicle moves. Additionally, the camera does not acquire duplicate images since no pulses are output from the encoder when the Hi-Rail vehicle is stationary. The camera acquires a single super-high resolution image at the end of every run. For example, if the Hi-Rail vehicle moved 100 feet, and encoder output 25000 pulses in accordance with the corresponding wheel circumference, the camera will acquire a single image that is 2048 x 25000.

In order to achieve this triggering setup, researchers setup the camera-link board so that an external device (the encoder) sends the trigger. A Transistor-Transistor Logic (TTL) signal is input to the board, which in turn sends out a Low-Voltage Differential Signal (LVDS) or Recommended Standard 422 (RS422) signal (depending on the board and its configuration) to the camera trigger line. The two pins of each trigger drive an opto-coupler through a 130-Ω series resistor. A few extensions to convert between single-ended and differential modes, together with some adapters, were required to interface the encoder to the camera link boards. The layout of this interface circuitry is shown in Figure 5.

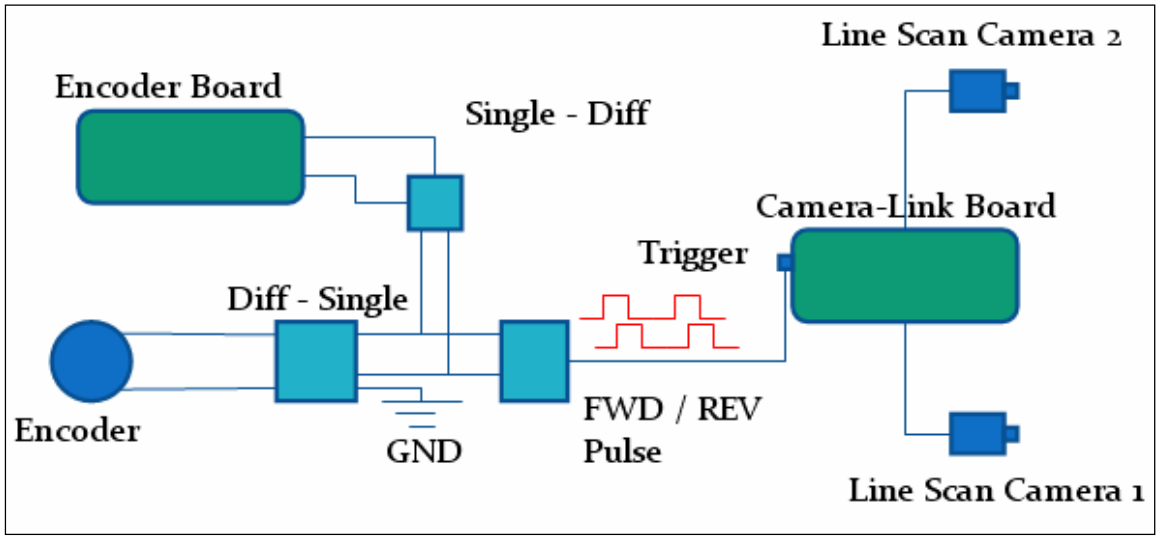


Figure 5: Interface Circuitry for Triggering

3. Software – Graphical User Interface (GUI)

Researchers installed the rail gage measurement system consisting of the optical encoder and laser sensors in the Hi-Rail vehicle. They have developed a Graphical User Interface (GUI) using the Internet Communications Engine (ICE) interface and LabView written in C/C++ and use dynamically shared libraries. Several enhancements to the system have been made to ensure its robustness. A known issue with the system has been the tendency of the on-board computer to shutdown while running off the inverter that is powered by the Hi-Rail vehicle's power source. To mitigate this, researchers designed the system to shutdown gracefully in the case of power failure thereby preventing any corruption to the onboard memory that may prevent the measurement system from functioning as desired. Additionally, researchers have designed a simulator on the on-board computer that they can use to mimic scenarios that the external hardware can experience. This is particularly useful in replicating circumstances in which failure may occur and designing the appropriate algorithms to cope with such scenarios. Finally, a start-up check is performed to ensure all drivers and Dynamically Linked Libraries (DLLs) are loaded without any errors, and an audible confirmation of the same is provided. Researchers have customized the interface to the system to meet the requirements as requested (i.e. it is easy to understand, user friendly, etc). A screenshot of the system is shown in Figure 6.

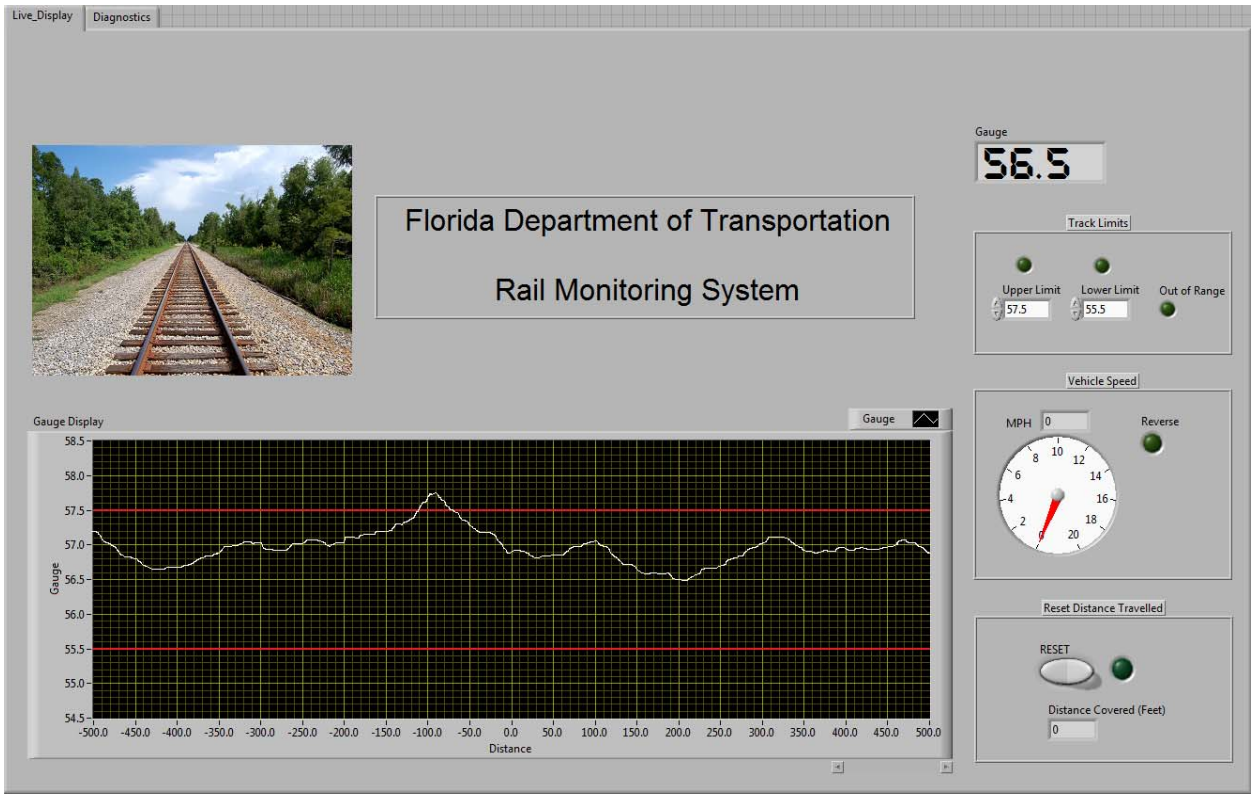


Figure 6: Screenshot of Application Software for measuring gage.

The Graphical User Interface (GUI) Window consists of the following components:

Live_Display: The main window is the live display window. The inspector primarily uses this window while monitoring the gage. It has a large graph display that plots the movement of the Hi-Rail vehicle on one axis, and the correspondingly measured gage by the laser sensors on the other axis. There is a user-input area, in which the inspector can key in the maximum and minimum allowable gages. The current gage being measured is displayed in digital format in a box on the right. If this value is not within bounds of the upper or lower user-defined limits, a light on the GUI turns on, and a 'Critical Error' waveform file plays alerting the inspector that the gage at the location is not within limits. He can then look at the graph on the window to view the deviation of the gage from the bounds. The inspector can then drive the Hi-Rail vehicle forward or backwards to find the exact location at which the gage is over the limit. As the inspector drives forward or backwards, the graph updates itself since the encoder motion triggers it. Researchers provide a speedometer gage in the user interface that monitors the speed of the Hi-Rail vehicle. An additional feature is the inclusion of a reset button. As the inspector covers a certain

distance on the track, he may choose to reset his trip count so that previous gage values are erased and the gage values beginning from the new position are recorded. As an example, this gives the inspector the ability to drive up to a mile-marker, reset the graph so that the new data displayed only includes information from the mile marker onwards, and indicates the distance covered during the inspection. Authors show a partial screenshot of the back-end implementation of the GUI in LabView in Figure 7. It involves using a drag-drop modular interface, to build the application with appropriate wiring that includes the use of numerical and algebraic operators,Analog to Digital Converter (ADC) / Digital to Analog Converter (DAC) modules, communication interfaces, dynamic libraries, etc. that run using the compiled C / C++ code, and display components.

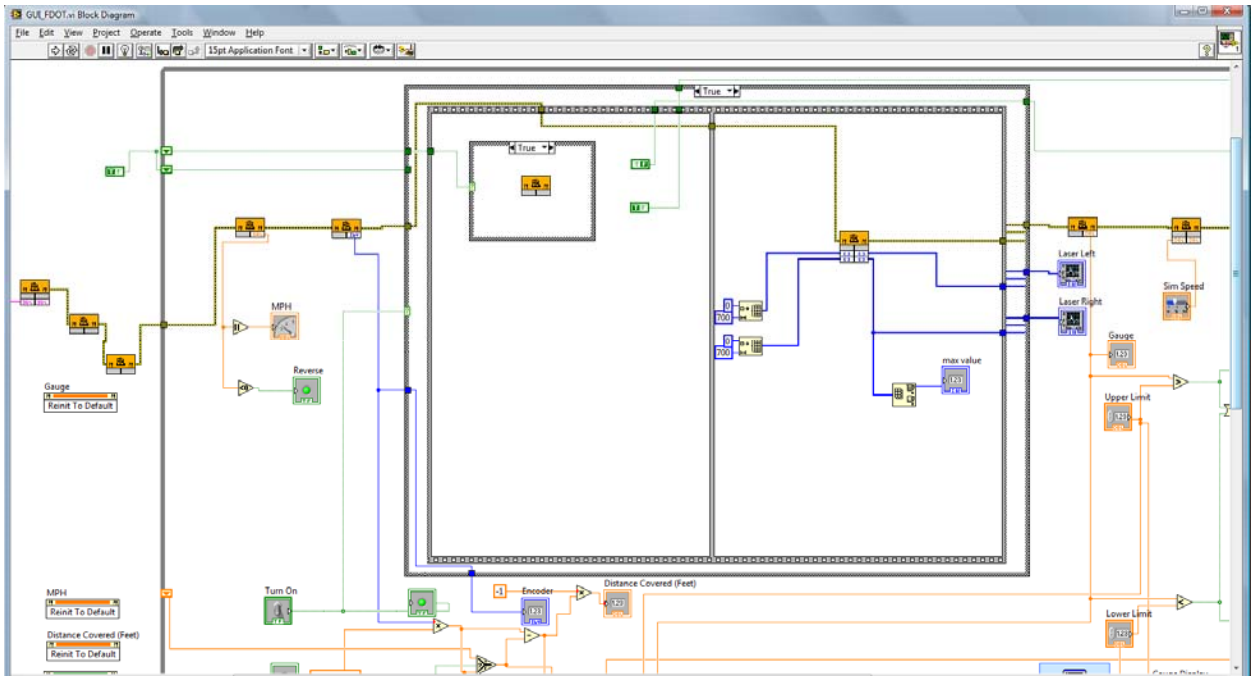


Figure 7: Partial Screen-shot of the back-end implementation of the GUI in LabView.

