The University of Manchester
School of Computer Science
MSc in Advanced Computer Science

Automatic Detection of Objects of Interest from Rail Track Images

Background report

Yohann Rubinsztejn

Supervisor - Ke Chen

May 9, 2011

Contents

Abstract

Rail inspection is an essential task in railway maintenance. It is periodically needed for preventing dangerous situations and ensuring safety in railways. At present, this task is operated manually by a trained human operator who periodically walks along the track searching for visual anomalies. This manual inspection is lengthy, laborious and subjective.

This project presents a new vision-based technique to automatically detect the presence or absence of parts of interest in rail tracks. This inspection system uses real images acquired by a digital line scan camera installed under a train. Data are processed according to a combination of image processing and pattern recognition methods to achieve high performance automated detection.

To date, we have attempted to apply the Viola-Jones object detection framework [23] to achieve automatic detection of rail track parts. The preliminary results are encouraging, revealing the presence of a particular kind of fasteners with an accuracy of 98%. Furthermore, we investigate a number of pre-processing and post-processing methods that may improve performance in terms of both detection accuracy and computation time.

Chapter 1

Introduction

Rail inspection consists in examining rail tracks for flaws that could lead to track failures and derailments. It is a crucial task in railway maintenance, and is periodically required in order to prevent dangerous situations. This task is usually operated manually by a trained human operator who periodically walks along the track searching for visual anomalies. This manual inspection is lengthy, laborious and subjective, since it relies entirely on the ability of the observer to detect possible anomalies.

With increased rail traffic carrying heavier loads at higher speeds, rail inspection is becoming more important and railway companies are interested in developing fast and efficient automatic inspection systems.

In the last decade, since computer vision systems have become increasingly powerful, smaller and cheaper, automatic visual inspection systems have become a possibility. These are especially suitable for high-speed, high-resolution and highly repetitive tasks. A large variety of algorithms for object detection problems have been studied by the computer vision community, especially for industrial inspection process. However, few works can be found on the use of computer vision in the specific area of rail inspection.

In this project, we propose to develop an effective vision-based automatic rail inspection system. The objective of this system is to detect the presence or absence of parts of interest in rail tracks, such as sleepers or fasteners, by inspecting real images acquired by a digital camera installed under a diagnostic train. The novelty of this work is the use of new learning algorithms (such as Viola-Jones object detection [23]) for visual pattern recognition in a rail inspection system.

The rest of this background report is structured as follows:

- Chapter 2 presents the background of this project. It outlines the motivation for this project, and also provides an overview of the state-of-the-art in the areas of rail inspection and object detection.
- Chapter 3 identifies the research methods involved in this project. It describes the methodology used to achieve the objectives, includes some preliminary results obtained by our inspection system, describes the deliverables of this project, and also provides the project plan that we try to follow.

Chapter 2

Background

This chapter describes the main objectives of this work and addresses typical issues involved in rail inspection. This also provides an overview of existing systems in the areas of rail inspection and object detection.

The rest of this chapter is organized as follows:

- Section 2.1 presents the motivation for this project.
- Section 2.2 provides an overview of the state-of-the-art in the areas of rail inspection and object detection.

2.1 Motivation

Manual monitoring for rail inspection is unacceptable for slowness and lack of objectivity. Nowadays, railway companies over the world are interested in developing automatic inspection systems that are able to detect rail defects. These automatic systems are to increase the ability to detect defects and reduce the inspection time.

The aim of this project is to develop an effective vision-based automatic rail inspection system, which is able to automatically detect the presence or absence of parts of interest in rail tracks. This system should be able to detect various objects such as sleepers or fastening elements (such as bolts, insulated block joints, clamps or clips) by inspecting the images acquired by a digital camera installed under a diagnostic train.

The problem of object recognition from 2-D images has been largely studied by the scientific community. Traditional object recognition methods include geometrical approaches, involving the use of rigid geometric models to represent the object to detect. However, railways represent a very rough environment and these methods do not succeed reliably in detecting objects of interest under varying conditions. Significant variety in lighting, viewing directions, sizes or shapes poses challenging problems and actually makes these objects difficult to model. Moreover, these methods usually require a human operator for tuning the parameters of the geometric models.

Other approaches include statistical learning techniques. These approaches involve the use of training sets to automatically learn a classification function that will be able to classify image subwindows and therefore detect the searched objects. These methods are suitable for generic shapes since they assume no geometrical model knowledge of the searched object.

These latter approaches provide enabling techniques to build up an effective automatic vision-based system for rail inspection. The next section describes in more detail the state-of-the-art techniques in the areas of rail inspection and object detection.

2.2 State of the art

This section provides an overview of the state-of-the-art in the areas of rail inspection and object detection.

The rest of this section is organized as follows:

- Section 2.2.1 covers existing systems in the area of rail inspection.
- Section 2.2.2 covers existing systems in the area of object detection.

2.2.1 Rail inspection

Two wide groups of analysis techniques can be used in industry to evaluate the properties of a material: *destructive techniques* and *non-destructive techniques*. Unlike destructive techniques, non-destructive techniques can identify deficiencies in a material without causing damage. In the area of rail inspection, traditional methods include destructive techniques, such as coring, and non-destrutive techniques, such as hammer sounding. Because of their limited effectiveness and the limited area covered by these techniques [1], further non-destructive techniques have been recently developed.

