STATISTICS BASED NEURAL NETWORKS METHOD FOR INDUSTRIAL IMAGE INSPECTION

by

Yi Zhu
Bachelor of Computer Science
Beijing Institute of Technology
Beijing, P.R.China

A thesis
presented to Ryerson University
in partial fulfillment of the
requirement for the degree of
Master of Applied Science
in the Program of
Electrical and Computer Engineering.

Toronto, Ontario, Canada, 2005

Author's Declaration

I hereby declare that I am the sole author of this thesis.

I authorize Ryerson University to lend this thesis to other institutions or individuals for the purpose of scholarly research.

Yi Zhu

I further authorize Ryerson University to reproduce this thesis by photocopying or by other means, in total or in part, at the request of other institutions or individuals for the purpose of scholarly research.

Yi Zhu

Instructions on Borrowers

Ryerson University requires the signatures of all persons using or photocopying this thesis. Please sign below, and give address and date.

Name	Signature	Address	Date

Abstract

Statistics Based Neural Networks Method for Industrial Image Inspection

© Yi Zhu, 2005

Master of Applied Science
Department of Electrical and Computer Engineering
Ryerson University

Automated industrial image inspection system has attracted a great deal of interest in recent years. In this thesis, a new method is presented by combining a statistics method with a Neural Networks method, which could reduce the interference of machine dynamics and improve the inspection accuracy. Different from the pixel-based or feature-based methods, the proposed method is based on two indices of an image, which are the variances of the rows and columns of the image. For image inspection, first Neural Networks are trained using these two indices from a set of good images in order to establish a tolerance zone. Then, the two indices of each inspection image are computed through trained Neural Networks and compared with the tolerance zone. A defective item is detected if either index falls out of the tolerance zone. The other contributions, such as two-point based image registration method and defect simulation algorithms, also help to improve the performance of inspection. Experimental results demonstrate that the proposed approach has better performance in comparison with traditional statistics approach.

Acknowledgments

I would like to address my gratitude to all those who gave me the possibility to complete this thesis.

I am deeply indebted to my supervisors Dr. Bin Wu and Dr. Jeff Xi whose support, stimulating suggestions and encouragement help me all the time.

I also want to thank Rotoflex company who supports the research stipend, and the members of vision group whose suggestions are very helpful.

My classmates from the Department of Electrical and Computer Engineering support me in my research work. I want to thank them for all their help. Especially I want to thank Z.W.Yang, Q.Sun, J.Jin and K.Momen.

Contents

List of Tables

List of Figures

Acronyms

AI:	Artificial Intelligence
ANN:	Artificial Neural Networks
AVI:	Automated Visual Inspection
CCS:	Code Composer Studio
FE:	Feature Extraction
IA:	Image Acquisition
IE:	Image Enhancement
IIn:	Image Inspection
MLP:	MultiLayer Perceptron
NN:	Neural Networks
PCA:	Principal Component Analysis
SC:	Smart Camera

1. Introduction

1.1 Motivation and Solution

With the technology advancement over the last two decades, Automated Visual Inspection (AVI) systems are being used in various applications of industrial image inspection, such as automotive, manufacturing and medical [1]. Typically, an AVI system can be decomposed into a sequence of processing steps [2]: *image acquisition* to acquire images by a CCD camera; *image enhancement* to improve the quality of the image for subsequent processing; *defect recognition* to detect defects by comparing the current image with a template image; *defect classification* to classify a defect type by feature extraction and classification; and *decision making* to decide if the image should be judged as qualified or defective. Theoretically, this routine is feasible for most of inspection cases. However, problems rise when implemented in real industrial environment. One of the common problems is how to quantify the uncertainties that affect the image fidelity. These uncertainties may be caused by *machines dynamics*, *product variation*, and *illumination variation.*

Machine dynamics is referred to as the dynamic behaviour of the inspection machine running under operating condition. Vibration (shaking), speed variation, and slippage are the main factors that could distort the image of a product under inspection by AVI. These distortions may include image tilting, blurring and so on. If distorted, a qualified product could be wrongly detected as defective or vice versa, thereby degrading inspection accuracy. Therefore, consideration of machine dynamics is very important, especially when AVI is running at high speed.

Due to process variability, no products can be made perfectly identical. This is referred to as product variation. Because of this reason, products are designed with dimensional tolerance, geometry shape tolerance, and surface roughness requirement. The first two factors may cause images to tilt. Illumination variation is another factor affecting image reflectivity.

In traditional AVI systems, the operator sets the thresholds or tolerances to account for uncertainties based on his or her experience or through some trials without a thoughtful approach. As a result, the set-up process is inefficient and the inspection system is less accurate. In order to meet these stringent requirements, the adaptive and optimal strategies should be investigated.

Over the last two decades, artificial intelligence (AI) methods have been applied to train AVI systems to improve their efficiency and accuracy. Neural Networks (NN) method is one of the popular AI methods used for image processing [3-6], and its applications span from *data reduction/feature extraction* [7], *segmentation* [8], to *object detection/recognition* [9]. Because of its inherently nonlinear nature, NN is considered particularly well suited for image processing applications where the classical spatial or frequency methods are not effective [10]. NN-based methods are developed to complement or replace conventional approaches for industrial inspection [11].

As summarized in [12] and [13], applications of NN-based inspection include *quality and process control, document processing, identification and authentication,* and *medical diagnosis*. The first application is to detect defective industrial products, such as metals, textiles, as well as food products, such as fruits. The second application is to read machine-generated and hand-written texts for mail sorting or other processing of forms. The third application contains license plate recognition, fingerprint analysis and face detection. The last application is to screen certain diseases, such as cancer.

The processing efficiency of NN-based image inspection methods is largely determined by the abstraction level of the input data, which may be classified into three levels, *pixel, local feature* and *object*. In the first category, the intensities of individual pixels are provided as the inputs. In the second category, a set of derived, pixel-based features constitutes the inputs. In the last two categories, the properties of individual objects are used as the inputs. In most archived work, the input data is either pixel level or feature level, the methods based on which are usually computationally intensive and time consuming.

In this thesis, a new method is developed that can set a tolerance zone for image inspection through NN training. The proposed method combines a statistics method with a NN method. Different from traditional image inspection methods, this method is base on two indices of an image, which are the variances of the rows and columns of tł image. Instead of using entire pixels, Neural Networks are trained using these two indice from a set of sample images. The minimum and maximum values of the two indices forı a tolerance zone. The application of NN in this method is also different with tradition method. In most of cases, NN is just used as classifier directly. But with new methoc The NN is used as a filter to reduce the effect of uncertainties. When inspecting, the tw indices of each inspection image are computed through trained Neural Networks an compared with the tolerance zone. Experimental results show that defective items can b effectively detected by examining if either index falls out of the tolerance zone. In wha follows, the details of this method are presented.

1.2 Thesis Outline

This thesis is organized into six chapters:

Charter 1 is introduction chapter.

Chapter 2 describes the previous approach of AVI in industrial image inspection anc image inspection application of Neural Networks.

Chapter 3 proposes a new method for industrial image inspection. Fundamental concepts algorithms and neural network structure are also presented.

Chapter 4 expresses the development of the inspection system and software.

Chapter 5 deals with the simulation and shows experiment results.

Chapter 6 summarizes the conclusion and future work.

2. Literature Review

In this chapter, the structure, algorithms and NN application of AVI system are presented. The drawbacks of traditional methods and improvement goals are discussed.

2.1 Structure of AVI System

Traditionally, visual inspection and quality control are performed by human experts [14]. Although humans can perform jobs better than machines in many cases, they are slower than the machines and get tired quickly. Moreover, human experts are difficult to find or maintain in an industry, require training and their skills may take time to develop. There are also cases were inspection tends to be tedious or difficult, even for the best-trained experts. In certain applications, precise information must be quickly or repetitively used. Computer vision may effectively replace human inspection in such demanding cases.

Automated Visual Inspection (AVI) system employs a camera and image processing routines. Figure 2.1 illustrates the structure of a typical industrial vision system [14].

A computer is employed for processing the acquired images. This is achieved by applying special purpose image processing analysis software [15]. Images are usually acquired by one or more cameras placed at the scene under inspection. The positions of

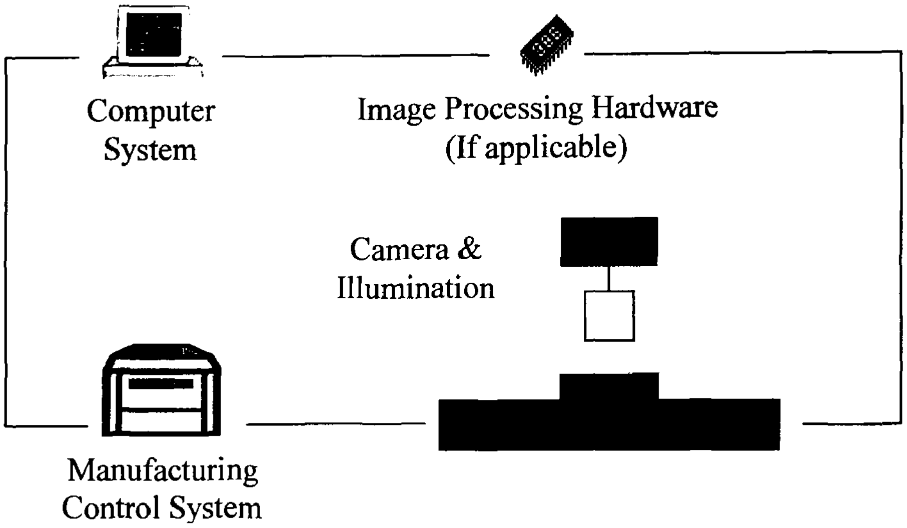


Figure 2.1 Industrial Image Inspection System

the cameras are usually fixed. In most cases, industrial automation systems are designed to inspect only known objects at fixed positions. The scene is appropriately illuminated and arranged in order to facilitate the reception of the image features necessary for processing and classification. When the process is highly time-constrained or computationally intensive and exceeds the processing capabilities of the main processor, application specific hardware, such as DSPs, ASICs, or FPGA, is employed to alleviate the problem of processing speed.

A lot of industrial activities have benefited from the application of AVI technology on manufacturing processes. These activities include: delicate electronics component manufacturing [16], quality textile production [13], printing [17] and circuits manufacturing [18] and many others.

A large amount of research has already been carried out [19-21]. Industry needs automated visual inspection (AVI) because in the manufacturing process some factors like defects, relative position and orientation error, etc. exist, which can be noticed by vision sensing [22].

AVI technology improves productivity and quality management and provides a competitive advantage to industries that employ this technology. It has undergone a development of over 30 years of research (in the computer vision field) and application (in the machine vision field). A number of papers could be found out for each processing step including Image Acquisition (IA), Image Enhancement (IE) Feature Extraction (FE) and Image Inspection (IIn). In this thesis, the focus is on the algorithms of IIn.

2.2 Image Inspection Algorithms of AVI System

A wide range of different algorithms have been used and also the classification methods could be different. Eduardo [23] has grouped the conventional visual inspection tasks into three broad categories based on the types of defects they detect: (a) dimensional verification, (b) surface detection methods, and (c) inspection of completeness. The conventional industrial image could as well be put into these categories.

A large number of image inspection technologies of AVI have been proposed in the literature. In general, they fall into one of three categories [18]: reference comparison (or referential approaches), non-referential approaches, and hybrid. Figure 2.2 shows the classification of these algorithms.

Among the cases of these three categories, image inspection algorithms are included in reference approaches, which is the focus of this review chapter.