Diagnostics: Researchers primarily use the diagnostics tab for trouble-shooting. It contains a display of the raw laser readings as well as the raw encoder values with a switch that one can use to turn the views on or off. Researchers have also designed the diagnostics window to accept inputs from a simulator to demonstrate the running of the Hi-Rail vehicle, which can read log files from previous runs to detect problems. This is a valuable tool for the inspector since he can use this display window to ascertain if the system is fully functional prior to beginning his run. At the beginning of the run, the raw lasers will detect any object underneath them and display their inverted profile. Movement of the Hi-Rail vehicle

will update the encoder values. Failure of any of these two components will indicate a non-functional system. The inspector can report the problem along with the recorded log-file to the personnel at UCF.

4. Technical Details – Defects Being Inspected and Results

In this section, the authors highlight the algorithms and procedures involved in finding defects in the tracks. They have been developed and extensively tested in a lab environment. Details of the platform used for data collection are included in the Appendix. The system level implementation of these, particularly to work in real-time, is however subject to certain controlled lighting conditions, with researchers making every effort to address these issues. With the infrastructure already in place, it is possible to implement these methods at the system level, but this may require additional offline computing. The researchers have implemented and tested extensively the algorithm for gage detection at the system level, with the installation of the unit on the Hi-Rail vehicle.

4.1 Measuring Gage using the Laser Scanners

The laser scanners measure a distance to the surrounding objects in polar (angular) coordinates. Every arc swipe of the scanner produces a set of 361 distance measurements in polar coordinates. After acquiring the data sets from both scanners, they are combined and aligned using a mean square distance family algorithm. This algorithm tunes the parameters such as the relative polar coordinates (length and angle values) of the data set, the relative position, and distance between cameras. After this phase is completed, the two data sets are merged into a single rail profile as seen in Figure 8.

Knowing that the distance between scanners is fixed, the polar coordinates are converted into Cartesian *(x, y)* coordinates. Due to the nature of the profile of the rail head, the topmost points in *y* dimension usually correspond to the location on the surface of the rail head. In addition, the rail's position in the scanner data set does not vary much and allows the algorithm to search for maximums only in the subset of data. In this way, researchers find points on both railheads. All adjacent points on the rail heads are found and thus the extent of both rail heads in the *x* dimension is determined. Researchers also find the edge (innermost) points on the railheads. The edge points are the ones between which the rail gage is measured. Once the (*x, y)* coordinates of the edge points for both rails are determined, the rail gage is calculated. The precision of *x* location of edge points is based on the frequency of scanner measurements in *x* dimension. Thus, if it is determined that those 10 points belong to the railhead, the researcher knows that the railhead is 2 ¾ inches wide, the precision is 0.27 inches, which is about 7 mm. Most railways in the United States use the standard rail gage of 1435 mm (56.5 inch). According to most

railroad regulations, this rail gage is defective if it varies more than 1 inch plus or minus from the standard rail gage.

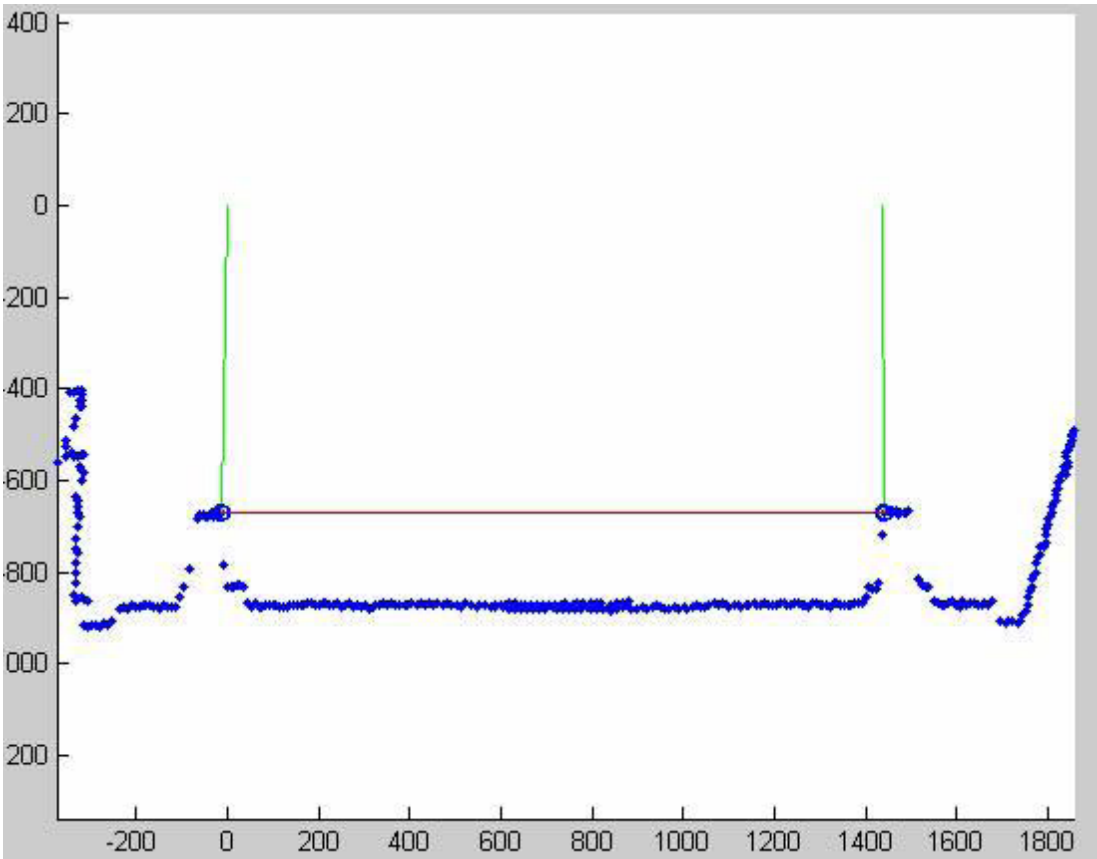


Figure 8: Merged profile from both laser scanners.

The precision of rail gage measurement with SICK scanner is +/- 7.5 mm, which is about 0.59 inches. Thus, this method is effective at detecting rail gage defects. Researchers have performed many experiments during different times of day, different weather conditions and on different railroad tracks. A sample 3D profile obtained from the laser scanners is shown in Figure 9.

Researchers have also made an effort to improve the gage measure across grade crossings and forks, where the configuration poses a challenge to the way the laser scanners measure the gage. They calculate the gage by finding the exact location of the rail edge from which an accurate measure of the rail gage is computed. The input to the system is one-dimensional data from the laser scanners in the polar coordinate system. Any outliers in this data are eliminated by accurately implementing digital

mean and median filters. The next step is to convert the scanned laser readings from polar coordinates to Cartesian coordinates. An edge map is then found which includes all the possible locations of the rail edge. To achieve this, researchers compute the discrete derivative of the data from the laser sensors in Cartesian coordinates. In order to have one strong edge instead of several weak consecutive edges, an algorithm called 1D non-maxima suppression is applied. The new rail edge is not expected to be very far from the previously found rail edge, since this is in keeping with the speed of the laser scans. The new edge is computed using information about previously found and currently found edges.

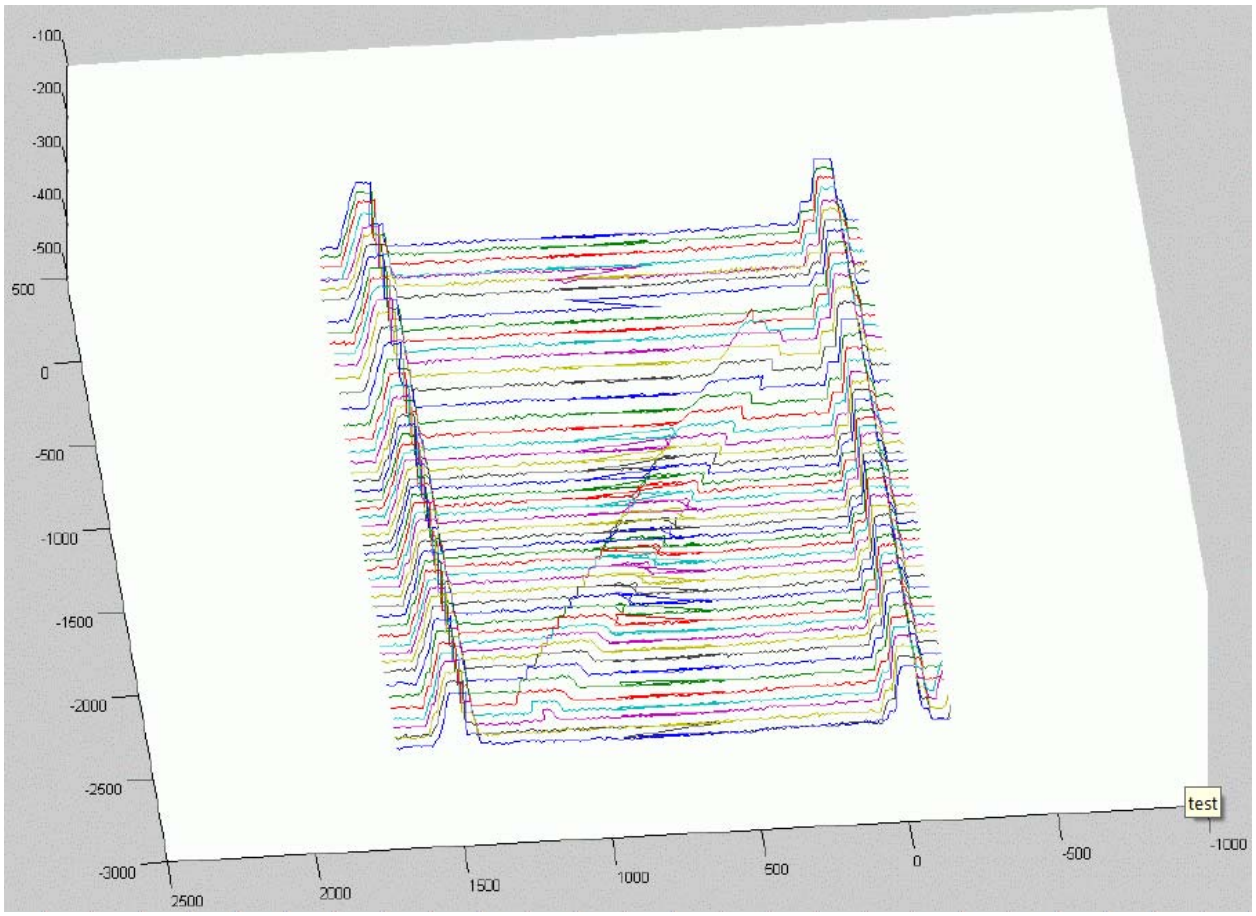


Figure 9: Sample 3D Profile obtained from Laser Scanners.