(a) An ultrasonic flaw detector with a combined probe for manual inspection

(b) An image acquisition system installed under an rail inspection car for automatic visual inspection

Figure 2.1: Two rail inspection techniques

These techniques include:

- Ultrasound inspection
- Magnetic methods, such as eddy current inspection, magnetic particle inspection (MPI), magnetic induction, magnetic flux leakage (MFL), electromagnetic acoustic transducer (EMAT)
- Ground penetrating radar (GPR)
- Laser light inspection
- Infrared inspection
- X-ray inspection
- Spectral analysis of surface waves (SASW)
- Impact-echo techniques
- Impulse-response techniques

Sato *et al.* [2] use ultrasonic sensors for obstruction detection. Kantor *et al.* [3] employs a laser light stripe to generate a 3-D profile of the railroad surface, and a ground penetrating radar to obtain subsurface measurements. Weil [4] combines a ground penetrating radar with infrared imaging systems to detect subsurface defects in railroad track beds.

These techniques rely on the use of specific devices, such as probes and transducers. These devices can be used on a hand pushed trolley, or in a hand held setup (Figure 2.1a). These devices are used to inspect small sections of track at precise locations. They are considered very slow and tedious, when there are thousands of miles of track that need inspection.

Visual inspection is another non-destructive technique. Unlike these previous techniques, visual inspection do not need specific devices. It uses a simple camera to acquire real images of tracks (Figure 2.1b). Thus, this technique relies to a big extent on classification algorithms in order to detect parts of interest. At present, visual inspection systems are typically used to measure rail profile [5], [6].

Rail inspection cars have been created in order to automate the analysis of railroad data and to answer to today's high mileage inspection needs. They are basically their own train with inspection equipment on board. The devices (probes, transducers or cameras) are mounted on carriages located underneath the inspection car. These inspection cars are loaded with high speed computers using advanced programs which recognize patterns and contain classification information. Systems capable of recording track geometry have been developed for railroad cars [7] and high-rail vehicles [8].

2.2.2 Object detection

Inspection devices, such as sensors or cameras, measure a physical quantity that can be represented by a signal. In particular, visual inspection use cameras to acquire real images. In order to achieve the automatic detection of parts of interest, missing elements or defects, captured images must be processed by pattern recognition algorithms.

This section describes the principles of these algorithms and outlines the main approaches to achieve object detection.

Basic principles of object detection

The objective of object detection is to identify, in the captured images, image areas (*subwindows*) that contain the patterns to be detected. To reach this goal, a basic method consists in exhaustively sliding a subwindow on a captured image. Data contained in each scanned subwindow are preprocessed with a feature extraction algorithm, and then provided to a classifier.

Figure 2.2 shows a real image of rail track acquired by a digital line scan

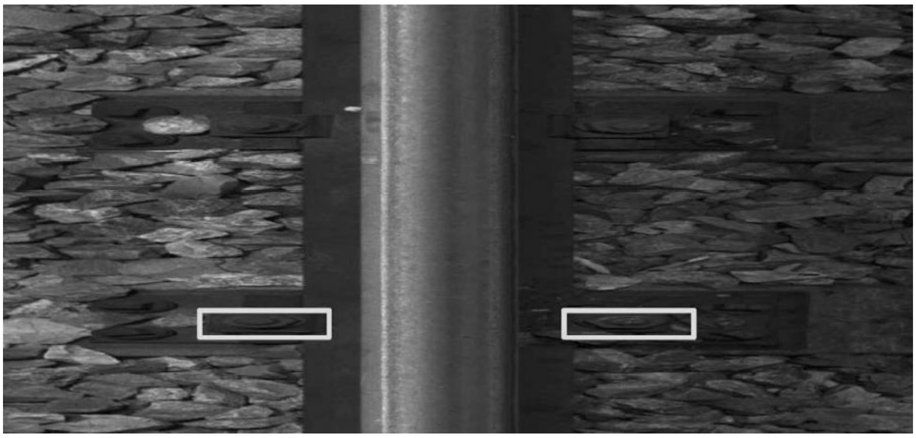

Figure 2.2: Locations of two detected fastening bolts, in a rail track image.

camera. The two subwindows show the locations of detected fastening bolts.

Feature extraction consists in reducing the size of the data that will be processed by the classifier, while revealing important information about these data. Classification consists, from the data preprocessed by feature extraction, in classifying each scanned subwindow as containing a pattern to be detected or not.

Therefore, different object detection algorithms differs in the choice of a feature extraction algorithm and a classifier. Two wide groups of approaches are usually used for object detection: geometrical approaches and statistical learning approaches.

Geometrical approaches

Geometry-based techniques require to build up a geometric model (*template*), or a set of handcrafted parameterized curves, to represent the object to detect. Usually, image processing techniques such as edge detection, border following, thinning algorithm, straight line extraction or active contours, are the low-level processes to prepare the data (the subwindow) to classification. Classification consists in matching these preprocessed data to the predefined template.