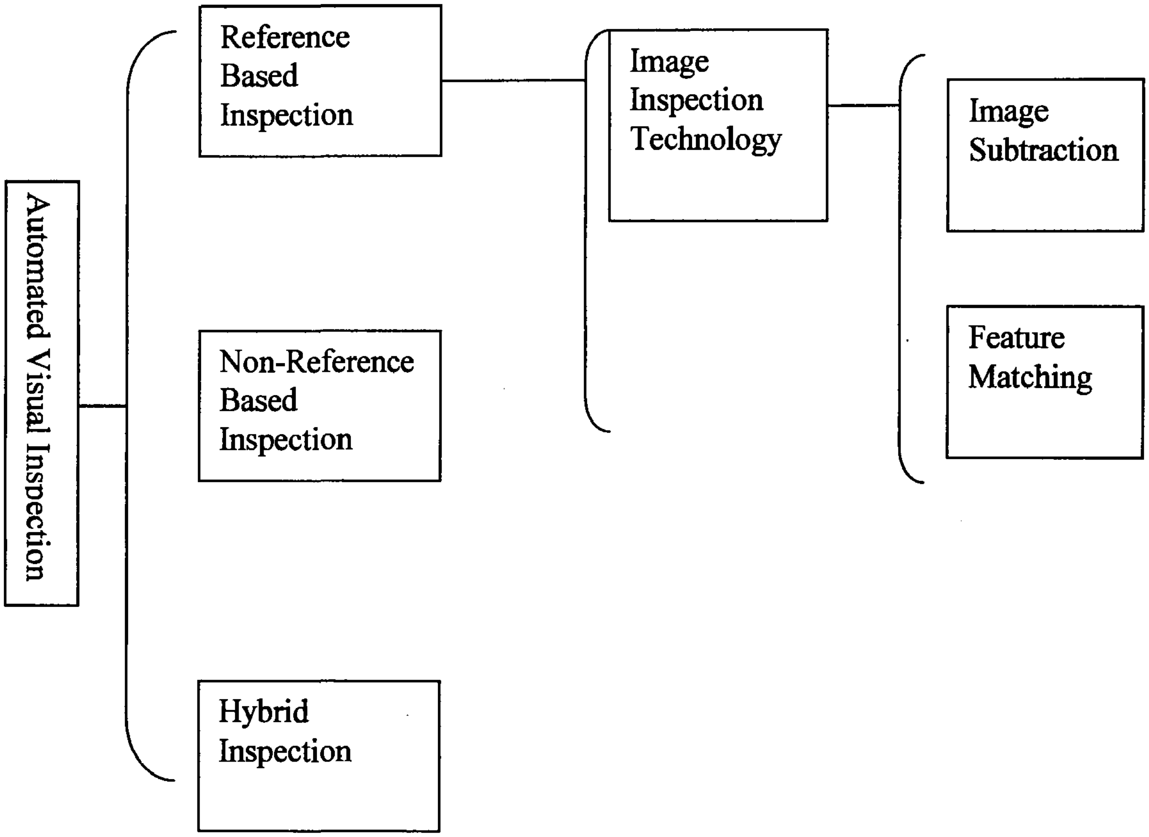

Automated Visual Inspection
Reference Based Inspection
Non-Reference Based Inspection
Hybrid Inspection
Image Inspection Technology
Image Subtraction
Feature Matching

Figure 2.2 Classifications of Inspection Algorithms

The reference methods execute a real point-to-point (or feature-to-feature) comparison between the image to be inspected and the "standard" image. These methods detect any difference between the image to be inspected and the "golden image" (standard image). The image comparison techniques mainly include two methods: Image Subtraction and Feature Matching.

Image Subtraction

Image subtraction is the simplest and most direct approach to the inspection problem. This is one of the earliest techniques employed in inspection. The image is acquired from IA system and is compared against the golden image. The subtracted image, showing defects, can subsequently be displayed and analyzed. Figure 2.3 shows this direct subtraction process as a logical XOR operation.

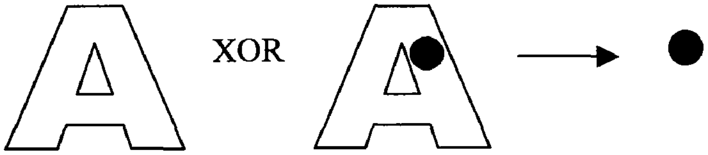


Figure 2.3 Image Subtraction

The advantage of this method is that it is easy to implement. Another advantage is that it allows for verification of the overall defects in the image. But this technique suffers from many practical problems, including registration, color variation, reflectivity variation, lighting sensitivity and other uncertain factors.

Feature Matching

Feature matching is an improved form of image subtraction, in which the extracted features from the object and those defined by the model are compared. The advantage of this matching is that it greatly compresses the data for storage, and at the same time reduces the sensitivity to the input data and enhances the robustness of the system. This matching process is also called template matching [24].

One of the major limitations of the template matching method is that an enormous number of templates must often be used, making the procedure computationally expensive. This problem can be eliminated if the features to be matched are invariant in size, location, and rotation. The disadvantages of this method are that it requires large data storage for the ideal image patterns, and precise registration is necessary for comparison. It is sensitive to illumination and digitization conditions, and the method lacks flexibility. Once the base image changed, the templates must be withdrawn again. To get the better inspection results, the template optimism usually has to be interfered manually.

2.3 Neural Network Application of AVI System

Artificial neural networks (ANN) have been studied almost from the beginning of the computer era. In the beginning the NN research was strongly motivated by biological considerations and the developed NN models were too weak to solve complex information processing tasks typically found in many industrial applications. New innovations in 1980s led to the emerge of more powerful NN models, and many successful case studies that were aimed at demonstrating NN concepts to various categories in a number of industrial application sectors stimulated the industry to look the NN as a important tool [25]. Nowadays NN has been used in a wide range of application sectors including the implementation in AVI systems.

2.3.1 Issues of Neural Network Application for AVI System

From the engineering point of view NN can be seen as highly parallel dynamical systems that can perform transformations by means of their state response to their input information. How the transformation is carried out depends on the NN model and its way of learning the transformation. The most natural application areas for the NN are those tasks that require the establishment of an appropriate transformation from certain inputs to certain outputs without analytical modeling. Therefore it is no wonder that the most successful applications of the NN can be found in the area of machine vision inspection, where such inputs to outputs transformations dominate the problem solving.

Much of the current research in NN is centered on individual network models, whereas in typical industrial applications, a system level view of NN is more desirable. Individual NN might be seen as components in a broader system, which also contains many other data processing techniques. This kind of use of NN leads to a hybrid architecture in which some of the processing modules are based on NN. Then the problem is to decide what benefits NN may provide for the given industrial application and what kinds of NN models should be used.

There are at least the following four main aspects that should be considered in NN application for vision inspection [25]:

1) Preparing the data

The training data should contain relevant information needed in building the NN model for the task. If possible, prior knowledge relevant for the problem should be considered.

2) Selecting the network model

The NN model acts crucially to the results obtained. Generalization ability, which is a measure of how well the network performs on the problem once training is complete, judges how good the network model is for the actual task. There are different types of NN models that can be divided to the following two broad categories according to their learning procedures: *supervised*, and *unsupervised* models.

3) Controlling the complexity

When training a network for a given problem, the task of a learning process is to construct a required transformation from the input space to the output space of the network. Any transformation of given inputs to outputs is a function approximation problem. The difficult problem is that the training samples might easily lead to multiple possible solutions. In order to obtain useful results, the NN complexity should be matched to the problem complexity and the number of available training examples.

If the network is too complex, it will perfectly learn the training set while generalizing very poorly. Controlling the complexity is therefore a necessity to ensure good generalization. It is specially a key issue when the training set is small, noisy and even partially incorrect. The practical methods for controlling the model complexity include methods such as early-stopped training, committees of early-stopped networks, weight decay or other regularization methods.

4) Assessing the performances of the network

The general way to determine how well a network has captured the nature of a function is to validate the network with additional test set examples that were not used during the learning steps. The results obtained with the test samples can be used as indicators of the generalization ability of the network. The aim should be at determining the level of condenses in the estimated generalization abilities of the model.

2.3.2 Training Data and Function of NN

Up till now many different NN models have been proposed for AVI applications, such as multilayer perceptron (MLP) [23]), principal component analysis (PCA) [23] and so on. As mentioned before, a system level view of NN is more desirable. Below will mention some NN applications in AVI and the focus is not on the type of NN models but how to prepare data for training and how to use NN for inspection.

In [26] about wood surface inspection system based on generic visual features, the shape feature is captured through the Gabor filter. NN is trained with shape feature of image and used as classifier of defect. In [27] about Neural quality inspection in industrial compact disc print stations, NN is trained with pixel data of image and is used as classifier of defect. In [28] about defect detection and classification on web textile fabric using multi-resolution decomposition and neural networks, NN is trained with Wavelet Transform parameters of image and used as classifier of defect. In [29] about automated inspection of solder joints-a neural network approach, NN is trained with Fourier Transform parameters of image and used as classifier of defect.

Similar research papers could be found more. Though the algorithms and NN model are different, there one thing in common: The NN is used to inspect image as a classifier. So the training data must include enough information and has to be pre-processed, e.g. Wavelet Transform calculation [28]. The intensity of data is large so that most of pre-processing and inspections in applications are very time consuming.

2.4 Drawbacks of Traditional AVI System

As mentioned before, referential methods, the most popular algorithms used for AVI, adopt a comparison no matter through pixel-based or feature-based method. An ideal image (standard image/golden image) or number of ideal images must be saved in the database in advance. To make a final decision, one or more threshold values must be setup. If the difference is out of the threshold value, it means that defect is found. Usually the threshold is set up based on operator's experience or trials.

Also referential methods do not consider uncertain factors like lighting sensitivity, machine dynamics, etc. But all these factors are not avoidable under real industrial environment.

Neural Networks are used in some application cases of AVI system. But the time consuming problem becomes the biggest obstacle for real industrial implement.

From the viewpoint of industry there are three main difficulties for AVI to be used widely:

- Speed: Usually high speed production line is used in modern industries. Real time monitoring and inspecting system is required to synchronize with the speed of product line and automatic tolerance definition.

- Technology: At present AVI systems require controlled environment and precise positioning, and assumes everything is in perfect condition. But the real working environment is not that perfect, and an improvement should be made to handle the uncertainties like illumination, etc.
- Re-usability : The threshold values have to re-setup due to the variation in base image. An auto-configuration strategy should be investigated.

The new method proposed in this thesis is to deal with these problems, and according to the experiment results, the proposed method is proven effective.

Summary

In this chapter, the traditional AVI system is introduced , followed by the discussion on the drawbacks and improvement goals of traditional AVI method.

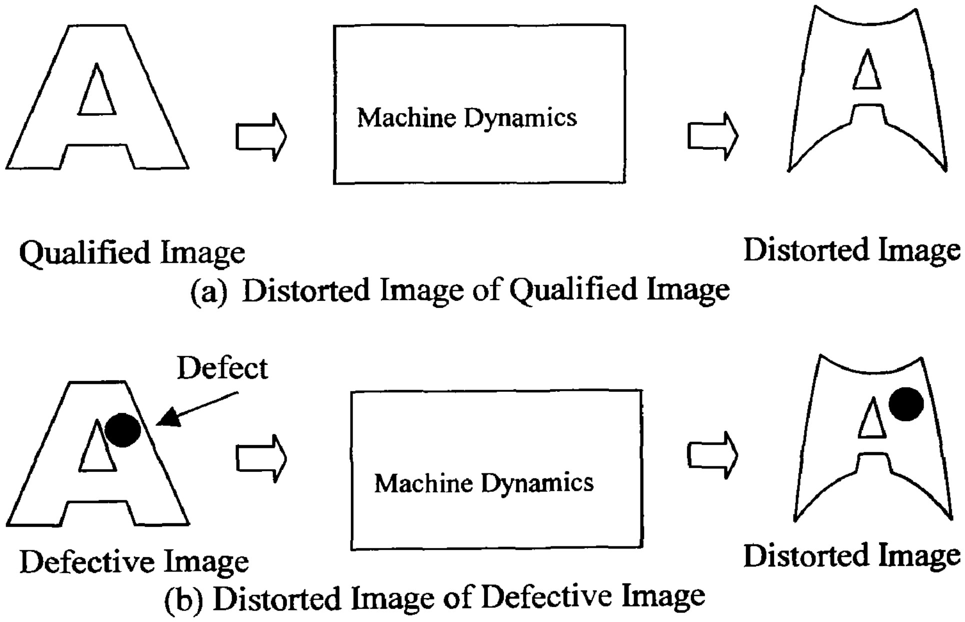
It has been identified that the inspection speed and interference of uncertain factors are the main problems of traditional AVI system. The following chapter describes a new method to deal with these problems.

3. Hybrid Method

In this chapter, a statistics based Neural Network method is addressed in detail. The new method mainly deals with the speed and accuracy problems.

3.1 Problem Statement

The problem under study is to detect defective items with a good accuracy using image inspection by accounting for uncertainties including machine dynamics, product variation and illumination variation. As an example to explain, Figure 3.1 shows a letter "A" under inspection. Due to machine dynamics, a good letter "A", as shown in Figure 3.1(a), is distorted when captured by a CCD camera. If not treated properly, this letter would be regarded as defective, while in reality the original letter is good.



(a) Distorted Image of Qualified Image

(b) Distorted Image of Defective Image

Figure 3.1 Distorted Images Due to Machine Dynamics

Therefore, the first requirement is to filter out the image distortions caused by uncertainties so that the image shown in Figure 3.1(a) will be judged as qualified and that in Figure 3.1(b) as defective. The second requirement is to automatically set a tolerance for inspection in order to account for the ranges of the uncertainties. The proposed algorithms use a statistics method to define a tolerance zone and use a NN method to take systematic uncertainties into consideration.

3.2 Statistics Method

3.2.1 Live Image and Sample Image

Live image is the image captured from the running machine. It could be a qualified one also could be a defective one. Sample image is a live image, which is a qualified image manually selected.

3.2.2 Golden Image

A common practice for mechanical inspection is to compare an item under inspection to a reference that may be a gauge or a template. Likewise, in AVI, an inspection image is required to compare to a reference image, called golden image in industry. In reality however, there is no single image that can be selected as a true golden image. There are only qualified images or defective images. A reasonable way is to create a golden image based on a number of qualified images. In this thesis, an averaging method is used to create such an image by adding the corresponding pixels of all the qualified sample images and dividing them by the number of sample images. The pixel values used are all in gray scale. Figure 3.2 shows the steps to generate a virtual golden image.