Researchers strengthen the edges that are close to the previously found edge while the edges that are far away from the previous edge location are weakened. A Gaussian function determines the degree of intensifying and attenuating. Since the railhead has a specific width and the approximate width of the railhead is known, the edges which have another edge within the distance of railhead width on their

back are intensified. When an edge is followed by another edge and the distance between the two edges is more or less close to the standard railhead width, the probability of this being the right edge increases. Finally, researchers approximate the rail gage width to stay within bounds of a certain threshold and remove the rail edges that do not conform to this relationship. The output of all of these steps is a map of edges with the value assigned to each edge being a probability of being the actual rail edge. The algorithm picks the edge with the highest probability as the actual rail edge. Researchers define this algorithm to be completely generic in nature so it does not require special steps to compute the rail gage at forks and crossings. Once researchers find the location of two rail edges, they measure the distance between them to obtain the rail gage (Figure 11). This process has been implemented on the system to work in real-time and has been tested and performs satisfactorily.

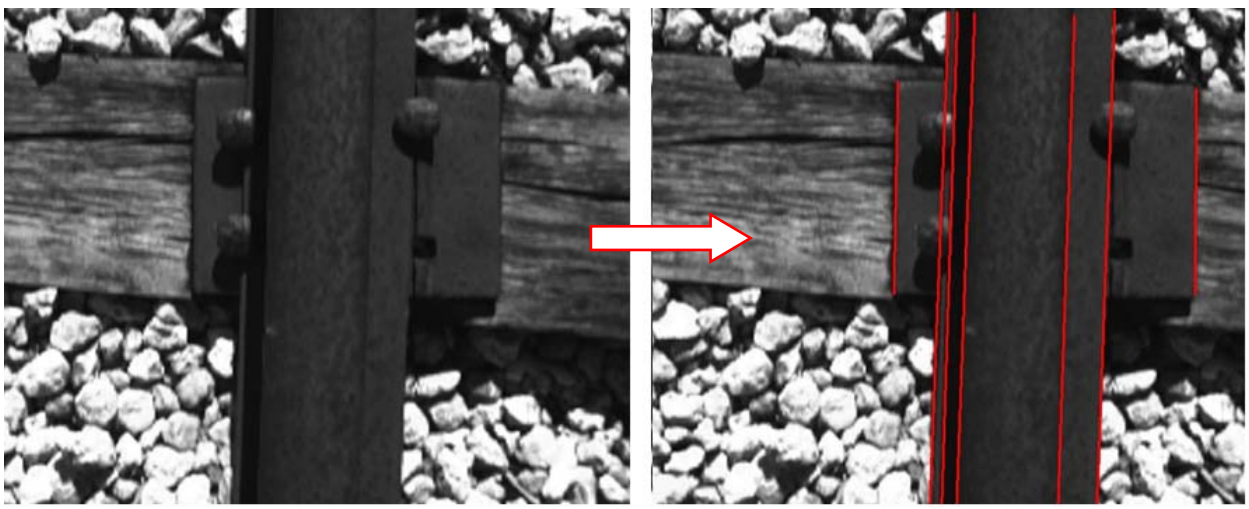


Figure 10: Algorithm depicting potential edges found on the rail in an image, before filtering.

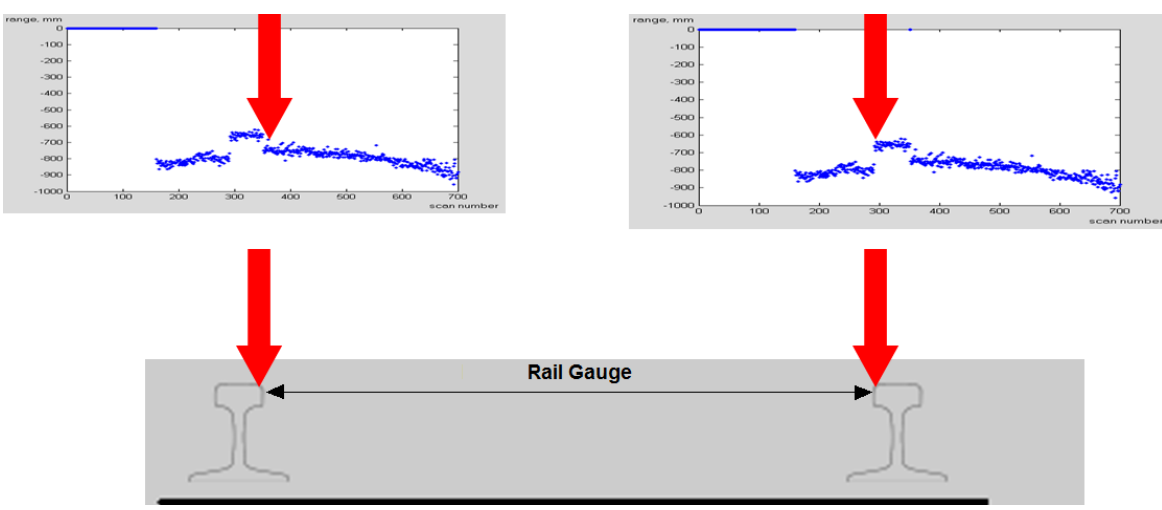
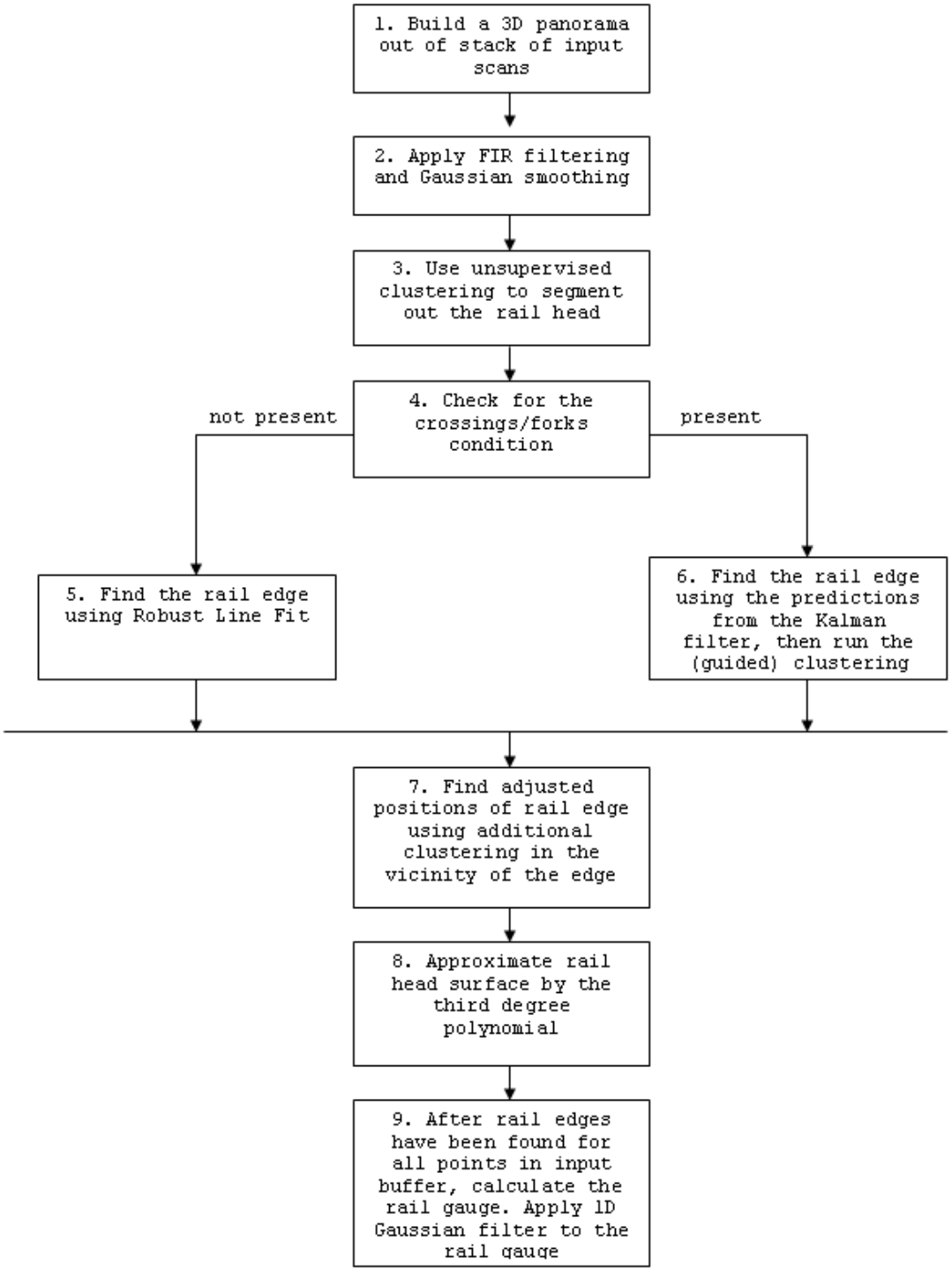


Figure 11: Measuring the Rail Gage using the laser scanner readings.

A detailed description of the various steps of the algorithm used for gage detection is provided. The algorithm to find the rail gage consists of several steps outlined in Schematic 1.



Schematic 1: Flow Diagram rail edge detection and gage measurement

The basic principle behind the working of the algorithm involves determining the inner edges of both rails in the scanner views and, knowing the distance between scanners, calculating the distance between the rails in inches. First, researchers build a 3D panorama of the rail out of a stack of input scans (step 1). The scale of the panorama buffer in the temporal dimension is determined by the hardware encoder, the results of which may be in turn improved by a built-in high precision Global Positioning System (GPS) unit. The encoder counts the number of cart wheel rotations and thus presents a measure of distance along the track with a relative precision of about 1%. If one knows spatial geometry of the track beforehand, the readings of the GPS unit can further improve the distance measurement. After the panorama is built, Finite Impulse Response (FIR) circular filtering and Gaussian smoothing is applied to the panorama buffer to suppress the noise component (step 2). Researchers apply two different algorithms to filter the input data sets. At the first step, a low-pass FIR filter is applied to remove high frequencies from the input data. This is followed by applying a standard 2D Gaussian filter in the desired temporal direction. In the next step (3), researchers attempt to segment the railheads in the panorama buffer. Due to the geometry of the rail, all data points are roughly grouped around 3 clusters: clusters S_1 and S_2 are for points to the left and right side of the rail, while cluster S_3 is for the points on the railhead. The authors illustrate a typical example of such clustering below.

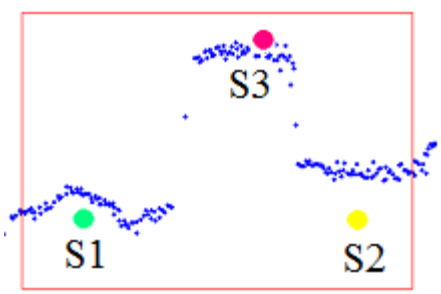


Illustration 1: Clustering

To separate the railhead cluster S_3 from the other two point clusters the iterative k-means clustering algorithm is used. Researchers then employ the method (step 4) which detects railroad crossings or forks in the panorama buffer. If a railroad fork is present then there is more than one railhead in the scan, and for both rail fork and rail crossing conditions the ground is level with the railheads, which makes railheads inseparable from the ground. There is, however, a small groove on the ground where cartwheel edges fit in, which is always present. The general solution to this situation is to not use k-means clustering but instead to track the positions of rail edges found in previous scans. In order to properly handle such situations it is required to first employ a method that detects unusual conditions in the scans. Researchers made use of a low-width 1D sliding window, which examines every input scan for the presence of unusual conditions. The responses from the scan procedure are given as the probability scores; the higher the score the higher the probability that the scan contains the forks/crossings. If they

are not present, the algorithm finds the rail edge using robust line fit (step 5). If they are present, an alternative is used.The rail edge positions are predicted using a Kalman filter (step 6). In the next step, for both fork/crossings conditions, the adjusted positions of rail edges is found using additional (mean shift) clustering in the vicinity of the edge. The rail head surface is approximated by the third degree polynomial and then two plane surfaces are fitted to find the exact position of the rail edge. Lastly, using rail edge information, the rail gage is calculated and smoothed with a 1D Gaussian filter.