The commercial vision systems [9] and [10] use geometrical approaches to pattern recognition to detect rail defects. Singh *et al.* [11] use image processing methods, such as edge detection and colour analysis, to detect missing clips. Deutschl *et al.* [12] use convolution filters and morphological image analysis to detect rail surface defects. Lin *et al.* [13] adopt geometrical analysis directly on a gray-level histogram curve of the smoothed rail head surface image to detect Rolling Contact Fatigue (RCF) defects.

These approaches are difficult to extent to complex objects, since they involve a significant amount of prior information and domain knowledge to build up a geometric model. These systems are likely to suffer from restrictive assumptions on the scene structure. They are difficult to apply to rail inspection because railways represent a rough environment in which variety in lighting and texture poses challenging problem for object modeling.

Statistical learning approaches

The specificity of learning-based approaches is the use of a training set of data, whose actual class (label) is known a priori. This training set is presented to the classifier, which automatically adjusts its internal parameters to minimize some measure of the error between the estimated class and the actual label. Learning-based methods avoid difficulties in modeling objects by considering examples of that object under various conditions. Thus, a first human contribution is needed to build up a training set of data, in which each data item is assigned a label.

Since the last decade, Distante, Stella *et al.* [14], [15], [16], [17], [18], [19] have made major contributions for vision-based automatic rail inspection using learning-based approaches.

[14] preprocesses the data with a Gabor filter and classifies with an adaptive Self Organized Map (SOM) in order to detect rail defects on the rolling surface.

[15] compares two types of neural network classifiers, a Multilayer Perceptron (MLP) and a Radial Basis Function (RBF), within the context of fastening elements recognition. The data are preprocessed by a combination of Daubechies Discrete Wavelet Transform (DDWT) and Principal Component Analysis (PCA) techniques.

[16] compares two preprocessing techniques, Independent Component Analysis (ICA) and Daubechies Discrete Wavelet Transform (DDWT), within the context of hexagonal-headed bolts recognition. A Support Vector Machine (SVM) is used for classification.

[17] compares three preprocessing techniques, which are Gabor filter, Discrete Wavelet Transform (DWT) and Gabor Wavelet Transform (GWT), within the context of corrugation (a particular class of surface defects) detection. A k-Nearest Neighbour (KNN) classifier and a Support Vector Machine (SVM) are used for classification.

[18] preprocesses the data with Principal Component Analysis (PCA)

and classifies with a Multilayer Perceptron (MLP) in order to detect and track the rail head.

[19] preprocesses the data with Daubechies and Haar Discrete Wavelet Transform (DDWT and HDWT) and classifies with two Multilayer Perceptron (MLP) neural classifiers in order to detect hexagonal-headed bolts.

Besides, general object detection frameworks have been developed [20], [21], [22], [23], and have been widely applied to the specific problem of face detection.

Papageorgiou *et al.* [20] preprocess the data using a Haar wavelet-like representation, which is used as an input to a Support Vector Machine (SVM) classifier.

Rowley *et al.* [21] preprocess the data through an extensive preprocessing stage (lighting correction, histogram equalization), and classify with retinally connected neural networks. The system arbitrates between multiple networks to improve performance over a single network.

Schneiderman and Kanade [22] use multiresolution information for different levels of wavelet transform. A nonlinear face and nonface classifier is constructed using statistics of products of histograms computed from face and nonface examples using AdaBoost learning [24]. This algorithm can detect profile views but is computationally expensive.

Viola and Jones [23] built a fast and robust object detection system in which AdaBoost learning is used to construct a nonlinear ('strong') classifier. AdaBoost is used to construct weak classifiers based on simple scalar Haar wavelet-like features, and boost them to construct a strong classifier. Viola and Jones make use of several techniques for fast computation of a large number of features. An attentional cascade of classifiers makes the computation even more efficient, allowing background regions of the image to be quickly discarded while spending more computation on object-like regions. Applied to face detection, their system is the first real-time frontal-view face detector.

To the best of our knowledge, there are no references in the literature to the use of these relatively recent object detection frameworks for an automatic vision-based rail inspection system. Nevertheless, these methods may prove to be efficient to detect the presence of objects of interest in rail tracks. Therefore, we propose in this work to build up an automatic vision-based rail inspection system based on one of these frameworks. We will focus on the Viola-Jones object detection framework, which is relatively easy to implement and because the preliminary results obtained with it are encouraging.

The next chapter describes in detail the methods we use to implement this framework and achieve an efficient automatic vision-based rail inspection system.

Chapter 3

Research methods

This chapter presents the research methods involved in this project. This describes in detail our methodology to reach our objectives. This also includes some preliminary results, describes the deliverables of the project, and provides a project plan.

The rest of this chapter is organized as follows:

- Section 3.1 describes the methodology used in this project to achieve an efficient automatic vision-based rail inspection system.
- Section 3.2 presents some preliminary results obtained by our prototype.
- Section 3.3 defines the deliverables of the project.
- Section 3.4 describes the project plan that we try to follow.

3.1 Methodology

This section covers the methods that will allow to achieve an efficient automatic vision-based rail inspection system. This shows how we construct an appropriate training dataset, how we learn a classifier that can detect parts of interest, and describes several pre-processing and post-processing techniques that can be used to improve detection performance. Some concepts of this methodology will raise issues that we will try to address.

The rest of this section is organized as follows:

- Section 3.1.1 describes how we construct the training dataset that will be used by the learning algorithm.
- Section 3.1.2 covers the methods involved in the learning algorithm.
- Section 3.1.3 covers further pre-processing and post-processing techniques.