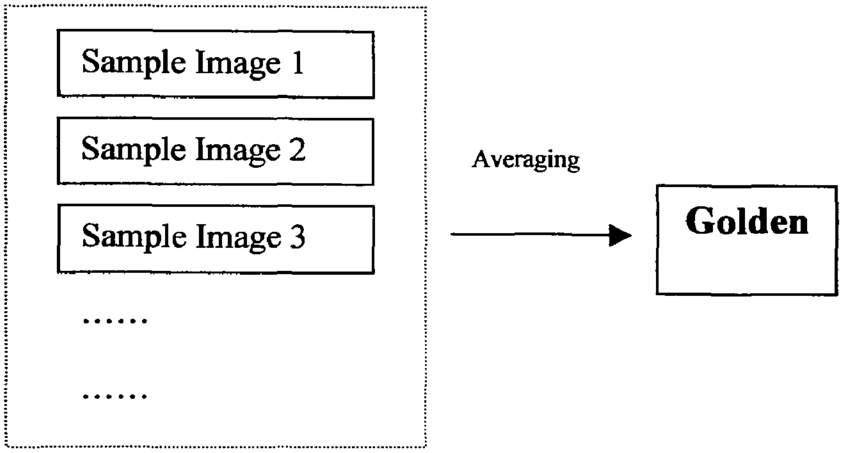


Figure 3.2 Generation of Virtual Golden Image

In general, uncertainties are divided into *random* and *systematic*. Background noise is an example of random uncertainty, whereas machine dynamics is an example of systematic uncertainty. From the signal processing theory, it is known that the averaging method can filter out random uncertainty to certain extent. In general, the more the samples are used for averaging, the better the result. The NN method is used to treat systematic uncertainty.

3.2.3 Two Indices

To develop an inspection method that can be readily visualized in a 2-dimensional plane, two indices are needed. Since variance measures the deviation from the mean value, it is selected to establish a tolerance zone that defines an allowable area within which the items under inspection are considered acceptable. Considering that images are 2-dimensional signals, the variances of the rows and the columns are selected as the two indices. The minimum and maximum values of the two indices provide four corner points that can form a rectangular tolerance zone in the 2-D plane.

The definitions of row/column numbers and pixel coordinates are given in Figure 3.3.

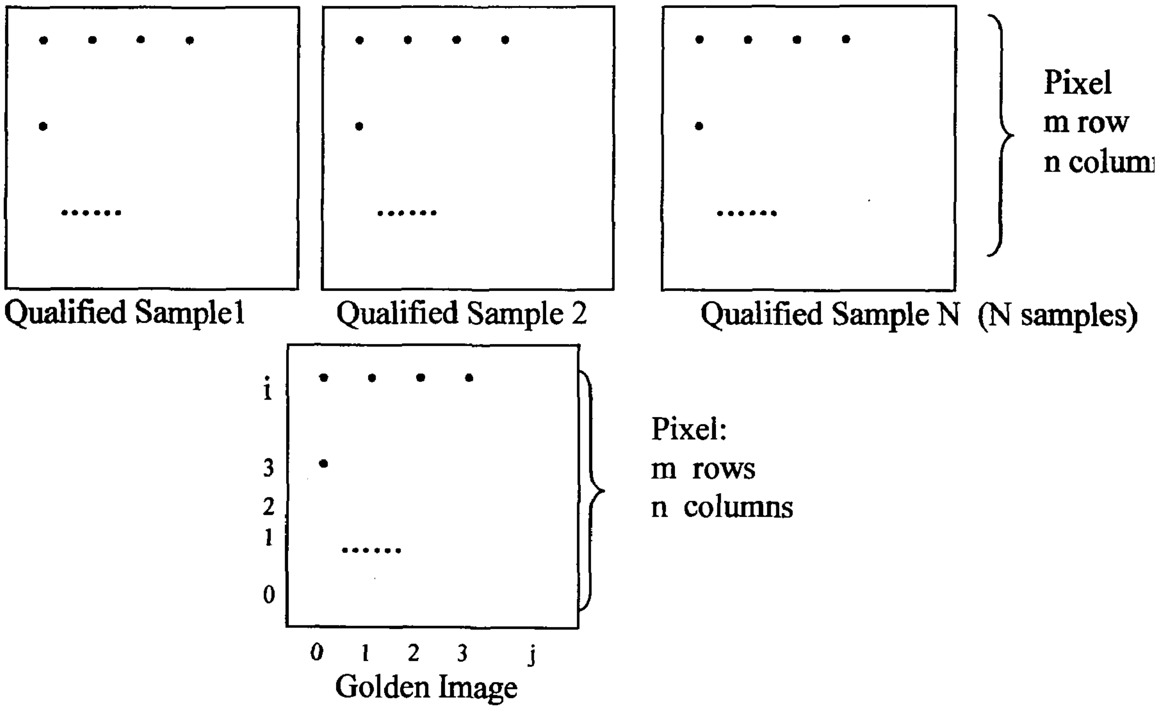


Figure 3.3 Row/Column Number and Pixel Coordinate Definition

Since the golden image is created by the averaging method, it is actually a "mean value image". Therefore, for every pixel, the variance is defined as

$$\sigma_{ij}^{\ 2} = \frac{1}{N}\sum_{k=1}^{N}(\mathrm{P}_{ij} - \mathrm{S}_{k\_ij})^2 \tag{3.1}$$

where N is the total number of the sample images, $\sigma_{ij}^{\ 2}$ denotes the variance of pixel at (i, j), P_{ij} is the gray level of the golden image at pixel (i,j), and S_{k_ij} is the gray level of the corresponding pixel in the *kth* sample image, see Figure 3.4.

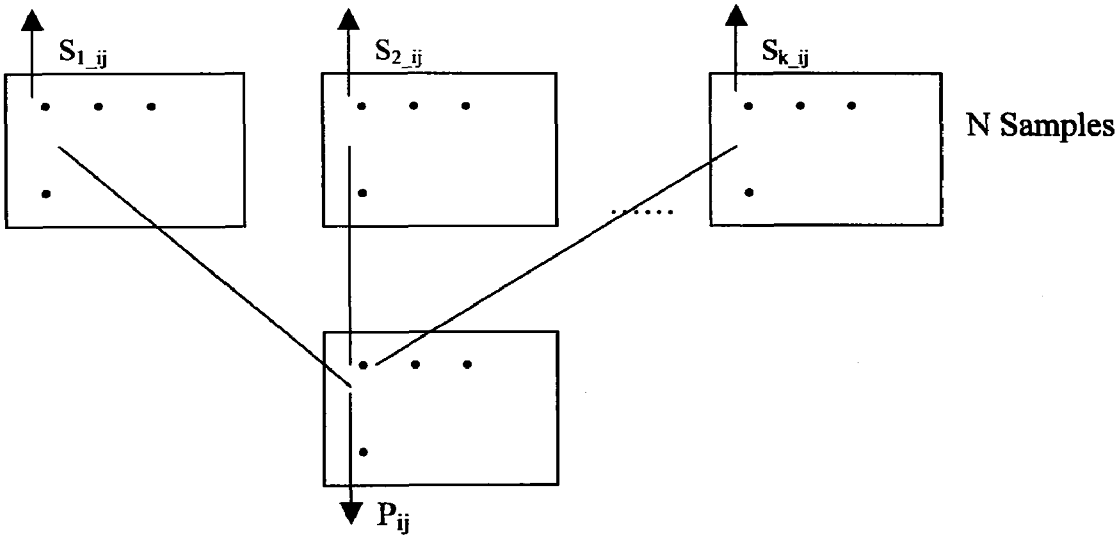


Figure 3.4 Row and Column Indices

The variance for a row is determined as

$$\sigma_{r\_i}^{\ 2} = \frac{1}{n}\sum_{j=0}^{n-1}\sigma_{ij}^{\ 2} \tag{3.2}$$

where *n* is the number of pixels in a row, and $\sigma^2_{r\_i}$ denotes the variance of the *ith* row. Likewise, the variance for a column is determined as

$$\sigma_{c\_j}{}^2 = \frac{1}{m}\sum_{i=0}^{m-1}\sigma_{ij}{}^2 \qquad (3.3)$$

where *m* is the number of pixels in a column, and $\sigma^2_{c\_i}$ denotes the variance of the *jth* column.

The minimum and maximum values of the row variance, denoted by $\sigma^2_{\text{r-min}}$ and $\sigma^2_{\text{r-max}}$, respectively, are obtained as

$$\sigma_{r\_\min}{}^2 = \min\,(\sigma_{r\_0}{}^2, \sigma_{r\_1}{}^2, \sigma_{r\_2}{}^2, \ldots.\ \sigma_{r\_m-1}{}^2) \qquad (3.4a)$$

$$\sigma_{r\_\max}{}^2 = \max\,(\sigma_{r\_0}{}^2, \sigma_{r\_1}{}^2, \sigma_{r\_2}{}^2, \ldots.\ \sigma_{r\_m-1}{}^2) \qquad (3.4b)$$

The minimum and maximum values of the column variance, denoted by $\sigma^2_{\text{r-min}}$ and $\sigma^2_{\text{r-max}}$, respectively, are obtained as

$$\sigma_{c\_\min}{}^2 = \min\,(\sigma_{c\_0}{}^2, \sigma_{c\_1}{}^2, \sigma_{c\_2}{}^2, \ldots.\ \sigma_{c\_n-1}{}^2) \qquad (3.5a)$$

$$\sigma_{c\_\max}{}^2 = \min\,(\sigma_{c\_0}{}^2, \sigma_{c^2\_1}, \sigma_{c\_2}{}^2, \ldots.\ \sigma_{c\_n-1}{}^2) \qquad (3.5b)$$

Note that *n* and *m* also indicate respectively the number of columns and the number of rows for the image under inspection, and *n*× *m* represents the resolution of the image.

3.2.4 Inspection

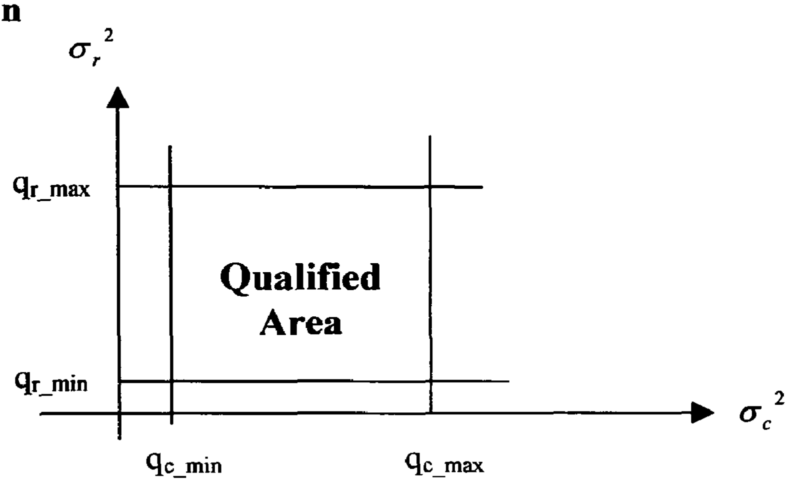


Figure 3.5 Tolerance Zone

The inspection of statistics method is just using $\sigma_{r\text{-max}}^2$ ($q_{r\text{-max}}$), $\sigma_{r\text{-min}}^2$ ($q_{r\text{-min}}$), $\sigma_{c\text{-max}}^2$ ($q_{c\text{-max}}$), and $\sigma_{c\text{-min}}^2$ ($q_{c\text{-min}}$) to create a tolerance zone in the 2-D plane, shown in Figure 3.5. Hence only the random uncertainties are taken into consideration.

3.3 Statistic Method Based Neural Networks Method

3.3.1 Rudiments on Neural Networks

The method of the neural networks is to mimic the phenomenon of the biological neurons of the human being. The biological neurons could learn to establish an internal relationship using the input signals and their response signals. This method provides an effective way of modeling complex systems that cannot be analytically modeled using the conventional theories.

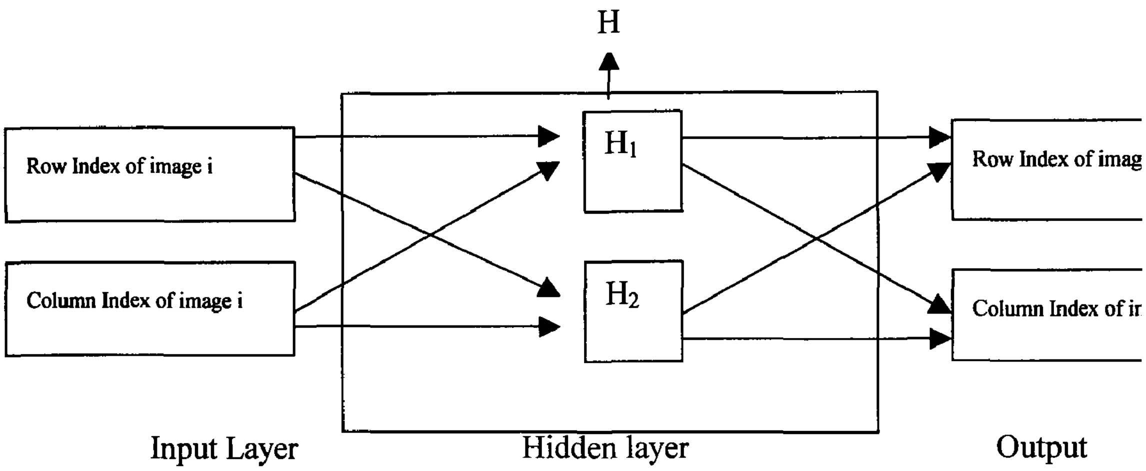


Figure 3.6 NN Structure

The basic structure of the neural networks consists of layers and nodes. As illustrate in Figure 3.6, a basic engineering neural network has an input layer, an output layer, and a number of hidden layers (in Figure 3.6, only one hidden layer is shown whose has two nodes).