Evaluation:

Researchers performed experiments on data obtained from visiting different track locations in Central Florida. During the experiments, researchers accumulated over 1000 GB of laser range scanner measured profiles of railroad tracks in which the profiles were synchronized for both rails. The experimental validation was performed in-situ by measuring actual gage and comparing it to calculated values. The indirect algorithm validation was performed by running a railroad cart over the same segments of track more than once and comparing the obtained gage curves. Researchers found the precision of the experiment to be consistent with the precision of the hardware used in the experiment. In the experiment, the cart was run over the same track more than once. The resulting gage curves are overlapped in Figure 12. It is observed that overlapped gage curves closely match. This performance level is typical of real world systems that are characterized by imperfect sensor readings and noise. It is practically impossible to produce exactly matching graphs.

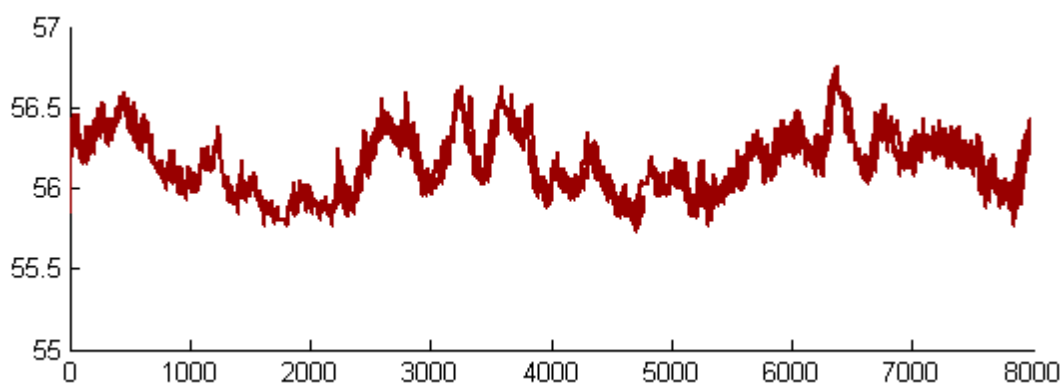


Figure 12: The profile of a rail gage measured in inches at a track segment of about 8000 feet long.

The detailed evaluation of the algorithm on the whole set of 1000 GB of data as well as a qualitative comparison to both other major algorithms, camera-based and laser triangulation methods, was performed and found to be acceptable.

4.2 Engine Burns

Engine burns are a class of defects that occur on the top of the rails due to friction from the slippage of the locomotive wheels. Although engine burns alone do not indicate a critical problem, if not attended to, burn fractures can develop which may lead to potentially hazardous operating conditions. The identification of engine burns consist of identifying round or oval rough spots on the head of the running surface of the rail. Figure 13 shows the flow diagram of the process used to detect engine burns.

Figure 13: Flow diagram of the algorithmic steps in finding engine burns.

The first phase in detecting engine burns consists of generating a panorama buffer based on a set of frames. Due to the high frame rate of the video camera, significant amounts of overlap exist in any set of consecutive frames. Therefore, in order to avoid detecting defective instances multiple times researchers employ a moving window approach. They construct a panorama buffer by extracting and matching Scale Invariant Feature Transform (SIFT)[2] features between a set of frames. SIFT features are located at scale-space maxima/ minima of a difference of Gaussian function. At each feature location, a characteristic scale and orientation is established. This gives a similarity-invariant frame in which to make measurements. The movement of the mobile platform is for the most part, translation. However, although the cameras are fixed to a mobile platform, small vibrations during data collection have the potential of introducing artifacts in the panorama buffer. Therefore, small vibrations are accounted for by assuming that the camera can rotate in small angles about its optical center. Researchers then extract

a set of SIFT features from every group of n frames. These features are then matched to one another by performing a k-nearest neighbor search. Subsequently, the researchers find all overlapping images from the previous step. This is done by considering the top *m* frame matches that have the greatest number of matching features as possible matches. Finally, the researchers use Random Sample Consensus(RANSAC) [3] to select a set of inliers based on the homography constraint. Once all of the pairwise matches have been established between frames, a panorama is created by stitching matching frames together.

The researchers then use a Gaussian Filter to perform smoothing of the image to eliminate noise. They then segment the image using Mean-Shift [4] segmentation following which a series of features from each image region is extracted. The shape of the region is determined by measuring the circularity of the area using the circular Hough transform [5]. In addition to determining the circularity of each region, researchers employ color moments as a means of determining intensity value distributions. Given that the intensity values in a region follow a certain probability distribution, the moments of that distribution can be used as features to identify engine burns based on color. Additionally, researchers use homogeneity and entropy of the region to reflect how uniform the image region is.

The researchers model engine burn defect detection as a nonlinear classification problem using a Support Vector Machine (SVM) classifier [6, 7, 8] with a Radial Basis Function (RBF) kernel [9]. An SVM maps the set of input features to a higher dimensional feature space via a kernel scheme. Subsequently, it generates a classification hyperplane based on the maximum margins in the feature space. We performed a series of experiments in order to assess the performance of our method. By varying the number of training examples we evaluated the effect of the training set size. The results of this experiment are depicted in the form of a Receiver Operating Characteristic (ROC) [14] curve.

These regions may also contain holes and hairline cracks in more severe cases. An appearance-based approach to modeling this class of defects is required in order to effectively classify potentially hazardous operating conditions in the presence of textured backgrounds on rail surfaces. The method selected combines morphological filters and texture analysis to detect and segment engine burns in video sequences obtained from a mobile platform. In addition to determining the circularity of each region, researchers employ color moments as a means of determining intensity value distributions. Given that the intensity values in a region follow a certain probability distribution, one can use the

moments of that distribution as features to identify engine burns based on color. Figure 14 illustrates some results as applied to the visual data. The detection of engine burns is heavily dependent on the number of training samples that are acquired. The number of training samples is small since engine burns do not occur frequently and hence, it is difficult to implement this at the system level. This feature is therefore not a part of the railroad inspection system, but the algorithms have been tested on a few images in the laboratory and shown to work under controlled conditions.

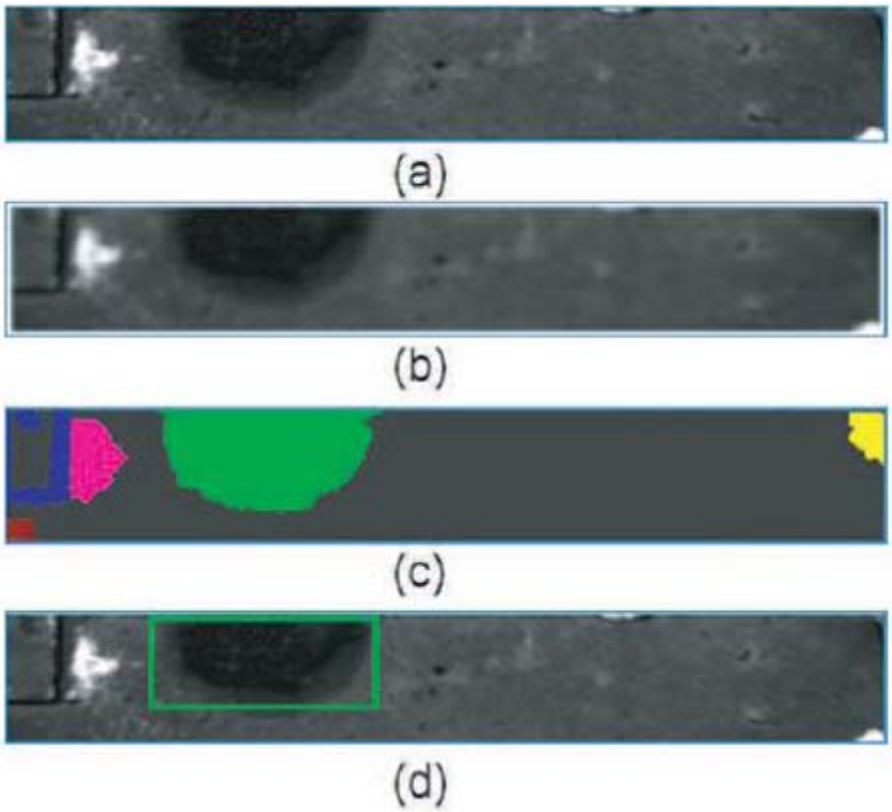


Figure 14: The main phases of the detection process. (a) The original panorama frame, (b) The blurred frame region, (c) The mean-shift segmentation results, (d) Detection results.

4.3 Cracks in Rails

Another class of defects for which researchers have developed a solution in the lab, is detecting the presence of cracks on the rails. They depict the main steps of the algorithm in Figure 15.

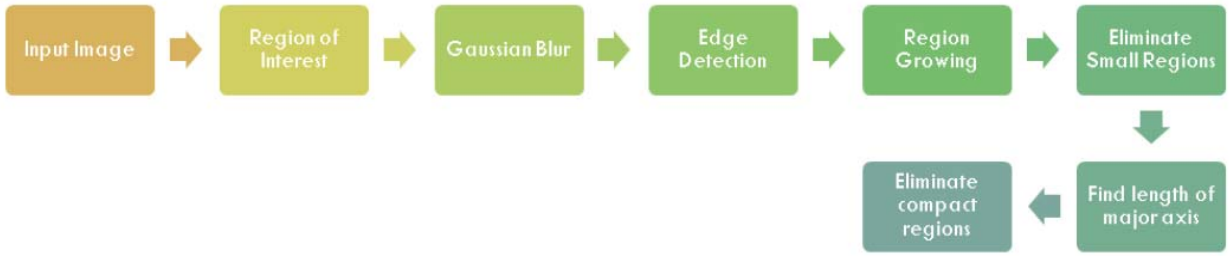


Figure 15: Flow diagram of the algorithmic steps to detect cracks in rails.

Given the location of the rail boundary, a Gaussian Kernel is used to smooth the region of interest. The Gaussian smoothing helps to filter out the noisy pixels in the image. Next, researchers perform edge detection using one of several detectors including the Canny [15] and Sobel [16] edge detectors. Region growing, which is a morphological operation to segment an image based on specific regions is then applied to grow each of the resulting lines. This is an iterative approach to segmentation and involves examining neighboring pixels of an initial 'seed' pixel and determining whether or not the pixels must be added to the region. An illustrative example of region growing applied to segment a streak of lightning in the sky can be found online [10]. Regions contained within a minimal area (which is given by a preselected threshold) are then eliminated. The remaining regions are processed using Principal Component Analysis (PCA) [11], after which the major and minor axes are extracted; these values are used to eliminate compact regions, resulting in a series of crack detections. Illustrative results are presented in Figure 16, outlining the depictive steps of the algorithm described in Figure 15. Although this algorithm has been shown to work in the lab, there have been insufficient samples of images in which these cracks exist to be able to thoroughly test the algorithm. Additionally, the process is highly sensitive to the range of thresholds chosen during the blurring and edge detection stages. Variation in lighting significantly affects these thresholds and hence the algorithm fails to cope, often falsely finding 'cracks' in rails due to image artifacts caused by shadows or reflections.