3.1.1 Constructing the training dataset

Statistical learning methods use a training dataset to learn a classifier. In the context of object detection, training data are subwindows, which contain the pattern that must be classified. Subwindows that represent an object to detect are called positive data. Subwindows that do not are called negative data. An appropriate training dataset must contain both positive data and negative data.

In our work, only real images acquired by a digital camera are given. These real images contain large parts of rail tracks. Objects of interest, such as fastening elements, are contained in some small subwindows of these real images. In order to construct an appropriate dataset, some of these subwindows must be manually extracted from the real images, and labeled as positive data. Some other subwindows that do not contain an object to detect must also be manually extracted, and labeled as negative data.

An efficient learning algorithm requires a training data of hundreds or thousands of fixed-size subwindows that represent positive and negative data under various conditions. This poses a challenging issue. Indeed, manually extracting and labeling that much data is lengthy. To reduce this difficulty, we can use an algorithm that randomly extracts subwindows from the real images, so that we can manually select and label them. Considering that real images usually contain much more negative data than positive data, this method is useful only to construct negative data. Manually constructing positive data is inevitable.

Another issue concerns the use of different types of objects of interest (positive data) in the training dataset. If different types of objects, such as different types of fasteners, need to be detected, examples of such objects need to be included in the training dataset. There are two main ways to include different types of objects of interest in the training dataset.

The first way is to make no distinction between these different objects. Thus, for the construction of the training dataset, such different types of objects are labeled equivalently as positive data. By increasing the intra-class

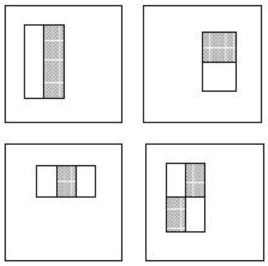

Figure 3.1: Four Haar wavelet-like features shown relative to their enclosing subwindows. The sum of the pixels which lie within the white rectangles are subtracted from the sum of the pixels in the grey rectangles.

variability of the positive data, this method may compromise the discriminative power of the trained classifier.

The second way is to divide the training dataset into several training datasets, each of them containing its own positive data. Thus, one classifier would be trained from each training dataset. Then, the trained classifiers must be combined (for example in a logical OR) to produce the output of the classification. By restricting the intra-class variability of the positive data, this method may compromise the generalization power of the trained classifiers.

The objective is to maximize the accuracy of the classification by making a trade-off between its discriminative power and its generalization power. When several types of objects of interest can be clearly distinguished by the human eye, it is preferable to use the second method so as to restrict the intra-class variability of the positive data.

3.1.2 The learning algorithm

This section describes the methods, introduced by Viola and Jones [23], to learn a classifier that can automatically recognize objects of interest. This presents the Haar wavelet-like features used to represent a data, the AdaBoost learning algorithm used to select features and construct a classifier, and the cascade of classifiers used to improve the performance of the classification. We will try to address any issue that will be raised within the context of our rail inspection system.

Haar wavelet-like features

Feature extraction consists in extracting certain properties of the data that will be processed by the classifier. The main purpose of using features rather than the raw data (raw pixel values) lies in the fact that features can encode knowledge about the data, which is difficult to learn from the raw data. Thus, feature extraction helps increase the generalization power of the classification. The other motivation for using features is to reduce the size of the data that will be processed by the classifier, thus reducing the computation time of the classification.

Viola and Jones [23] propose four basic types of scalar features, called Haar wavelet-like features. These features are reminiscent of Haar wavelets, which have been developed for basis functions to encode signals. The objective of these features is to collect local oriented intensity information at different scales and locations for representing image patterns. These features can be represented by rectangular blocks located in subregions of a subwindow. These rectangular blocks can vary in shape (aspect ratio), size, and location inside the subwindow. The value of a Haar wavelet-like feature is the difference between the sum of the pixels within rectangular regions of these blocks. Figure 3.1 shows some examples of the four basic Haar wavelet-like features. For a subwindow of size 24×24 pixels and using the four basic types of Haar wavelet-like features, a total of approximately $160,000$ features can then be constructed.

The Haar wavelet-like features are interesting for two reasons. First, powerful classifiers can be constructed based on these features. Second, they can be computed efficiently using the integral image technique, also introduced by Viola-Jones [23].

For a rail inspection system, we may modify or extend the four basic types of Haar wavelet-like features introduced previously. In order to improve the accuracy of the classification, new appropriate Haar wavelet-like features may be designed to fit the patterns of the objects of interest. To do so, false negative results of the classification using the basic types of Haar wavelet-like features must be investigated. Thus, appropriate features may be designed by trying to adapt the features to the patterns contained in these false negative results. Similar work has been carried out within the context of face detection, which led to the creation of an extended set of Haar wavelet-like features [25].