There are a number of nodes in each layer. At the input layer, the number of nodes is determined by the number of the input signals. At the output layer, the number of nodes is determined by the number of the output signals. The values of the inputs and outputs should be obtained by experiments.

The hidden layers represent the internal relationship that requires to be identified through training using the inputs and outputs. Determination of the number of the hidden layers and the number of the nodes on each hidden layers is one part of the neural networks training. Another part is to determine the weights on each layer.

The weights determine how the values are transmitted from a layer to another. For the neural networks training, inputs and outputs are known from experiment. In the first run of the training, the weights are arbitrarily given. Therefore the values of the inputs will be passed onto the hidden layer(s) and in turn to the outputs. However, there will be a difference in the outputs between the value from computation and the value from the experiment. The difference will then be used to compute a set of updated weights so the computed output will converge to the experiment output.

This process usually takes a number of iterations. Once the difference becomes smaller than a given tolerance, the trained neural network is considered as the black box that can represent the internal relation. Again, this neural network is defined by the number of layers, the number of nodes and their weights.

3.3.2 Challenging Issues

The challenge in the neural networks training for the proposed method is that it i impossible to run a series of tests with measurable factors and then to obtain th associated images. Only available data includes:

1) Live images obtained during machine running

2) Still images obtained without running the machine

There are two methods that could be considered for the Neural Networks training.

Training method 1

This method is solely based on the live images. For a given set of the live images, half c images will be taken for training and another half for validation. For training, a neur network will be trained between two live images, one as an input and another as a output. The first training is considered done when the input live image can t transformed into the second one. This process continues till all possible pairs of th training images are trained. The resulting neural network is considered having capture the systematic uncertainty.

The trained neural network will then be validated by the set of the validation live images. This method may be considered as capturing relative machine dynamics and image printing variation.

Training method 2

This method would use the live images and still images. Again, for a given set of images, half images will be taken for training and another half for validation. For training, a neural network will be trained between a live image (input) and a still image (output). The first training is considered done when the input live image can be transformed into the still image. This process continues till all possible pairs of the training images are trained. The resulting neural network is considered having captured systematic uncertainty.

The trained neural network will then be validated by the set of the validation live images. This method may be considered as capturing absolute systematic uncertainty.

Theoretically, the second method would be more effective. However, according to the real industrial environment, only live images are available. Therefore, the neural networks training has to be performed using the first method, which may capture the relative, but not absolute, systematic uncertainty.

3.3.3 Inputs and Outputs

Figure 3.6 shows the Neural Networks structure used in this paper, and it is a three-laye model, consisting of input layer, hidden layer, and output layer, with two nodes in eac layer.

As mentioned before, there are two basic types of NN training, namely, *supervisec* *unsupervised* [23]. The supervised NN training requires both inputs and outputs. In cas that there are input and output data when training, the supervised method is applied. A mentioned above, in most image inspection applications uncertainties about machin dynamics and product variations are difficult to measure and only attainable informatio is usually the image. In this research, the Hebbian learning rule [23] of the supervised N training is adopted that only requires the excitation of a pair of measurable items unde study. For image inspection, this pair would be a pair of images. Therefore, for th selected NN structure under our study, the input and output are the average row an column index of a pair of live images, and they are defined as

$$\sigma_{r\_average}^{2} = \frac{1}{m}\sum_{i=0}^{m-1}\sigma_{r\_i}^{2} \qquad (3.6a)$$

$$\sigma_{c\_average}^{2} = \frac{1}{n}\sum_{j=0}^{n-1}\sigma_{c\_j}^{2} \qquad (3.6b)$$

where $\sigma^{2}_{\text{r-average}}$ and $\sigma_{\text{c\_average}}^{2}$ denote the average row and column index of an image respectively, and $\sigma_{\text{r-i}}^{2}$ and $\sigma_{\text{c\_j}}^{2}$ denote the variance of the i*th* row and the j*th* column respectively.

3.3.4 Training

For a given set of qualified images, the training process is carried out considering all pairs. The total number of pairs is determined using the binomial coefficients as

$$N_{Pairs} = \begin{bmatrix} N \\ K \end{bmatrix} = \frac{N(N-1)}{2} \qquad (3.7)$$

where N is the number of sample images as defined before, and k is the number of a pair, i.e. k=2. For example, if ten images (N=10) are used for training, there are 45 pairs in total. Figure 3.7 shows how these pairs are formed and Table 3.1 lists the input values and output values of all these pairs.

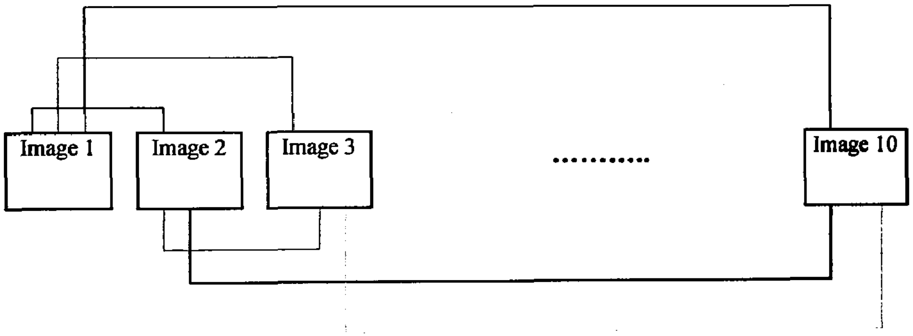


Figure 3.7 Formation of Image Pairs

Table 3.1 Inputs and Outputs of Image Pairs for Training

	Input	Output
No.1	i1_RowIndex	i2_RowIndex
	i1_ ColumnIndex	i2_ ColumnIndex
No.2	i1_RowIndex	i3_RowIndex
	i1_ ColumnIndex	i3_ ColumnIndex
……	……	……
No.9	i1_RowIndex	i10_RowIndex
	i1_ ColumnIndex	i10_ ColumnIndex
No.10	i2_RowIndex	i3_RowIndex
	i2_ ColumnIndex	i3_ ColumnIndex
No.11	i2_RowIndex	i4_RowIndex
	i2_ ColumnIndex	i4_ ColumnIndex
……	……	……
No.17	i2_RowIndex	i10_RowIndex
	i2_ ColumnIndex	i10_ ColumnIndex
……	……	……
No.45	i9_RowIndex	i10_RowIndex
	i9_ ColumnIndex	i10_ ColumnIndex

The first training is carried out for the first pair. It is considered done when the two indices of the first sample image is transformed into that of the second one. This process continues till all the pairs of the sample images are trained. The resulting Neural Networks are considered having captured the systematic uncertainties.

The hidden layer in fact represents a transfer function between the inputs and outputs. After training, the transfer function, denoted by H, is available, and it can be regarded as the filter that can filter out the systematic uncertainties to certain extent.

3.3.5 Tolerance Zone

After H is obtained, the maximum and minimum indices of each sample image are re-processed to filter out the systematic uncertainties. As illustrated in Figure 3.8, $\sigma_{r\text{-}min}^2$, $\sigma_{r\text{-}max}^2$, $\sigma_{c\text{-}min}^2$ and $\sigma_{c\text{-}max}^2$ before filtering are re-computed to $G_{r\text{-}min}$, $G_{r\text{-}max}$, $G_{c\text{-}min}$ and $G_{c\text{-}max}$ after filtering.

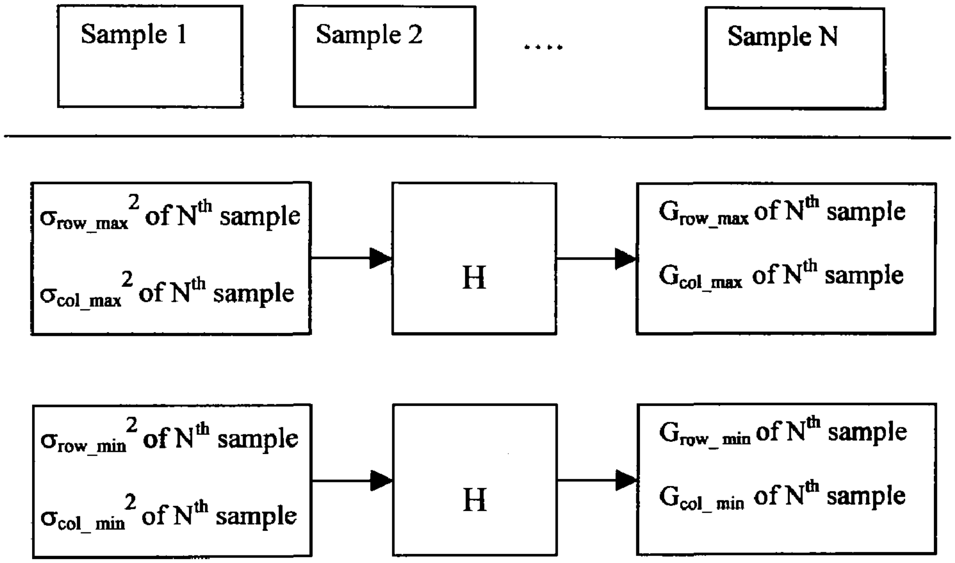

Sample 1
Sample 2
....
Sample N
$\sigma_{row\_max}^2$ of Nth sample
$\sigma_{col\_max}^2$ of Nth sample
H
G_{row_max} of Nth sample
G_{col_max} of Nth sample
$\sigma_{row\_min}^2$ of Nth sample
$\sigma_{col\_min}^2$ of Nth sample
H
G_{row_min} of Nth sample
G_{col_min} of Nth sample

Figure 3.8 Filtering by H

To define a tolerance zone, the global maximum and minimum indices are searched over all the sample images, as defined below

$$q_{r\_max} = \max\{(G_{r\_max})_1, (G_{r\_max})_2, \ldots\ldots, (G_{r\_max})_N\} \quad (3.8a)$$

$$q_{r\_min} = \min\{(G_{r\_min})_1, (G_{r\_min})_2, \ldots\ldots, (G_{r\_min})_N\} \quad (3.8b)$$

where q_{r_min} and q_{r_max} define the qualified row index range. Likewise,

$$q_{c\_max} = \max\{(G_{c\_max})_1, (G_{c\_max})_2, \ldots\ldots, (G_{c\_max})_N\} \qquad (3.9a)$$

$$q_{c\_min} = \min\{(G_{c\_min})_1, (G_{c\_min})_2, \ldots\ldots, (G_{c\_min})_N\} \qquad (3.9b)$$

where q_{c_min} and q_{c_max} define the qualified column index range. The tolerance zone is then defined by q_{r_max}, q_{r_min}, q_{c_max} and q_{c_min} as shown in Figure 3.5.

3.3.6 Inspection

Once a tolerance zone is established, the image inspection can be carried out. To do so, as shown in Figure 3.9, the maximum and minimum indices of a live image are processed by the transfer function to filter out the relatively machine dynamics and image printing variation. The point (L_{col_max}, L_{row_max}) corresponds to the maximum index of live image without machine dynamics and image printing variation; the point ($L_{col\_ min}$, $L_{row\_ min}$) corresponds to the minimum index of live image without machine dynamics and image printing variation.

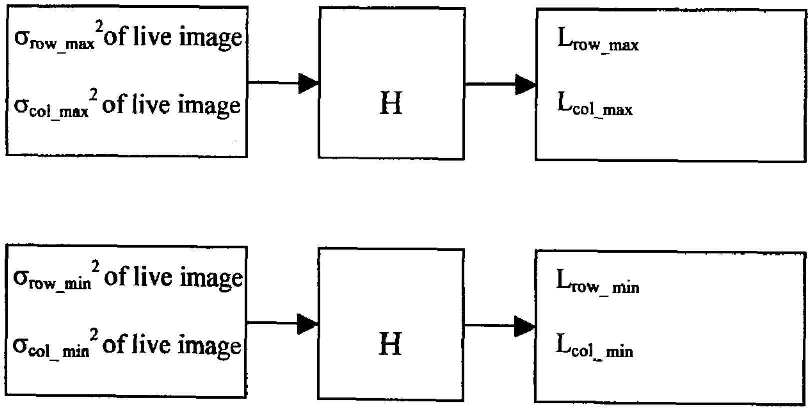

$\sigma_{row\_max}{}^2$ of live image
$\sigma_{col\_max}{}^2$ of live image
H
L_{row_max}
L_{col_max}
$\sigma_{row\_min}{}^2$ of live image
$\sigma_{col\_ min}{}^2$ of live image
H
$L_{row\_ min}$
$L_{col\_ min}$

Figure 3.9 Inspection

After filtering, if it is within the qualified row index range and qualified column index range, the image is considered qualified. Otherwise it is considered defective. In other words, if the point (L_{col_max} , L_{row_max}) and the point ($L_{col\_ min}$, $L_{row\_ min}$) are all located in the tolerance zone, the live image is regarded as qualified.