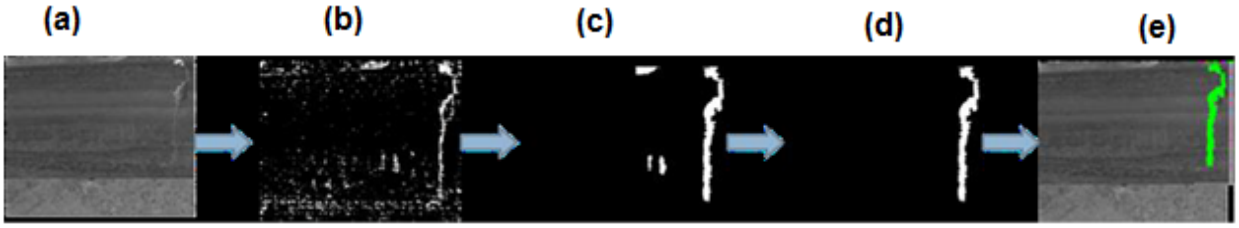


Figure 16: Stages of the detection process – (a) Original image, (b) Blurred image, (c) Edge detection algorithm, (d) Eliminating regions according to their major axis length, (e) Final result.

4.4 Scarring across Cross-Ties

This defect is visually distinguishable from both the tie-plate and the cross-tie itself. The first appearance of the defect is in the form of scarring on the wood, while the second defect is visible as a change in color. In order to detect this defect, researchers have defined a filter with four different orientations. Authors show the orientation masks for different window sizes in Figure 17.

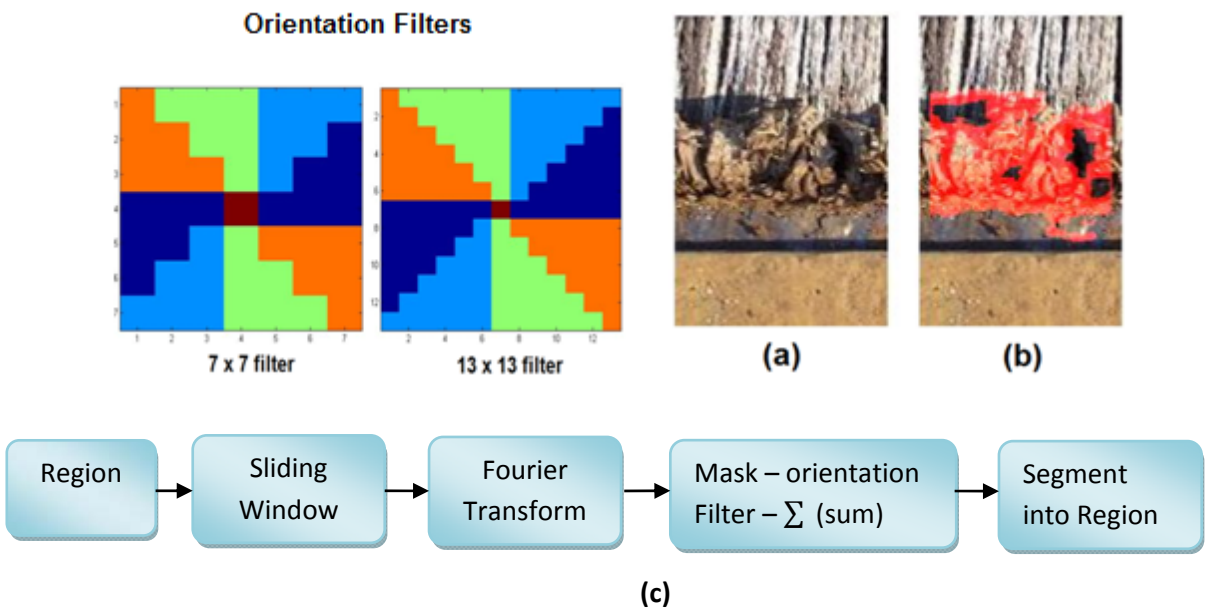


Figure 17: The defined orientation filters of different sizes used to find the defect. (a) Input image showing scarring of wood. (b) Segmented image detecting scarred region on the cross-tie. (c) Stages in the process.

First, researchers employ a sliding window over the potential region of defect. When the window is moved, the Fourier transform for every patch is calculated. Then, they use the orientation filter as a mask, and put the mask on the calculated Fourier transform for every patch. The sum of Fourier transform values which fall inside every orientation is considered as a value which represents an approximation for the orientation of the fabric inside that patch for that specific orientation. The scarred part of the wood has no specific pattern while the untouched part of the wood has a defined orientation along a single axis, which results in the orientation filter returning a high value for areas of the fabric for which no specific pattern is found. The output of the orientation filter is then segmented and the resultant corresponding to the defective area is shown in Figure 17.

The steps of the process are shown in Figure 17 (c). Alternatively, since the defect is visible in the form of some change in the color, color based segmentation can be performed. Researchers cluster the region of interest based on color in Red, Green, Blue (RGB) color space and using K-means clustering. To overcome the drawback that the segmented area of the defect is scattered over the image due to a similarity in color with other parts of the image, the location of every pixel is taken into account and the horizontal location of each point along with its RGB values is fed into the K-mean clustering algorithm. The result obtained is shown in Figure 18. The process of clustering and segmenting involves a lot of computational time and hence the algorithm has not been implemented at the system level.

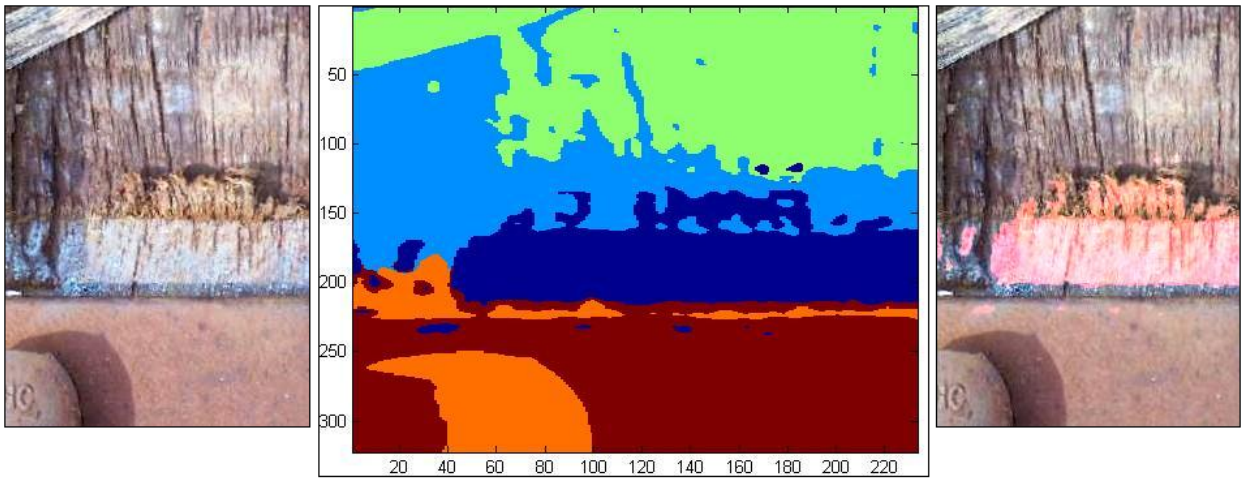


Figure 18: Color based segmentation with K-means clustering.

4.5 Missing Rail-Fasteners

Detecting missing rail fasteners is another area that researchers investigated thoroughly. This has however not been implemented at the system level, since the algorithm requires large amounts of training data to be effective, and also requires a lot of time for computation and classification. First, researchers collected a database of rail fasteners. Steel clips refer to the fasteners that, when viewed from above, appear to be a rectangular washer held down by a bolt; E-clips have a shape similar to the English letter □E□ or its vertical reflection and are sometimes known as Pandrol standard fasteners; fast clips have the shape of the English letter □w□ and are also manufactured by the Pandrol company. These shapes can be seen in Figure 20 (a). The researchers then annotated the videos to keep track of the number and location of all rail fasteners in the current video collection. Currently, researchers have annotated 2195 rectangular steel clips, 565 E-clips on concrete sleepers (E-Clips (c)), 18 E-clips on

wooden sleepers (E-Clips (w)), 19 W-clips (fast clips), and 21 missing clips. The video collections typically contain 2 to 4 views of each of these unique clips. Fasteners vary significantly in their appearance, depending on their type for several reasons. Steel fasteners used different bolts. E-clips for wooden and concrete sleepers differed in appearance. Fast clips were found with and without plastic covers. Rocks, leaves, and other debris sometimes occluded portions of the fasteners. Despite the variance in appearance, as a whole the data was relatively similar in appearance leading one to expect practical results in experiments. Emphasis was placed on automatic detection of the missing clips. For this purpose a Maximum Average Correlation Height (MACH) filter was trained on a number of examples from the database. Given these examples, a single filter is generated which captures the intra-class variation of the clips class.

In the video, Scale Invariant Feature Transform(SIFT) is used from frame to frame in the video to find the correspondence between frames. Using SIFT features, the distance between frames is measured. As the Optimum Trade-off Maximum Average Correlation Height (OT-MACH) algorithm detects clips, distances between clips are measured. Unusually large gaps between detected fasteners become candidate locations for missing fasteners. False detections usually cause statistically small gaps between detected fasteners. As an exception, the ties / connectors are spaced closely in curves and at grade crossings. By predicting the expected location of the fastener, a likelihood of a false detection can be determined independent of the MACH correlation peak value. A missed detection or a missing fastener usually causes large gaps. By comparing the measured gap to the average gap, researchers can estimate the number of missed detections. Similarly, the location of these missed detections may be given and these frames of interest can be presented to the field expert for further review.

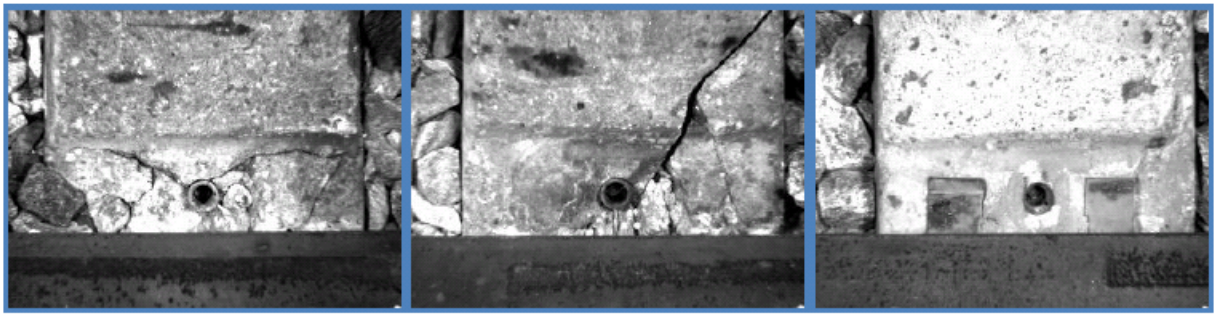
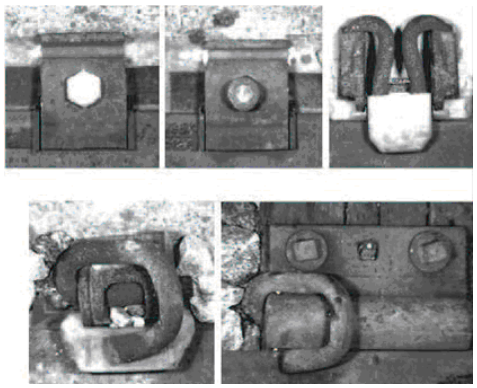


Figure 19: Typical appearance of missing fasteners.