AdaBoost learning

The AdaBoost learning procedure [24] is aimed at constructing a nonlinear 'strong' classifier from a sequence of best weak classifiers. AdaBoost is used to:

1. select effective features from a large feature set (recall that there are over $160,000$ features associated with each subwindow)
2. construct weak classifiers, each of which is based on one of the selected features
3. boost the weak classifiers to construct a strong classifier

The AdaBoost learning procedure is summarized in Figure 3.5 (Appendix A). At each round of the procedure, a weak classifier is constructed by thresholding the value of a feature at an optimal threshold value. At each round, finding the best weak classifier, which minimizes the weighted classification error, requires to construct every possible weak classifier (as many as the number of features) and find the best one. This process is the most time-consuming part of the training procedure. After T rounds, the T best weak classifiers, based on T features, are combined to construct a strong classifier.

The inputs of the AdaBoost learning procedure need to be investigated. The number N of training examples, the size of the training subwindows, and the number T of rounds must be determined. Increasing these parameters improves the accuracy of the final classifier but also increases the computation time of the training procedure. Therefore, these parameters must be adjusted to result in a high enough accuracy, while keeping the training stage computationally feasible.

Other boosting variants may be investigated. In real versions of AdaBoost, such as RealBoost [26] and LogitBoost [27], weak classifiers are real-valued or output the class label with a probability value. Less aggressive versions of AdaBoost, such as GentleBoost [27], may be preferable in dealing with training data containing outliers. FloatBoost [28] incorporates the idea of floating search into AdaBoost. It backtracks and examines the already selected features to remove those that are least significant. It is also more computationally expensive.

Cascade of classifiers

A boosted strong classifier effectively eliminates a large portion of negative subwindows while maintaining a high detection rate. Nonetheless, a single

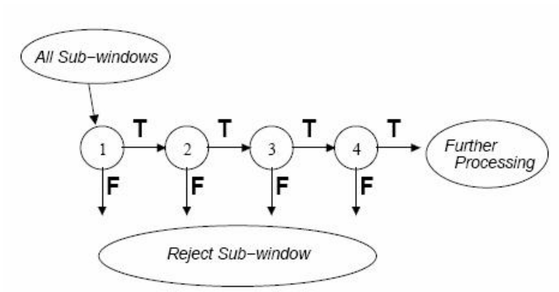

Figure 3.2: Cascade of classifiers

strong classifier may not meet the requirement of an extremely low false positive rate. A solution is to arbitrate between several strong classifiers, for example using a logical AND. This leads to the concept of a cascade of strong classifiers, as illustrated in Figure 3.2. Each subwindow that fails to pass a strong classifier is not further processed by subsequent strong classifiers. At each stage in the cascade, the threshold of a strong classifier can be adjusted to minimize false negative rate. The motivation behind the cascade of classifier is that simple classifiers at early stage can filter out most negative examples efficiently, and stronger classifiers at later stage are only necessary to deal with instances that are likely to be positive. Taking into account that most of the subwindows in a real image are negative, this strategy can significantly speed up the detection and reduce false positives, with a little sacrifice of the detection rate.

The implementation of a cascade of classifiers requires to adjust some parameters: the number of classifier stages, the number of weak classifiers (or features) used to boost a strong classifier in each stage, and the threshold of the strong classifier in each stage. In practice, we will manually define, at each stage, targets for detection rate and false positive rate. The parameters will then be adjusted to meet these targets, by testing the classifier on a validation dataset.

3.1.3 Further pre-processing and post-processing

This section presents some image pre-processing and post-processing techniques that can be used to improve detection performance. Pre-processing techniques are applied to the real images and subwindows before detection. Post-processing techniques are applied to the real images and subwindows after detection.

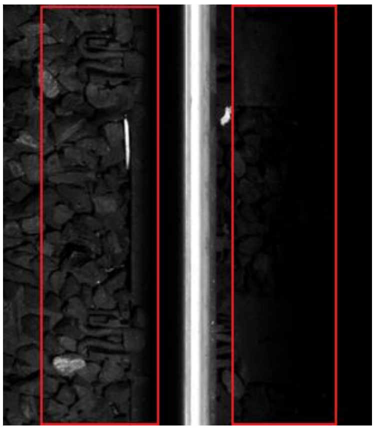

Figure 3.3: A real image of rail track and two regions of interest for fastening element detection.

Pre-processing techniques

The first pre-processing technique is to rescale the real images of the rail track, in which we want to apply our detector. Indeed, since the training data contain subwindows that are rescaled to a fixed size (for example 24×24 pixels), the test real images must be rescaled at the same ratio, so that the test subwindows will have the same size as the training subwindows.

The second pre-processing technique that we can apply is to crop the real images of the rail track. The objective is to make the detector search only within regions of interest, where objects to detect are present, in order to reduce the computation time of the detection. For example, fastening elements are located on the left and the right of the rail head. Thus, by detecting the horizontal position of the rail head by some image processing techniques, we can retrieve these two regions of interest and apply the detector only in these regions. In our real images, the rail head is highly illuminated compared to the rest of the image. By inspecting the mean value of the pixel intensities in each column of the image, it is then easy to retrieve its horizontal position. Figure 3.3 shows the two regions of interest for fastening element detection.

Other pre-processing techniques include variance normalization and histogram equalization. These techniques can help correct variations in lighting condition. However, they are computationally extensive. Therefore, the gain in the accuracy of the detection obtained by applying these techniques must be investigated to know if these techniques are worthwhile.

Post-processing techniques

Multiple detections will usually occur around each detected object. In order to return one final detection per object, it is useful to merge overlapping detection into a single detection. This is done by averaging the corners of all overlapping detection regions.