Figure 3.10 shows one example of inspecting a defective image. From the training result, the qualified row index range [q_{r_min} , q_{r_max}] = [5.1671,6.8696], and the qualified column index range [q_{c_min}, q_{c_max}] = [5.7181, 6.9459]. The processed live image indices are: (L_{col_max} , L_{row_max}) = (6.9395, 6.8606), ($L_{col\_ min}$, $L_{row\_ min}$) = (5.7165, 5.1376). When compared with the tolerance zone, the point ($L_{col\_ min}$, $L_{row\_ min}$) is not inside of the qualified area. Therefore, this live image is considered as un-qualified.

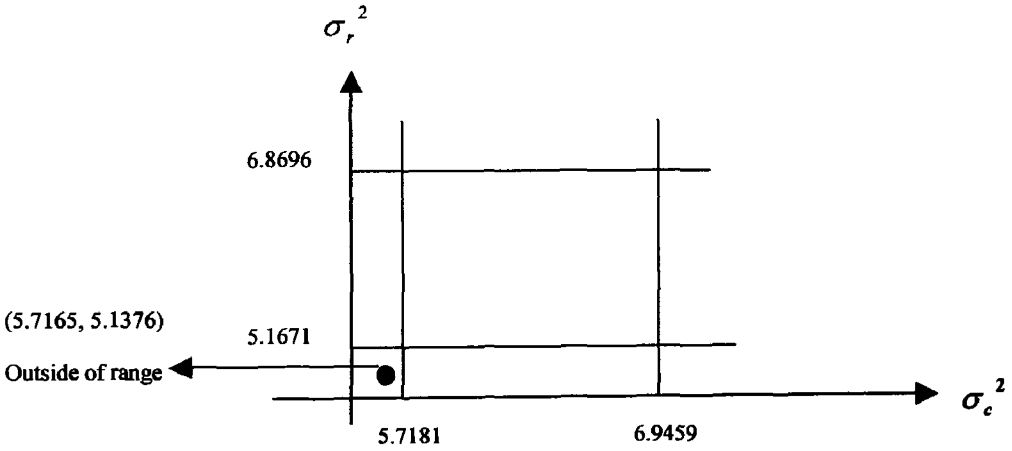


Figure 3.10 Inspection (sample of defective image)

Summary

In this chapter, the statistic method and statistic method based Neural Network method are expressed in detail. The statistic method is fast but does not consider the uncertain factors. The combined method inherits the fast speed of property and also accounts for the uncertain factors. Theoretically, it is more accurate than pure statistic method. The experiment based on real industrial machine is done in subsequent chapters.

4. Implement of AVI System with New Algorithm

In this chapter, the implement of new AVI system based on the proposed method is described. The new AVI system use sample images to train Neural Network which could filter out the uncertainties. The image for inspection is filtered by Neural Network first to get rid of uncertainties and then compared with the tolerance zone. The software package of AVI system is developed by using Visual C++6.0 and Matlab for Microsoft Windows system. The functions of software include image manipulation, image acquisition, image registration, image training, and image inspection.

4.1 Image Manipulation

Figure 4.1 is a block diagram listing the major components of the software package development. Besides the main functions, basic software interface of images processing tools has been developed, which can perform basic image processing functions, including image input in BMP and TIFF format (two image format files); image display; conversion of color images to gray scale images; pixels manipulation; defect simulation; conversion between bmp format and tiff format. It can also be used to view, crop and register images.

4.1.1 Tiff and Bmp Access

Tiff is the format of images that industrial camera supports. Because the standard C/C++ does not have libraries to access the tiff format, ***Libtiff*** is used to open and save the tiff image files. ***Libtiff*** is a standard ANSI C library for implementing the tiff specifications, which can work on a number of operating systems. Through ***Libtiff*** and Visual C++ API, tiff images can be opened and saved on Windows system.

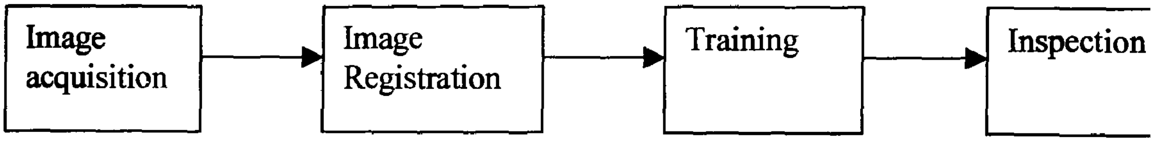


Figure 4.1 Software Components

BMP is a simple image format and easy to handle for image processing, and hence it is used for our AVI system.

A BMP file consists of 4 parts as shown in Table 4.1. The first part is header; the second part is information section; the third part is the palette if the image is indexed color format; and last part is the pixel data. The position of the image data with respect to the start of the file is contained in the header. Information such as the image width and height, the type of compression, the number of colors is contained in the information header. Then immediately after the information header there will be a color table. Every item of color table consists of 4 bytes. The first three bytes correspond to blue, green, red components; the last byte is reserved/unused but could obviously represent the alpha channel. The color pointer of every pixel is saved in pixel data section. The color pointer points to a color index in color table.

To improve the compatibility and flexibility of software, 3 types of conversion are also implemented in software system, including BMP to TIFF, TIFF to BMP and color image to gray scale image, software interface is indicated in Figure 4.2.

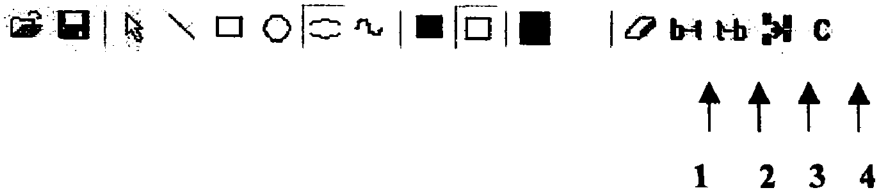


Figure 4.2 Format Conversion

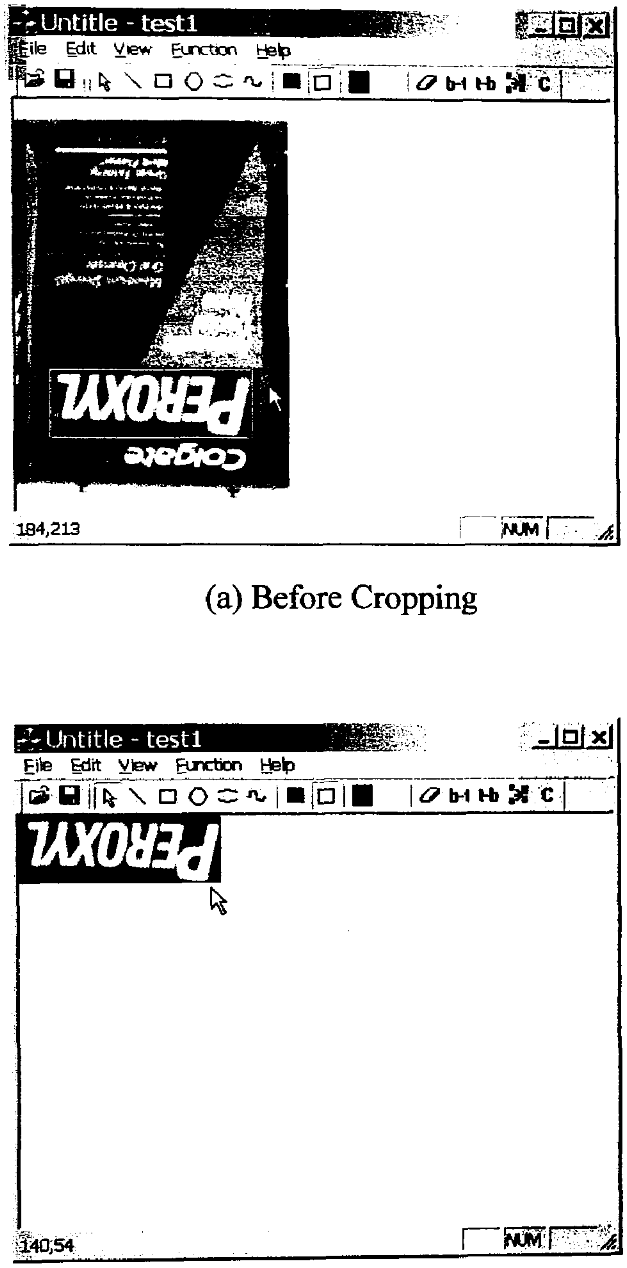
4.1.2 Image Edit

The edit function is mainly used to crop the desired part of the image, as shown in Figure 4.3, and simulate defects, as shown in Figure 4.4. The simulated defects include line, circle, curve, rectangle, and ellipse.

Table 4.1 BMP Format

Constitution of BMP file
Bitmap-file header
Bitmap-information header
Color table
Pixel data

Detail structure of BMP file				
	Offset	Field	Length	Description
Bitmap-file header	000h	File ID	2 bytes	Type of BMP
	0002h	File Size	1 dword	
	0006h	Reserved	1 dword	0
	000Ah	Bitmap Data offset	1 dword	Pixel data of offset
	000Eh	Bitmap Header Size	1 dword	
	0012h	Width	1 dword	
	0016h	Height	1 dword	
	001Ah	Planes	1 word	
Bitmap-information header	001Ch	Bits Per Pixel	1 word	
	001Eh	Compression	1 dword	0 – No compress 1 - RLE 8 2 - RLE 4 3 – Bit fields
	0022h	Bitmap Data Size	1 dword	
	0026h	HResolution	1 dword	Pixel/meter
	002Ah	VResolution	1 dword	Pixel/meter
	002Eh	Colors	1 dword	
	0032h	Important Colors	1 dword	
Palette data	----	Color table	N * 4 byte	1 byte for blue 1 byte for green 1 byte for red 1 byte reserved
Pixel data	----	Pixel data	----	



(a) Before Cropping

(b) After Cropping

Figure 4.3 Crop Function

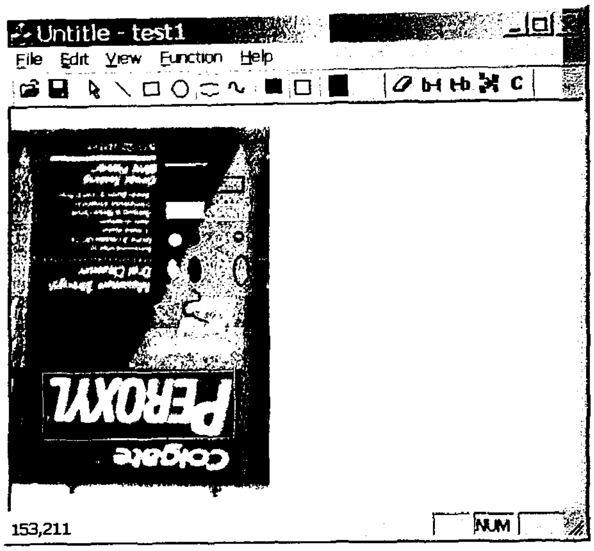


Figure 4.4 Defect Simulation

4.2 Image Acquisition

In development, a CCD camera, called *Smart Camera* (SC), is being used in the inspection machines. SC is produced by the Vision Component Company.

The Smart Camera is a compact, lightweight black-and-white video camera. It integrates a high-resolution CCD sensor with a fast frame-processing signal processor. A dynamic RAM is used to store data and video frames. The Smart Camera has its own operation system, API and communicates with the computer through TCP/IP protocol.

The main structure of the program developed to operate the camera is described in Figure 4.5. A Client-Server model software program has been developed for this project. The Smart Camera is regarded as a server, as it has its own IP address and operating system. The computer is regarded as a client.