In order to test the missing fastener detection algorithm, researchers created a database of fasteners to train correlation filters and measure detection accuracy. The standard steel fasteners and e-clips made up more than 90% of the data collected. The ratio of steel fasteners to e-clips was usually 2-to-1 and they appeared in that ratio almost regularly in the data. Only a handful of fast clips, e-clips on wooden sleepers, missing fasteners were present in our data collected. Typically, the video data contained 2-3 images of each fastener, some of which might have been clipped by the right and left edges of the image.

Experiments performed involved using 3-fold cross-validation to measure the separability of the classes chosen. Researchers randomly divided this data into three groups. Two groups were used as training data and the third group was used for testing. The separability of the classes was then illustrated with the confusion matrix shown in Figure 20 (b).



(a)

True Class							
Steel Clip	1890	1	0	0	0	0	0
E-clip (C)	1	492	0	0	0	0	0
Missing	0	1	12	0	0	0	0
Defective	1	0	0	0	0	0	0
E-clip (W)	0	0	0	0	17	0	0
Fast clip	0	0	0	0	0	19	0
Steel Clip #2	2	0	0	0	0	0	0

(b)

Figure 20: (a) Clockwise from the upper left: two steel fasteners, a fast clip, an e-clip on a wooden sleeper, and an e-clip on a concrete sleeper. (b) Confusion Matrix for missing clip detection.

4.6 Re-laying of Tracks

As mentioned previously, researchers have developed a secondary system that can be mounted on the Hi-Rail vehicle that the inspector can use to determine if the tracks shifted from their original positions during the process of re-laying. It is important that the tracks are laid back in the same place that they were originally at, since the ground beneath them is well packed and helps in keeping them in the same place, thus helping in maintaining the gage. The system being developed involves the GC1020 GigE cameras described previously, arranged in a unique setup as shown in Figure 21.

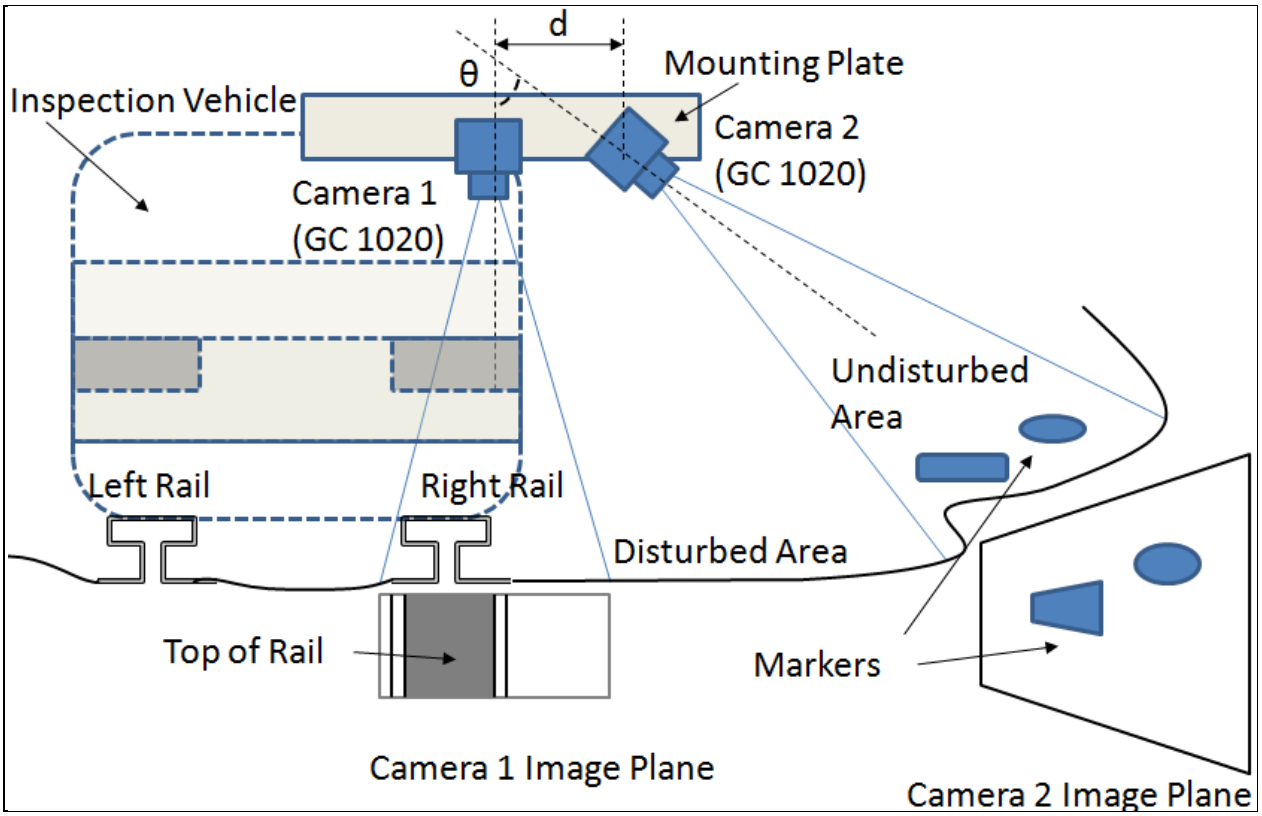


Figure 21: Inspection setup for detecting shifted tracks during re-laying.

The setup involves the two cameras separated by a distance *d* (m), and subtending an angle θ (rad) between them. The downward facing camera acquires images of the track prior to their re-laying. The area on the outside of the tracks is disturbed during the process of re-laying which involves lifting the tracks, packing the ground underneath it, and re-laying the tracks back in place. At a slightly farther distance, the ground is undisturbed and the personnel lay out markers prior to the re-laying process. The second camera is installed such that the markers are in its field of view. The images from the two

cameras are now stitched together prior to the re-laying process to obtain the first set of images. Once the re-laying process is complete, the Hi-Rail vehicle drives over the newly laid tracks and obtains the same two sets of images again. The images are again stitched together to obtain the second set of images. The first and second sets of images are now aligned using a process called image registration which involves computing the homography between the two sets of images. However, during this process, only the features (markers) in the second camera's field of view are used. Since these are unchanged across both sets of images, any shifting in the tracks between the initial and final runs will be detected during the registration process.

Figure 22 shows the two sets of images taken using the two GigE cameras before and after the re-laying task. On row 1, the images include the view from the downward facing camera, and the view from the camera that is looking at the undisturbed area prior to the re-laying process. On row 2, the same set of images is acquired after the re-laying of the tracks is completed.

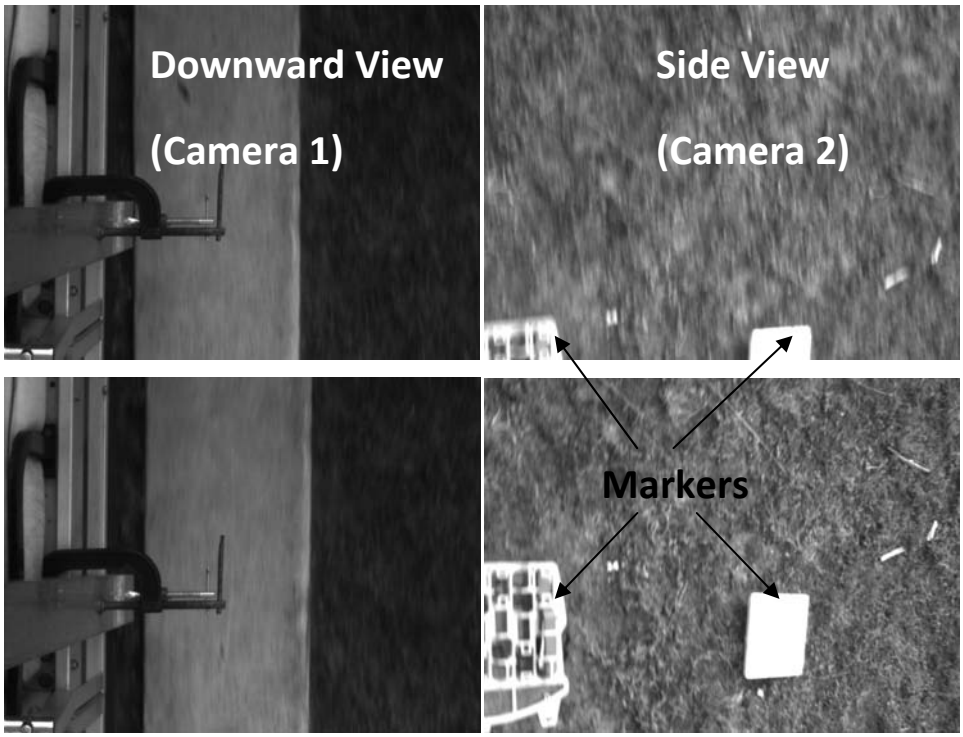


Figure 22: The two sets of images taken using the GigE Setup (Top 2 images – Row 1, Bottom 2 images, - Row 2).

In order to verify that the new position of the rails is precisely the same as the old location, the images showing the rail are aligned using information (features) from the images showing the undisturbed area.

Researchers based this on the assumption that the replacement process does not have any effect on the undisturbed area, and features remain in the same location prior to and after the re-laying process. First, the Homography transformation between the references images before and after the relaying are found. Homography is an image invertible transformation from the real projective plane to the projective plane that maps straight lines to straight lines. The Homography transformation is represented by a 3 by 3 matrix, which has the image translation and rotation elements in it. In order to find the Homography transformation between the reference images, a number of matching points are needed. These matching points are found using the SIFT method. SIFT or Scale-Invariant Feature Transform is an algorithm to detect and describe local features in an image, and is invariant to rotation, translation and scaling. Once the Homography transformation between the references images is found, the transformation is converted to a form which is applicable to the rail images taken using the downward facing camera. Since the physical structure of the hardware system is fixed, this conversion can easily be found and applied. The lining up of the rails in the downward facing cameras is dependent on the whether or not the location of the rail has changed during the re-laying process. In Figure 23, the left image shows a simulated result for well aligned rails, while the right one shows an example of the case when the rail has moved slightly during re-laying. The misalignment is in the horizontal direction.

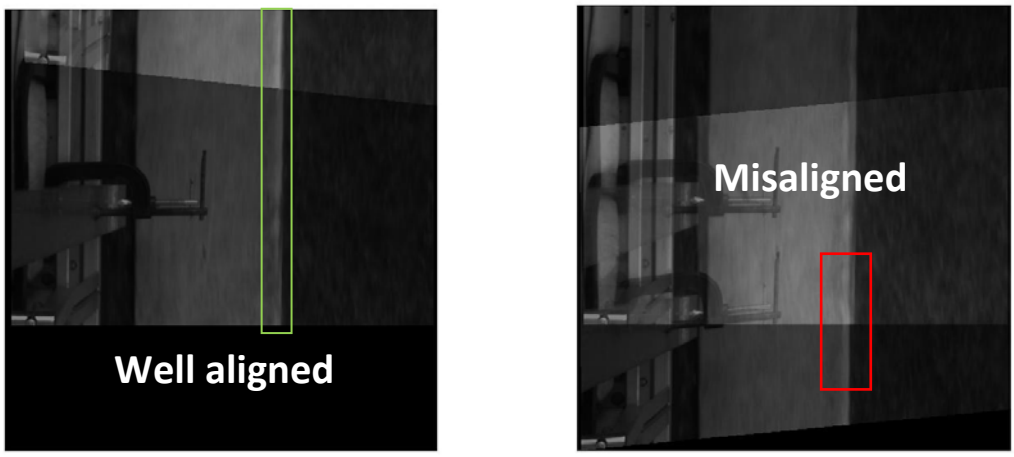


Figure 23: Simulated results showing aligned and misaligned images.