3.2 Preliminary results

This sections presents the preliminary results obtained with a basic implementation of the Viola-Jones object detection framework [23]. Our objective is to detect a specific type of fastening elements in real images of rail tracks.

The rest of this section is organized as follows:

- Section 3.2.1 presents the results relative to the training stage.
- Section 3.2.2 presents the results relative to the tests.

3.2.1 Training results

We constructed a training dataset containing 150 positive data and 300 negative data. The size of the training subwindows is 190×110 pixels, but they are resized to 19×11 pixels before applying the training algorithm. We only use the four basic types of Haar wavelet-like features introduced by Viola and Jones [23]. The number of rounds T used in the AdaBoost learning procedure is 20. Figure 3.6 (Appendix B) shows that the training error of the strong classifier approaches zero exponentially in the number of rounds.

3.2.2 Test results

We constructed a validation dataset of 50 positive data and 100 negative data. Figure 3.7 (Appendix B) shows the Receiver Operating Characteristic (ROC) of our detector, calculated by testing the strong classifier on the validation dataset. This curve shows that an ideal performance can be reached, with a detection rate of 100% and a false alarm rate of 0%. However, the threshold of the stong classifier have been automatically chosen, and results in a detection rate of 94%, a false positive rate of 0%, and an overall accuracy of 98%. These seemingly good results can be explained by the fact that our validation dataset contains very little data. Thus, unless constructing hundreds of data for a validation set, it is hard to evaluate precisely the performance of our detector. Moreover, these data have been manually extracted. Therefore, they cannot be considered as true random data.

In order to have an overview of the actual performance of our detector, further investigation must be made. The results obtained by the detector on real images must be observed. Figure 3.8 (Appendix B) shows the detections obtained on real images of a rail track. The first image actually reveals a false positive and a false negative.

3.3 Deliverables

This section defines the deliverables of the project. This includes the final software to be produced, and the documents to be provided.

The rest of this section is organized as follows:

- Section 3.3.1 presents the demo system that must be achieved.
- Section 3.3.2 presents the documents that must be delivered.

3.3.1 Demo system

In this project, a demo system with a graphical user interface (GUI) is expected to be developed. This system is expected to allow to train and test a detector on real images of rail tracks. The GUI should allow to modify easily the parameters of the training and test stages. Figure 3.4 shows a simple GUI that has been developed for our basic detector. Because it provides a suitable environment for scientific developments and experiments, MATLAB has been used to implement our basic system and its GUI.

3.3.2 Documents

This project will lead to a final dissertation that will describe the results of my work. We will also aim at publishing a short research paper describing the methods and results obtained by the final version of our automatic vision-based rail inspection system.

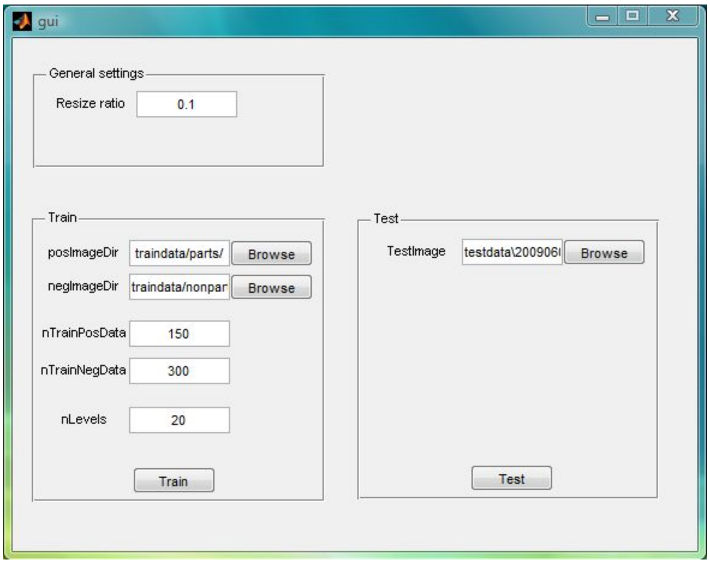

Figure 3.4: Simple GUI for the basic detector

3.4 Project plan

Figure 3.9 (Appendix C) shows the Gantt chart of the project.

To date, we have managed to build a basic version of an automatic vision-based rail inspection system. The next step is to enhance this system by investigating the methods discussed in this chapter. Specifically, we will try to:

- increase the number of training data and validation data,
- include other types of objects to be detected,
- extend the set of Haar wavelet-like features,
- find better input parameters for the AdaBoost learning algorithm,
- apply some boosting variants,
- implement a cascade of classifiers,
- implement further pre-processing and post-processing methods.

The objective of these enhancements will be to improve performance of the system in terms of detection accuracy and/or computation time.

Appendix A

Input

(1) Training examples $(x_1, y_1), ..., (x_N, y_N)$,
where x_i is a subwindow, and y_i its label ($y_i \in \{-1, +1\}$).

(2) The number T of weak classifiers to be combined.