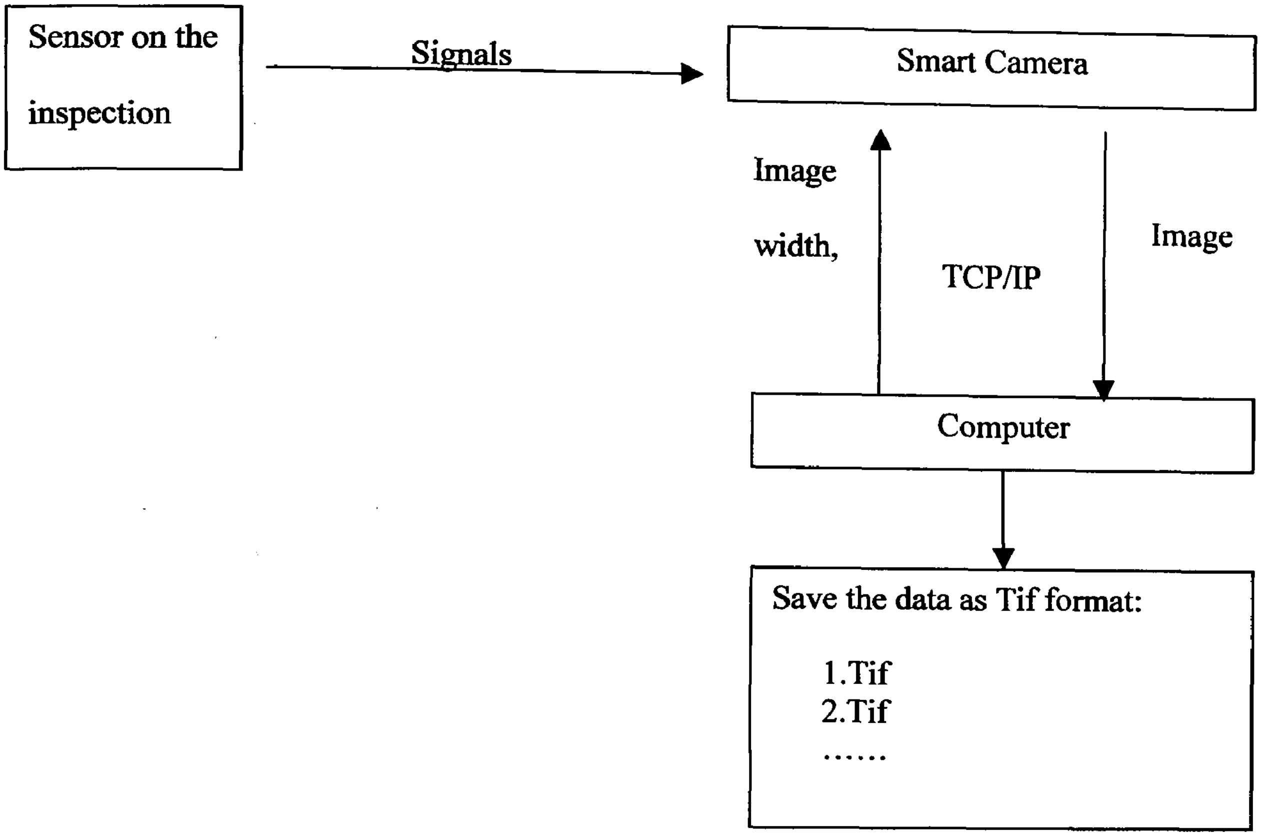


Figure 4.5 Program Structure for Operating Smart Camera

4.2.1 Server Program Running on the Smart Camera

The server program is used to monitor the signals coming from the sensor installed on the inspection machine. A coming signal indicates that an image passes through the camera. Once the server program receives a signal, it will trigger the flash to take a picture at once and then send the image data to the Client computer through TCP/IP. The image data is the data stream consisting of gray scale values for all the pixels of the picture.

It should also be mentioned that the Smart Camera is a DSP equipment. To create the program that can run on it, the Code Composer Studio (CCS) has to be used. The full name of CCS is *TMS320C6000 Code Composer Studio 2.2.1* that is the product of the Texas Instruments Incorporated. It can compile and link C programs to be executed on the DSP equipment.

4.2.2 Client Program Running on the Client Computer

The client program is used to send the desired image size to the server (Smart Camera), and then wait for the response from the server. When the client program receives the data of a picture through TCP/IP, it will convert it to the bmp format and save it on a computer. This program is developed by using Microsoft Visual C++ 6.0 under Windows system.

4.3 Image Registration

4.3.1 Light Review on Image Registration

Image registration is the process of overlaying two or more images of the same scene taken at different times, from different viewpoints, and/or by different sensors. It geometrically aligns two images - the reference and original images. Image registration is a crucial step in all image analysis. It is usually regarded as a critical pre-processing step to many modern image processing and computer vision tasks. Many algorithms and techniques have been proposed to address the registration problem [31-33].

Typically, registration is required in many fields including [32]: remote sensing (multispectral classification, environmental monitoring, change detection, image mosaicing, weather forecasting, creating super-resolution images, integrating information into geographic information systems (GIS)), in medicine (combining computer topography (CT) and NMR data to obtain more complete information about the patient, monitoring tumour growth, treatment verification, comparison of the patient's data with anatomical atlases), in cartography (map updating), and in computer vision (target localization, automatic quality control) .

Because only available data of this project is the image itself, image-based method for registration is the only way, which could be roughly divided into extrinsic methods [31] and intrinsic methods [31]. The extrinsic method is based on foreign objects introduced into the imaged space. The intrinsic method is based on the image information itself. In the proposed method, the intrinsic method is used because the only available data is image itself.

Registration of intrinsic methods can be based on a limited set of identified salient points (landmarks), on the alignment of segmented binary structures (segmentation based), most commonly object surfaces, or directly onto measures computed from the image grey values (voxel property based).

Landmark Based Registration

In landmark based registration, the set of identified points is compared to the original image content, which makes for relatively fast optimization procedures. Such algorithms optimize measures such as the average distance between each landmark and its closest counterpart (the Procrustean metric), or iterated minimal landmark distances. But first of all, the landmarks should be properly assigned on the original image and reference image.

Segmentation Based Registration

With segmentation based registration method, the same structures (mostly surfaces) are extracted from both images to be registered, and used as the sole input for the alignment procedure. An extracted structure (also mostly surfaces, and curves) from one image is elastically deformed to fit the second image.

The segmentation task is fairly easy to perform, and the computational complexity is relatively low. But the drawback of segmentation-based methods is that the registration accuracy is limited to the accuracy of the segmentation step. In theory, segmentation based registration is commonly automated but for the segmentation step, it is performed semi-automatically for most of the time.

Voxel Property Based Registration

The voxel property based registration methods actually operate directly on the image grey values, without prior data reduction by the user or segmentation. There are two distinct approaches. The first is to use the full image content throughout the registration process, and the second is to immediately reduce the image grey value content to a representative set of scalars and orientations.

Theoretically, voxel property based methods are the most flexible of registration methods since they do not start with reducing the grey valued image to relatively sparse extracted information, but use all of the available information throughout the registration process. But applications based on this method are limited by the considerable computational costs.

Principal axes and moments based methods are the prime examples of reductive registration methods. Within these methods the image centre of gravity and its principal orientations (principal axes) are computed from the image zeroth and first order moments.

Registration is then performed by aligning the center of gravity and the principal orientations. Sometimes, higher order moments are also computed and used in the process. The result is usually not very accurate. Despite its drawbacks, principal axes methods are widely used in registration problems that require no high accuracy, because of the automatic and very fast nature of its use and the easy implementation.

4.3.2 Two Point-Based Registration

A new, two point-based registration method is developed for image registration. The concept of this method is a hybrid of landmark-based method and axes/moments based method.

This method is based on computing the center positions of two letter images. By using two points each on the base image and registration image, two corresponding lines can be identified on the two images and used to register them both in position and orientation. It is very simple and computationally efficient.

Step 1: Zone Selection

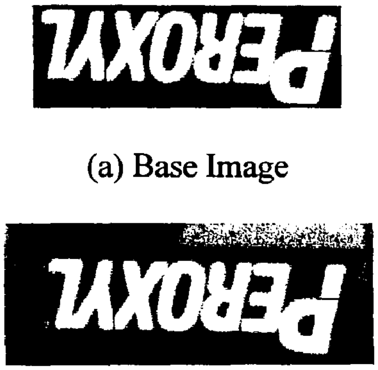
Select two registration zones (usually letters) and find the center of each zone for the base image and sample image.

Figure 4.6(a) shows a base image for registration in which two areas are picked. The centers of these two areas are used as two reference points and computed as

$$Gx = \sum_{X=0}^{N-1} \sum_{Y=0}^{M-1} \frac{xb_{x,y}}{K} \quad \text{and} \quad Gy = \sum_{X=0}^{N-1} \sum_{Y=0}^{M-1} \frac{yb_{x,y}}{K} \qquad (4.1)$$

where $b_{x,y}$ denotes the pixel value of position (x,y) in the image, *Gx and Gy* are the x and y coordinates of the center position, and

Figure 4.6(b) shows an image to be registered, in which two corresponding areas are also picked. Figure 19(b) is used to computer their centre positions.



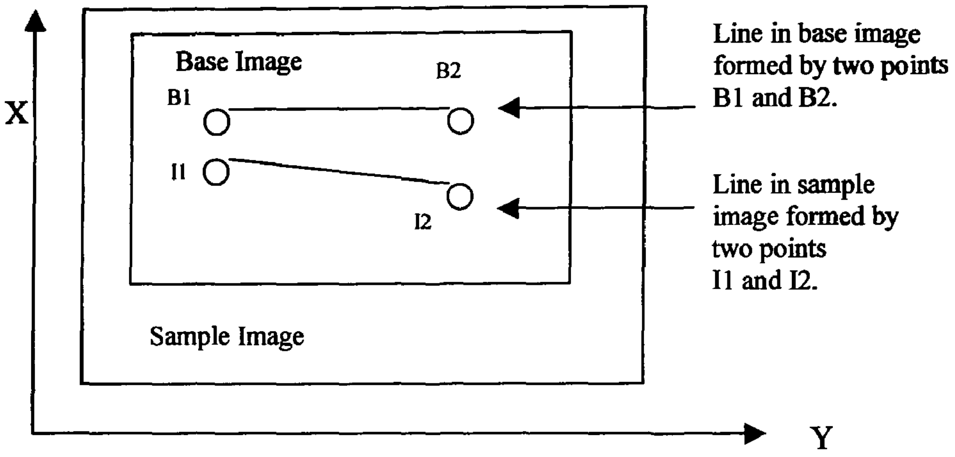
(a) Base Image

(b) Image to Be Registered

Figure 4.6 Two Point Registration

Step 2: Position Registration

With the two pairs of points at hand, two lines can be formed for both the base image and registration image, respectively. A middle point can be readily computed for each of the two lines. The two images can be registered in the right position using the two corresponding middle points. Position registration is simple addition or subtraction of the registration image pixels by the difference relative to the base image, see Figure 4.7.



X
Y
Base Image
B1
B2
I1
I2
Sample Image
Line in base image formed by two points B1 and B2.
Line in sample image formed by two points I1 and I2.

Figure 4.7 Position Registration

Step 3: Orientation Registration

Orientation registration of the two images is done by aligning the two lines around thei middle points. The angle difference is computed as:

$$Tan\partial_{base} = \frac{B2_x - B1_x}{B2_y - B1_y} \quad Tan\partial_{sample} = \frac{I2_x - I1_x}{I2_y - I1_y} \tag{4.2}$$

$$Tan\partial = Tan(\partial_{sample} - \partial_{base}) = \frac{Tan\partial_{sample} - Tan\partial_{base}}{1 + Tan\partial_{sample}Tan\partial_{base}} \tag{4.3}$$

Where ($B1_x$,$B1_y$) and ($B2_x$,$B2_y$) denote the coordinates of two gravity center of bas image. ($I1_x$, $I1_y$) and ($I2_x$,$I2_y$) denote the coordinates of two gravity center of sampl image. ∂_{base} denotes the angle of gravity center line of base image. ∂_{sample} denotes th angle of gravity center line of sample image. ∂ denotes the angle between two lines Orientation registration of the two images is done by aligning the two lines around thei middle points.

All the pixels in sample image would be rotated angel ∂ around the middle point of two gravity centres.

The pixel coordinate equation for rotating around center is given as (see Figure 4.8):

$$x1 = (x - xCenter)\cos\partial + (y - yCenter)\sin\partial + xCenter \tag{4.4}$$

$$y1 = (y - yCenter)\cos\partial - (x - xCenter)\sin\partial + yCenter \tag{4.5}$$

In case that the rotation angle is very small, Sine and Cosine of rotation angles could be approximately replaced by radian of rotation angle and 1, respectively.

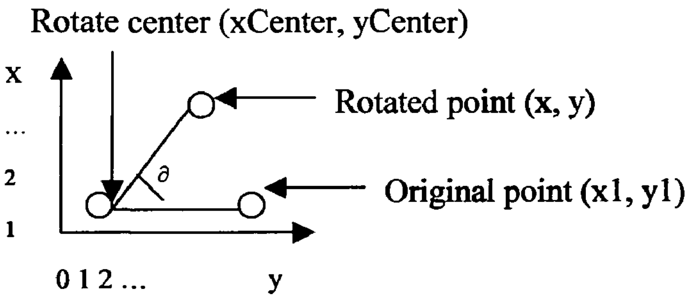


Figure 4.8 Orientation Registration

The equations are given as

Basic Method (Sin&Cos calculation):

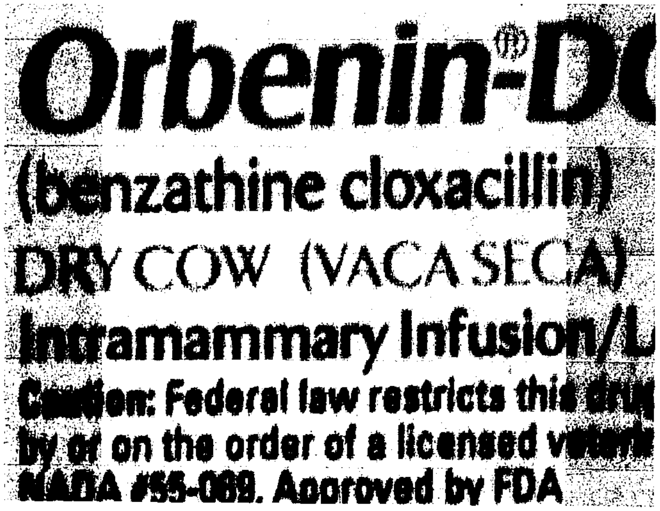
$$\begin{bmatrix} X1 \\ Y1 \end{bmatrix} = \begin{bmatrix} Cos\theta & -Sin\theta \\ Sin\theta & Cos\theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \tag{4.6}$$

Simplified Method (Radian calculation)

$$\begin{bmatrix} X1 \\ Y1 \end{bmatrix} = \begin{bmatrix} 1 & -\theta \\ \theta & 1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \qquad (4.7)$$

Where θ is the radian of rotation angel, (x,y) is the coordinate of pixel, (x1,y1) is th coordinate of pixel after rotating.