The markers being used are square-shaped and silver in color, making them easy to identify in the image. Elevation plays a key role in the appearance of these markers, since homography is only applicable in a 2D scenario. In the 3D scenario, additional information about the camera setup is required, which provides the necessary shape information about the square markers. This can then be used to determine the exact location of the markers with respect to the rails in the original image. Currently, elevation information is not being used, and it is assumed that the region around the tracks is reasonably flat to be approximated to be 2D. However, this does not apply to areas that are re-ballasted and/or curves, where replacement would result in restoration of super elevation. As a result, it is insufficient for the application and research in this direction forms a major component of the future work. As a result, this module has not been fitted on the Hi-Rail vehicle.

5. Conclusion

During the course of this project, the UCF Computer Vision Lab has worked on the research and development of an automated system for inspection of railroad tracks. Currently, a rail gage measurement system is in place on the Hi-Rail vehicle whose working has been shown to be satisfactory over the last few months based on feedback from the Department's rail inspector, Mr. Larry Jacobs. Personnel from the University (Technicians and Researchers) have also been visiting the site regularly and performing the needed services and upgrades on the Hi-Rail vehicle to ensure the proper functioning of the system. Personnel from the UCF Computer Vision Lab will continue to be in touch with Mr. Larry Jacobs and assist him with the functioning of the system, and to fix any problems that he may encounter. The system primarily uses high precision laser sensors to perform the gage detection, and includes an audio interface that not only alerts the operator when the system detects a faulty gage but also verifies the diagnostics of the system upon startup.

Several other defects are being inspected which include detecting engine burns, cracks in rails, missing fasteners, scarring of tie-plates and shifting of tracks during re-laying. However, the algorithms developed to perform this detection have been performed under controlled conditions, and research has shown that there has been difficulty in implementing these algorithms at the system level. Researchers continued investigation of the problem, and found a promising avenue for further work that includes setting up a controlled image acquisition platform on the Hi-Rail vehicle. This platform includes the encoder-camera interface described in this report, together with the shaded area and super-bright LED lamps. Researchers expect the platform to significantly improve the quality of the images acquired, meaning that the algorithms developed are much more robust. Currently, the variation in lighting of the imagery acquired together with the lengthy computation times for the developed vision algorithms have made it hard to implement these at the system level. With the advent of technology with superior computing power, dedicated hardware and a setup which provides controlled lighting under consistent conditions, it is possible to implement the algorithms that are described in this report at the system level.

References

[1] http://safetydata.fra.dot.gov/OfficeofSafety/

[2] Lowe, David G. (1999). "Object recognition from local scale-invariant features". *Proceedings of the International Conference on Computer Vision*. **2**. pp. 1150–1157. doi:10.1109/ICCV.1999.790410.

[3] Martin A. Fischler and Robert C. Bolles (June 1981). "Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography". *Comm. Of the ACM* **24**: 381–395. doi:10.1145/358669.358692.

[4] D. Comaniciu and P. Meer, Mean Shift: A Robust Approach towards Feature Space A Analysis *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 24, no. 5, pp. 603-619, May 2002.

[5] Daisheng Luo, Peter Smart, James E. S. Macleod, Circular hough transform for roundness measurement of objects, Pattern Recognition, Volume 28, Issue 11, November 1995, Pages 1745-1749, ISSN 0031-3203, DOI: 10.1016/0031-3203(95)00045-2.

[6] V. Vapnik, “The nature of statistical learning theory,” Springer-Verlag: New York, 1995.

[7] V. Vapnik, “Statistical learning theory,” John Wiley: New York, 1998.

[8] V. Vapnik, “The Support Vector method of function estimation,” in J.A.K. Suykens and J.Vandewalle (Eds) Nonlinear Modeling: Advanced Black-Box Techniques, Kluwer Academic Publishers, Boston, pp. 55–85, 1998.

[9] B. Schölkopf, K.-K. Sung, C. Burges, F. Girosi, P. Niyogi, T. Poggio and V. Vapnik, “Comparing Support Vector Machines with Gaussian kernels to radial basis function classifiers,” IEEE Transactions on Signal Processing, Vol. 45, No. 11, pp. 2758–2765, 1997.

[10] http://en.wikipedia.org/wiki/Region_growing

[11] Wold S, Esbensen K and Geladi P 1987 Principal component analysis Chemometrics Intell. Lab. Syst. 2 37-52

[12] http://www.tina-vision.net/

[13] http://www.wikipedia.org/

[14] Metz CE. Basic principles of ROC analysis. Semin Nucl Med 1978; 8:283-298.

[15] Canny, J., *A Computational Approach To Edge Detection*, IEEE Trans. Pattern Analysis and Machine Intelligence, 8(6):679–698, 1986.

[16] http://homepages.inf.ed.ac.uk/rbf/HIPR2/sobel.htm

APPENDIX A – Additional System Modules and Features

In addition to the already described system, researchers at the UCF Computer Vision Lab have also developed other visualization and control modules. These provide the user with a simple and intuitive interface which is described in the following subsections.

Mapping Interface

Researchers built a mapping interface on top of an existing GIS package called MapServer. This software provides the interface to industry standard SHAPE files and world image projection for conversion between longitude/latitude and other systems. This has allowed the team to acquire freely available mapping information from governmental sources such as Florida Department of Transportation and Florida Department of Environmental Protection. Through these sources, there is access to accurate rail, county, and road maps.

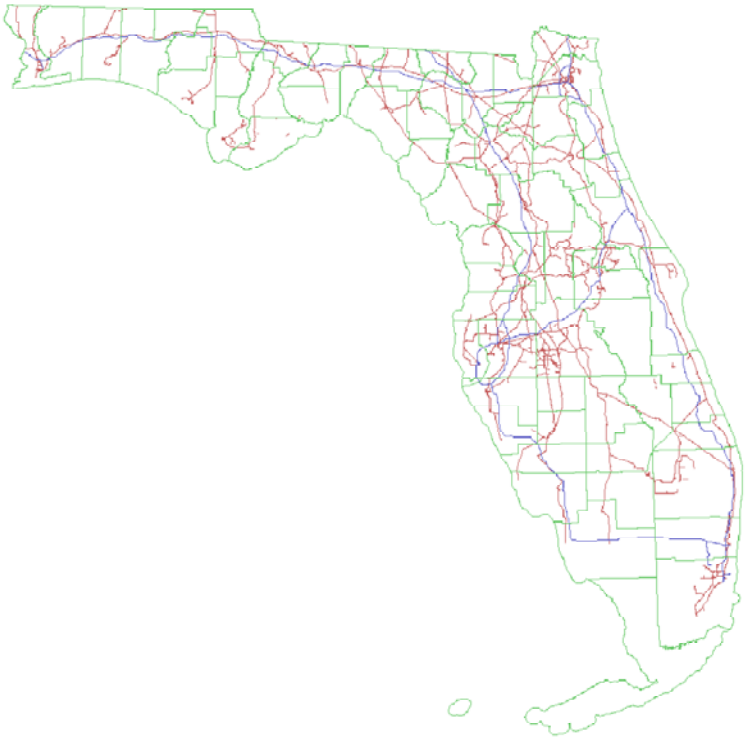


Figure 24: County (Green), Interstate (Blue), and Rail (Red) for the State of Florida.

Using another open-source project called Ka-Maps; users can view all mapping data through a web interface without requiring any installation of software on the user's (FDOT) computers. This software

allows for the selection of individual runs, panning, and zooming on areas of the state, and selection of data points. Finally, using the LABINS project from the Bureau of Survey and Mapping, one can use free aerial photography to give better awareness of the local area.

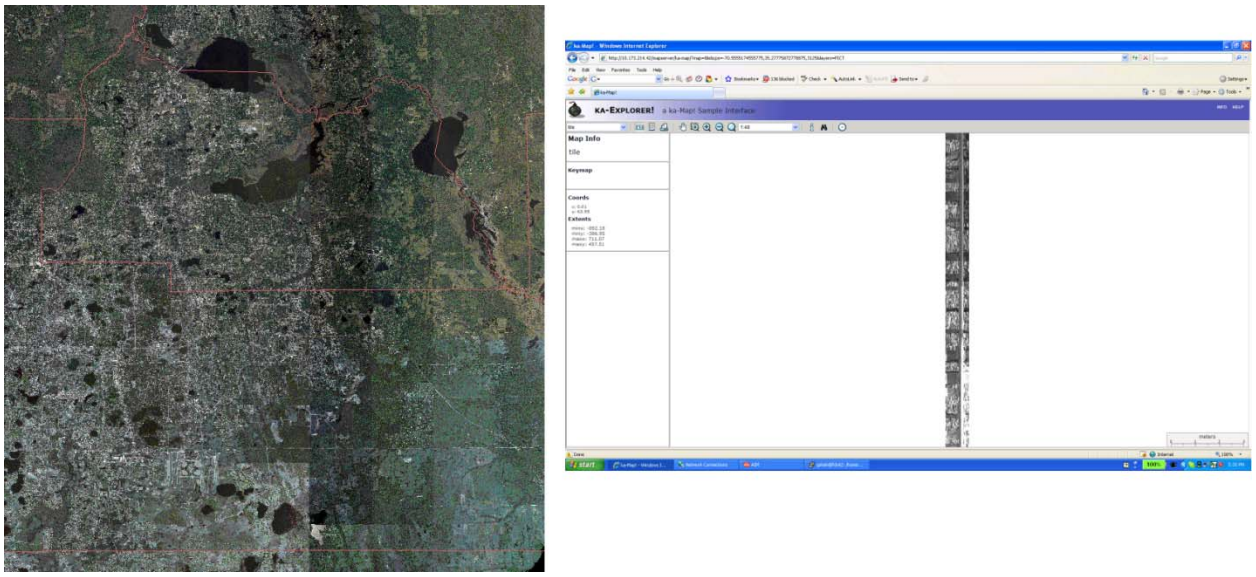


Figure 25: The LABINS Project and Web Interface through Ka-Maps.

Logging

When a user performs a run, the system logs positional and visual data onto the on-board server. Researchers calculate the positional data from the optical encoders that give the local position of the Hi-Rail vehicle for reference and an accurate linear position on the tracks. The visual data consists of four cameras: two standard area scan GigE cameras and two line-scan cameras. These are captured in real-time and time-stamped with an encoder value, with the possibility of incorporating a global positioning system (GPS) to be used for later post-processing. When the user knows the initial location of the run, this data log can be overlaid on a map of the state for searching and quick reference. When problems do arise, the logged points are color coded on the map depending on the fault.

Image Viewer Development

Currently, the setup researchers are using to acquire images results in capturing about 2000 pixels per line of an image. Researchers expect the FDOT Hi-Rail vehicle to run a minimum of 5 miles during each inspection. During this 5 mile run, the system continuously acquires lines to form a single large image that the algorithms need to load and process. At a speed of 20 mph, each tick of the encoder covers 0.0754 inches. Each of these ticks corresponds to a line that is 2000 pixels wide in the image. Over five miles, this would result in nearly 4.2 million lines. The size of the entire image will therefore be at least 16 Gigabytes. Currently, there is no image viewer that can handle such a large file size. It is therefore required to dynamically load in a finite set of lines, and scroll through the image at the end of the acquisition. Researchers have therefore implemented and created a custom viewer as an application package to be redistributable.