Initialization

$D_1(i) = 1/N$, $i = 1, ..., N$

Repeat

For $t = 1, ..., T$

(1) Find the classifier h_t that minimizes the weighted error:

$$h_t = \arg\min_{h_t \in H} \epsilon_t, \text{ where } \epsilon_t = \sum_{i=1}^{N} D_t(i)[y_i \neq h_t(x_i)]$$

(2) Choose the coefficient α_t:

$\alpha_t = \frac{1}{2} \ln \frac{1-\epsilon_t}{\epsilon_t}$

(3) Update the distribution D_{t+1}:

$D_{t+1}(i) = D_t(i) \exp(-\alpha_t y_i h_t(x_i))$, $i = 1, ..., N$,

and normalize to $\sum_{i=1}^{N} D_{t+1}(i) = 1$.

Output

Final (strong) classifier: $H(x) = sign\left(\sum_{t=1}^{T} \alpha_t h_t(x)\right)$

Figure 3.5: AdaBoost learning procedure

Appendix B

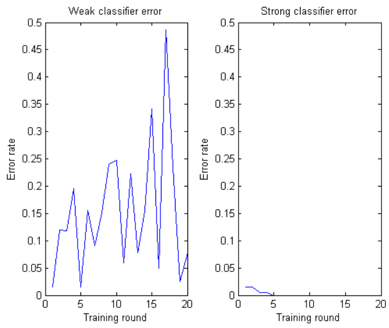

Figure 3.6: Training errors: weak classifier error and strong classifier error.

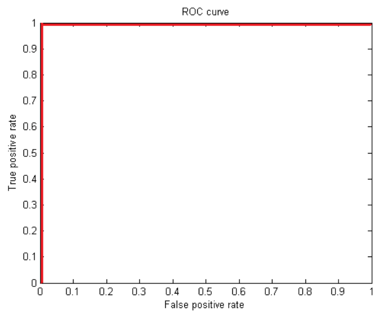

Figure 3.7: Receiver Operating Characteristic (ROC)

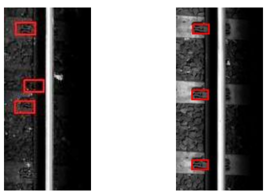

Figure 3.8: Application of the detector on two real images of a rail track

Appendix C

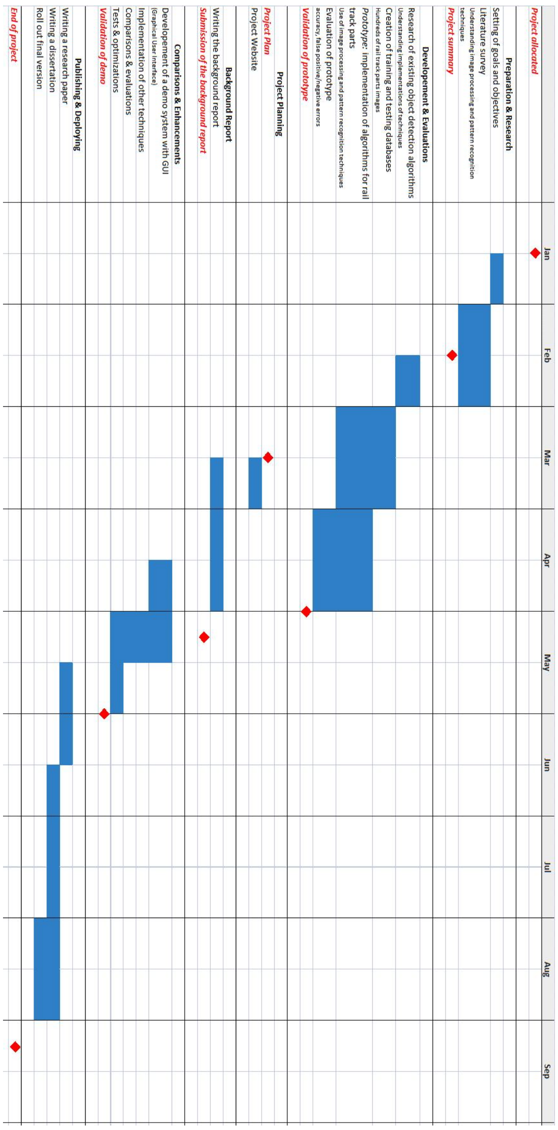

Figure 3.9: Gantt chart of the project

References

[1] N. Delatte, S. Chen, N. Maini, N. Parker, A. Agrawal, G. Mylonakis, K. Subramaniam, A. Kawaguchi, P. Bosela, S. McNeil, R. Miller, "The application of non-destructive evaluation to subway tunnel system," *TRB 2003 Annual Meeting*, 2003.

[2] K. Sato, H. Arai, T. Shimizu, M. Takada, "Obstruction detector using ultrasonic sensors for upgrading the safety of a level crossing," *Proc. IEE Int. Conf. Developments in Mass Transit Systems*, London, Apr. 1998, pp. 190-195.

[3] G. Kantor, H. Herman, S. Singh, J. Tabacchi, W. Kaufman, "Automatic railway classification using surface and subsurface measurements," *Proc. Int. Conf. Field and Service Robotics*, pp. 43-48, 2001.

[4] G.J. Weil, "Non-destructive, remote sensing technologies for locating subsurface anomalies on railroad track beds," *Proc. SPIE, Nondestructive Evaluation of Aging Railroads*, vol. 2458, pp. 74-81, 1995.