The proposed testing method is to rotate the original image with the same angle through 2 methods separately, and then compare the difference between rotating results by the basi method and the rotating result by simplified method. The rotation centre is located at th image centre.



Orbenin-D
(benzathine cloxacillin)
DRY COW (VACA SECA)
Intramammary Infusion/L
Caution: Federal law restricts this drug
by or on the order of a licensed veteri
NADA #55-069. Approved by FDA

Figure 4.9 Original Image

Figure 4.9 shows the original image: 320 x 240 pixels. Figure 4.10 shows the images tha are rotated by basic method and simplified method. Although no obvious difference coul be found by eyes, the rotated image through the simplified method is not exactly the sam as the rotated image through the basic method.

For example, a certain pixel's coordinate in the original image is (10,10). Through the basic method, the new pixel coordinate after rotating is (7,14). Through the simplified method, the coordinate calculation result is (8,14). This pixel will be counted for the different pixels.

Base on the original image (320x240pixels), Table 4.2 shows the detail different pixel number between the images rotated through the two methods. From Table 4.2 it could be said that the simplified method performs very well when rotating angel is 1°. The rotated images through two methods are almost same.

When the rotating angle goes up to 5°, almost 49% pixels are different between the rotated image through the basic method and the rotated image through the simplified method.

Figure 4.11 shows a registration example of the software package.

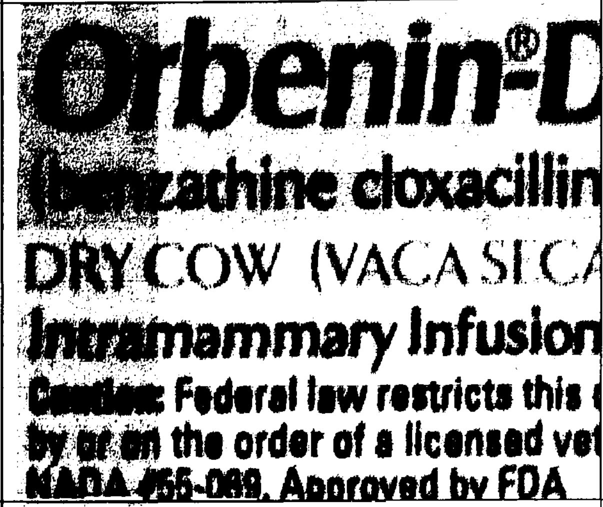
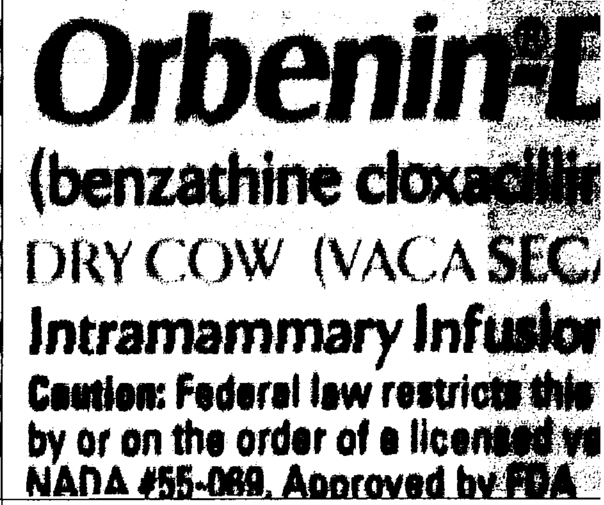
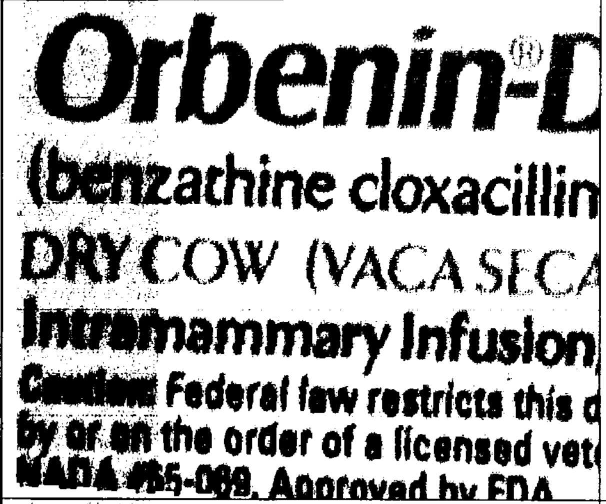
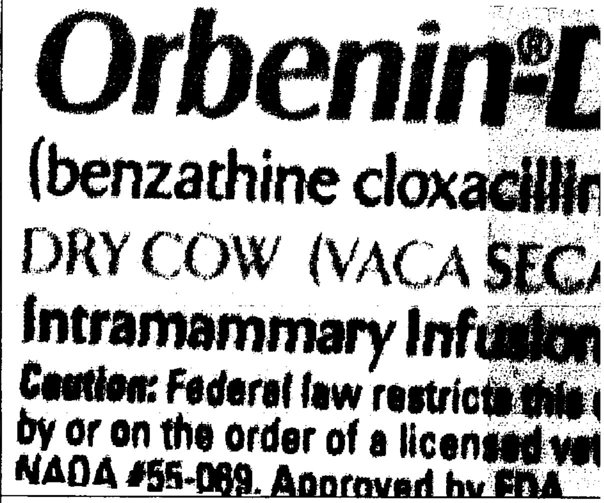
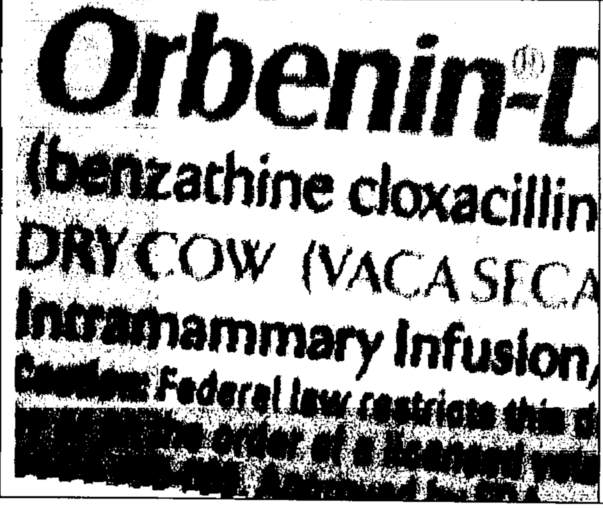
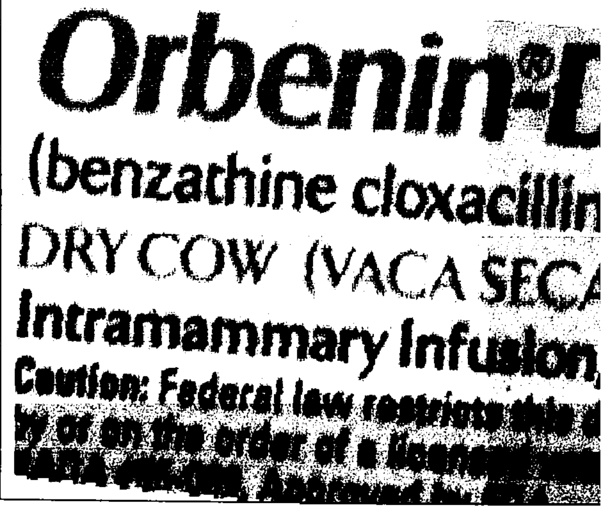
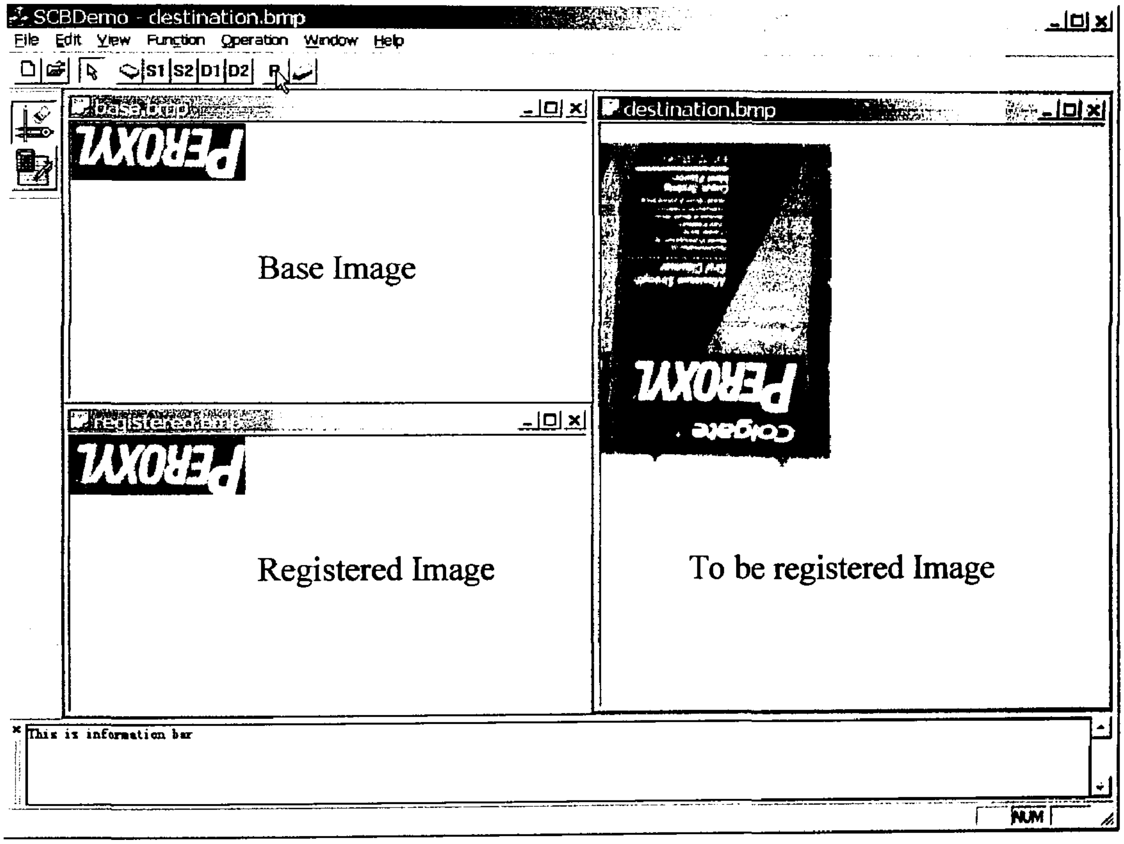
Angel	Rotated image (basic method)	Rotated image (simplified method)
1°		
3°		
5°		

Figure 4.10 Rotated Images

Table 4.2 Comparison Between Images Rotated Through Two Methods

Rotation angel	The number of different pixels between 2 images	Ratio: different pixels to total pixels
1°	1677	2%
3°	14844	19%
5°	38227	49%



This is information bar

Figure 4.11 Registration

4.4 Training

A number of sample images can be selected from the software for training. The two indices of each image will be calculated and paired automatically, then sent to Neural

Networks for training. Figure 4.12 shows the training process and the convergenc performance curve.