Authors show a screenshot of this viewer in Figure 26 below. The image viewer has the following features:

- Display a histogram of the portion of the image being viewed.
- Capabilities to manually scroll upwards and downwards in the image.
- Automatic scrolling feature to view the image.
- User can choose the rate at which the image is viewed and the lines that are being viewed.
- Display the lines that are currently being loaded into memory dynamically.

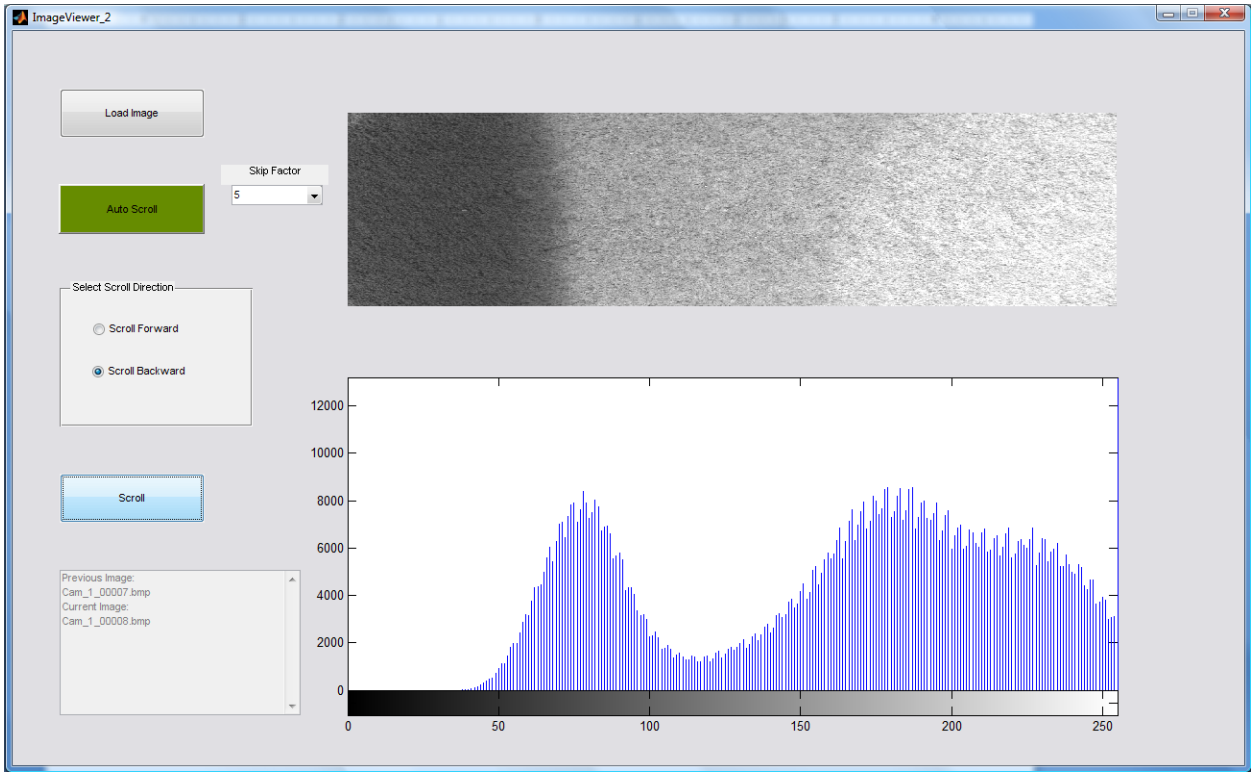


Figure 26: Screenshot of Mega-size Image Viewer.

APPENDIX B - Mobile Simulation Platforms

In order to acquire test-data for developing Computer Vision Algorithms, researchers designed a railroad cart in the Computer Vision Laboratory at UCF. This cart was created specifically for this project and was equipped with two fast-speed digital cameras, two SICK laser scanners, two 'line lasers' (laser which projects a line on a surface of the rail) as well as an encoder and GPS systems. The cart was constructed to be able to run on real railroad tracks to acquire data. An image of the cart is shown in Figure 27.

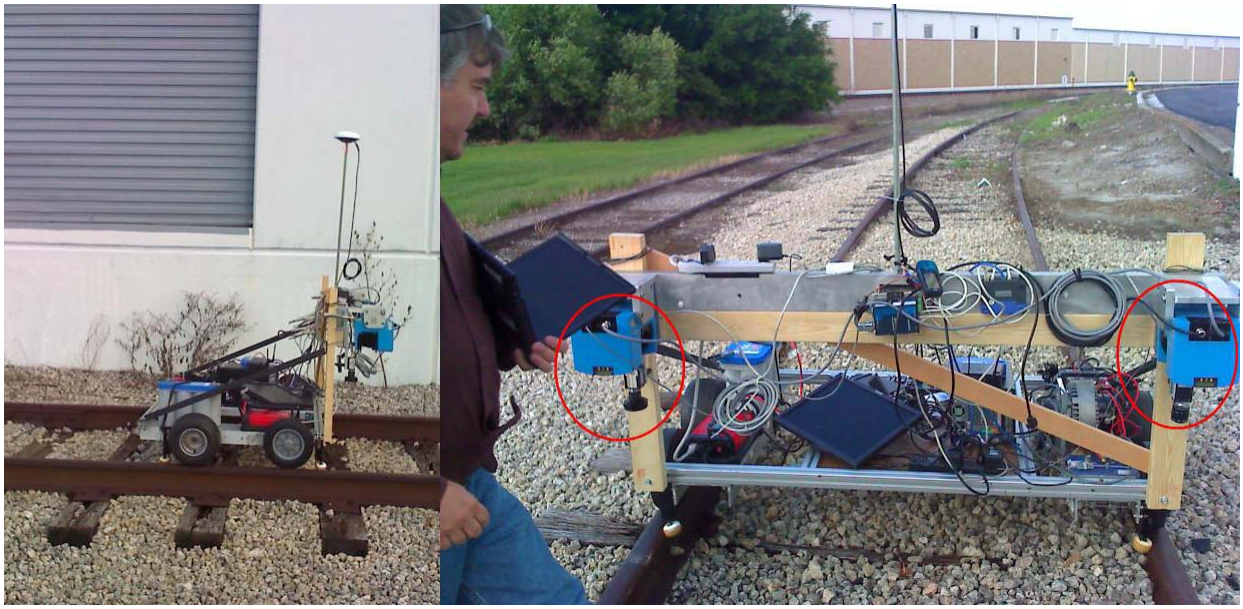


Figure 27: The Mobile Simulation Platform for acquiring data.

Cameras and both laser systems were installed directly over each rail; one camera, line laser and scanner over each rail. Therefore, as the cart moved, researchers obtained data synchronously for both left and right rails. After pre-processing and calculations this data allowed the researchers to find the rail gage in real time.

APPENDIX C - Computer Vision Terminology [12, 13]

Blurring: This is the process by which predominantly high frequency components in the image are removed, and is analogous to 'Smoothing'.

Classification: This process often involves steps that include probabilistically determining the category of a certain characteristic feature and applying a decision based on it.

Clustering: This is a method of unsupervised learning, where the number of clusters in the data is unknown, and the goal is to find the number of clusters that can accurately represent the data using only unlabelled examples.

Color moments: These are measures that are used to differentiate images based on their color features and are developed on the basis that the color in an image can be interpreted as a probability distribution. They provide a measure of the color similarity between images.

Computer Vision: This is a subject area that deals with automatic identification and analysis of images, for purposes of control or quantification. It is different from 'Image Processing' in the sense that it deals with not only computation, but also reasoning for the purposes of interpretation.

Connectivity: This applies to binary (black/white) images in which pixels are assigned 'labels' and a path can be found between two similar pixels (labels) by moving along adjacent similarly labeled pixels.

Convolution: This is a linear operator which involves constructing the output image by computing each pixel as a weighted sum of a local region of the input image with a fixed array or 'convolution mask'.

Edges: Extracting the scene 'structure' from a 2D image forms a major task in Computer Vision. The only 'features' that are preserved in an image are large differential discontinuities. i.e., places where there is a large isolated 'gradient' in the image data. These discontinuities are referred to as edges. Several methods including gradients, Laplacian, zero-crossing and morphological operations can be used to extract the edges. The first stage of the process is enhancement which involves generating an image in which ridge correspond to statistical evidence for an edge. It is achieved using linear convolution operators named after their inventors such as 'Roberts', 'Prewitt', 'Canny' etc.

Feature: A 'localizable' component of an image that has a set of measurable characteristics.

FIR Filtering: This is a method used in digital filtering of a signal and stands for 'Finite Impulse Response' filter. It is usually implemented using delays, multipliers and adders so that it is possible to operate on prior input samples. In simpler terms, it produces a weighted average of its most recent 'N' samples.

Gaussian Filtering: This involves image convolution with a 2D Gaussian 'kernel' to remove noise or high frequency components from an image.

Gradient: This is the first derivative or difference in pixel values along a particular direction in the 2D image.

Grey and Binary Levels: Images are represented by pixels, each of which has an integer value associated with it. Depending on whether the image is 6-bit, 8-bit or 10-bit, the corresponding integer values of the pixels range from 0-64, 0-256, or 0-1024 respectively. These are grey-level images. In a binary image, a single bit is used per pixel, where 1 indicates the presence of a feature, while 0 indicates the absence of the feature. For example, edges.

Homography: The homography defines a relation between two different planes such that every point in one plane can be correspondingly mapped onto the same point in the other plane.

Hough Transform: This is a method that is used to select a set of parametric values from a model which best describes a set of data. It works by looking for 'peaks' in the histogram across the parameter space and is a method based on probability and statistical theory.

Image Segmentation: This is the process of assigning classification groups to an image on a pixel-pixel basis in order to isolate particular structural regions in the image. This is often used as a pre-processing stage to aid high level interpretation such as recognition or classification.

K-Means: This is a method of clustering in which 'n' observations are clustered into 'k' partitions where each observation belongs to the cluster with the nearest mean.

Kalman Filter: This is a statistical process that estimates an optimal set of parameters based on the log-likelihood of the solution estimates in a sequential fashion. An optimal solution is estimated using data accumulated up to that point, following which an update equation is used to include successive points based on covariance of the parameters.

Laplacian: This is a differential operator used as a blob detector for tasks such as object recognition and texture analysis.

Localization: This is the process of defining a single-coordinate and sometimes, the spatial extent of a particular feature in an image.

OT-MACH/MACH filter: The Optimum Trade-off Maximum Average Correlation Height filter uses several training examples of the target to generate a single image template that can be used to detect future occurrences of the target.

RGB Color Space: The RGB color space is an additive color space based on the red, green, blue primary colors where chromaticity can take any value in between depending on their additive values.

Region Growing: This is an intermediate stage which involves merging regions with similar characteristics in order to obtain a simpler and more meaningful interpretation of the data.

SIFT: The Scale Invariant Feature Transform (SIFT) is an algorithm that is invariant to scale, rotation and translation and used to detect and describe local features in an image.

Support Vector Machine (SVM): This is a supervised learning method that is used for classification. Given a set of labeled examples, a Support Vector Machine training algorithm builds a model that predicts whether a new example falls into one category or another. An SVM essentially builds a hyperplane in high dimensional space that can be used for classification.

Texture Analysis: This refers to the spatial distribution of image intensities and can be described in terms of uniformity, density, coarseness, roughness, intensity, regularity and directionality.