[5] D.L. Magnus, "Non-contact technology for track speed rail measurement (ORIAN)," *Proc. SPIE, Nondestructive Evaluation of Aging Railroads*, vol. 2458, pp. 45-51, 1995.

[6] G.S. Bachinsky, "The electronic BAR gauge (a customized optical rail profile measurement system for rail grinding applications)," *Proc. SPIE, Nondestructive Evaluation of Aging Railroads*, vol. 2458, pp. 52-63, 1995.

[7] W. Ebersohn, E.T. Selig, "Use of geometry measurements for maintenance planning," *Transportation Research Record No. 1470, Railroad Research Issues*, pp. 84-92, 1994.

[8] A.M. Zarembski, W.T. McCarthy, "Development of nonconventional tie and track structure inspection systems," *Transportation Research Record No. 1489, Railroad Transportation Research*, pp. 26-32, 1995.

[9] Cybernetix Group (France), "IVOIRE: A system for rail inspection," internal documentation, [Online]. Available: http://www.cybernetix.fr

[10] Benntec Systemtechnik Gmbh, "RAILCHECK: Image processing for rail analysis," internal documentation, [Online]. Available: http://www.benntec.com

[11] M. Singh, S. Singh, J. Jaiswal, J. Hempshall, "Autonomous Rail Track Inspection using Vision Based System," *IEEE Int. Conf. Computational Intelligence for Homeland Security and Personal Safety*, Alexandria, USA, Oct 16-17, 2006.

[12] E. Deutschl, C. Gasser, A. Niel, J. Werschonig, "Defect detection on rail surfaces by a vision based system," *IEEE Intelligent Vehicles Symposium*, pp. 507-511, Jun. 2004.

[13] J. Lin, S. Luo, "Real-time Rail Head Surface Defect Detection: a Geometrical Approach," *IEEE Int. Symp. Industrial Electronics*, pp. 769-774, 2009.

[14] M. Nitti, C. Mandriota, C. Distante, E. Stella, "Rail defect classification by adaptive self organized map," *SPIE Int. Symp. Intelligent System and Advanced Manufacturing, Part of Photonics*, Boston, 2001.

[15] E. Stella, P.L. Mazzeo, M. Nitti, G. Cicirelli, A. Distante, T. D'Orazio, "Visual recognition of missing fastening elements for railroad maintenance," *Proc. IEEE-ITSC Int. Conf. Intell. Transp. Syst.*, Singapore, 2002, pp. 94-99.

[16] P.L. Mazzeo, N. Ancona, E. Stella, A. Distante, "Visual Recognition of hexagonal headed bolts by comparing ICA to wavelets," *Proc. IEEE Int. Symp. Intelligent Control*, Houston, 2003, pp. 636-641.

[17] C. Mandriota, M. Nitti, N. Ancona, E. Stella, A. Distante, "Filter-based feature selection for rail defect detection," *Machine Vision and Applications, Springer-Verlag*, vol. 15, pp. 179-185, Oct. 2004.

[18] F. Marino, A. Distante, M. Nitti, E. Stella, "A real-time visual inspection system for railway maintenance: Automatic rail detection and tracking," *Politecnico di Bari, Italy, Intern. Rep. DEE*, 2005.

[19] F. Marino, A. Distante, P.L. Mazzeo, E. Stella, "A Real-Time Visual Inspection System for Railway Maintenance: Automatic Hexagonal-Headed Bolts Detection," *IEEE Trans. Systems, Man and Cybernetics-Part C*, vol. 37, no. 3, pp. 418-428, May 2007.

[20] C. Papageorgiou, M. Oren, T. Poggio, "A general framework for object detection," *Proc. IEEE Int. Conf. Computer Vision*, Bombay, 1998, pp. 555-562.

[21] H. Rowley, S. Baluja, T. Kanade, "Neural network-based face detection," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 20, pp. 22-38, 1998.

[22] H. Schneiderman, T. Kanade, "A statistical method for 3D object detection applied to faces and cars," *Proc. IEEE Computer Society Conf. Computer Vision and Pattern Recognition*, 2000.

[23] P. Viola, M. Jones, "Rapid object detection using a boosted cascade of simple features," *Proc. IEEE Computer Society Conf. Computer Vision and Pattern Recognition*, Hawaii, Dec. 2001.

[24] Y. Freund, R.E. Schapire, "A decision-theoretic generalization of online learning and an application to boosting," *Journal of Computer and System Sciences*, vol. 55, no. 1, pp. 119-139, Aug. 1997.

[25] R. Lienhart, J. Maydt, "An Extended Set of Haar-Like Features for Rapid Object Detection," *Proc. IEEE Int. Conf. Image Processing*, vol. 1, pp. 900-903, 2002.

[26] R.E. Schapire, Y. Singer, "Improved Boosting Algorithms Using Confidence-Rated Predictions," *Proc. 11th Ann. Conf. Computational Learning Theory*, pp. 80-91, 1998.

[27] J. Friedman, T. Hastie, R. Tibshirani, "Additive Logistic Regression: A Statistical View of Boosting," *The Annals of Statistics*, vol. 28, no. 2, pp. 337-374, Apr. 2000.

[28] S.Z. Li, Z.Q. Zhang, H.-Y. Shum, H. Zhang, "FloatBoost learning for classification," *Proc. of Neural Information Processing Systems*, Vancouver, 2002.