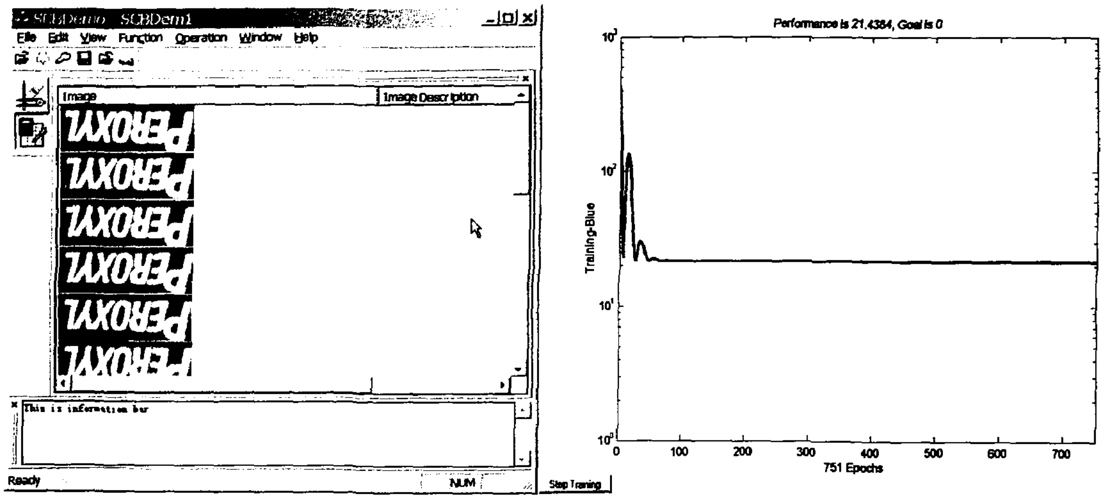


Figure 4.12 Neural Network Training

Neural Networks operations are implemented by calling Matlab external libraries. Matla supports external libraries for C/C++ to access Matlab Engine. The Visual C++ send Matlab command scripts to the background Matlab Engine. After calculation, Visual C+ also retrieves the results from Matlab Engine through Matlab external library interfac Figure 4.13 shows the procedure.

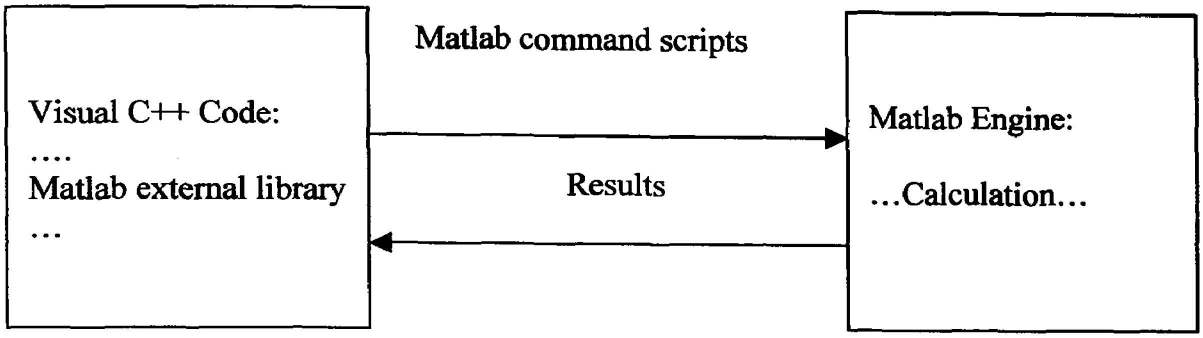


Figure 4.13 Neural Networks Implementation

4.5 Inspection

After the tolerance zone is established, image inspection can be carried out. To do so, the maximum and minimum indices of a live (inspection) image are processed by the transfer function to account for the systematic uncertainties. As shown in Figure 3.9, $\sigma_{r\text{-min}}^2$, $\sigma_{r\text{-max}}^2$, $\sigma_{c\text{-min}}^2$ and $\sigma_{c\text{-max}}^2$ before filtering are re-computed to $L_{r\text{-min}}$, $L_{r\text{-max}}$, $L_{c\text{-min}}$ and $L_{c\text{-max}}$ after filtering.

Two points $(L_{col\_max}, L_{row\_max})$ and $(L_{col\_min}, L_{row\_min})$ are used to compare to the tolerance zone. If both points lie inside, the inspection item by AVI is considered qualified, otherwise as defective.

Figures 4.14 and 4.15 show an example of inspection.

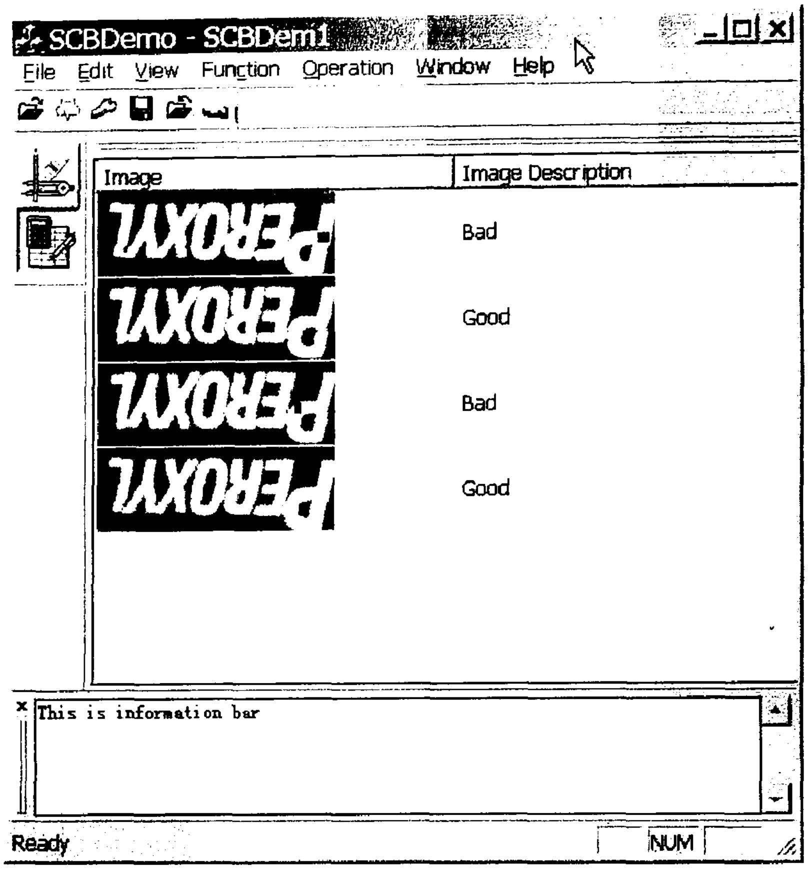
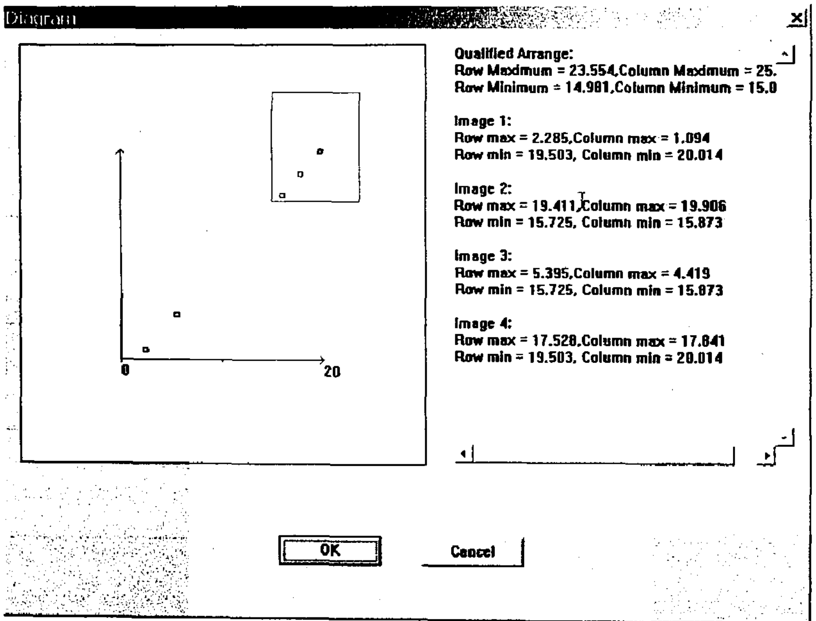


Image	Image Description
	Bad
	Good
	Bad
	Good

```
This is information bar
```

Figure 4.14 Live Images Inspected



```
Qualified Arrange:
Row Maxdmum = 23.554,Column Maxdmum = 25.
Row Minimum = 14.981,Column Minimum = 15.0

Image 1:
Row max = 2.285,Column max = 1.094
Row min = 19.503, Column min = 20.014

Image 2:
Row max = 19.411,Column max = 19.906
Row min = 15.725, Column min = 15.873

Image 3:
Row max = 5.395,Column max = 4.419
Row min = 15.725, Column min = 15.873

Image 4:
Row max = 17.528,Column max = 17.841
Row min = 19.503, Column min = 20.014
```

Figure 4.15 Indices Calculated and Comparison to The Qualified Area

Summary

In this chapter, the software development based on new method for AVI is presented. A new two-points based registration method is proposed in detail. The whole software program is implemented by using Visual C++6.0, Matlab and Code Composer Studio. In following chapter, the new method for AVI is verified on the real industrial machine.

5. Experimental Verification

In this chapter, the statistics based Neural Network method is verified on real industrial inspection machine. The sensitivity and accuracy of new algorithms are verified based on large number of defect simulations.

5.1 System Introduction

The AVI system consists of computer system and machine system. The machine system is used to control the status of image running, acquire the image and send the image to the computer. The computer system is employed for processing the acquired images.

Figure 5.1 shows the machine system used for algorithm verification. The panel on the left of figure is used to setup parameters of machine, such as the speed of image running. The black box on the top of figure is the camera that is used to capture the image. There is an Ethernet cable that connects the camera and computer system so that the image can be transferred from machine system to computer system. The computer system is the regular Microsoft Windows computer system with the software package described in last chapter.

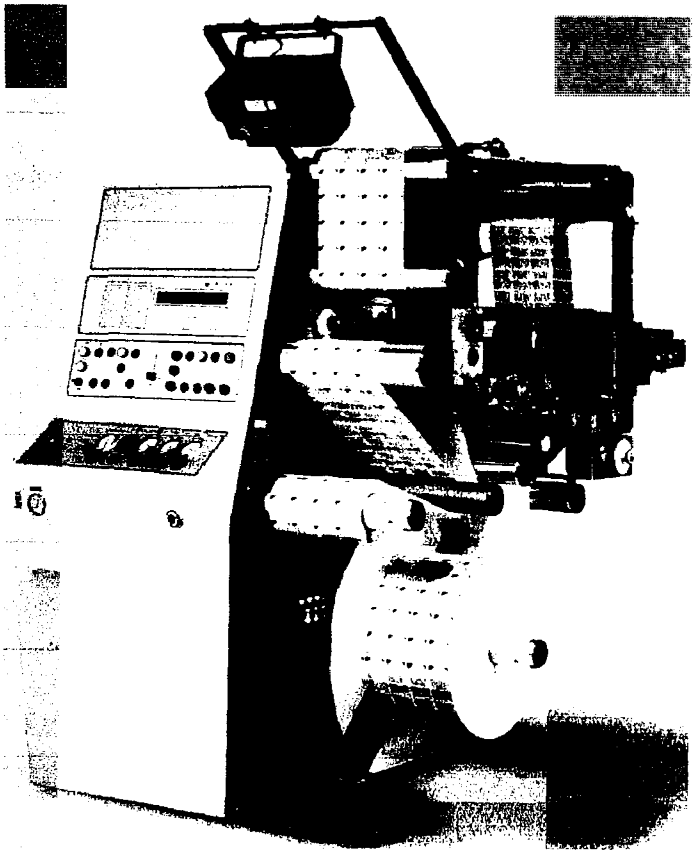


Figure 5.1 Industrial Inspection Machine (Rotoflex International Ltd)

5.2 Sensitivity Verification

Sensitivity verification is to study how the two inspection indices change with the changes in the image qualities. The factors considered include:

- Web stretch effect.

  A simple way to simulate is to scale up the width of the image and scale down the height.

- Shift effect.

  The shift effect could be simulated by moving all the pixels of the image up or down, or left or right.

- Printing Pressure Variation.

  The printing pressure variation could be simulated by adding or subtracting a graylevel value to all the pixels of the image.

- Content-based variation.

  This will simulate the same defect with different background images, such as regular background, pure white background, and pure black background. It will show how the image contents affect inspection.

- Computational time of two indices.

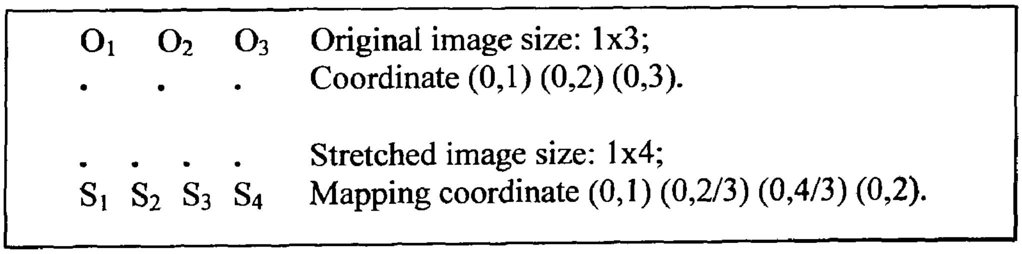
  The computation time on the two inspection indices will be recorded in terms of image size. This will provide a base for consideration of real time inspection.

5.2.1 Web Stretch Effect

It assumes that stretch is even in the width direction, that is, the running direction of the web. Stretch simulation requires two steps: the creation of new pixel location, and the assignment of gray value to new locations.

Suppose that an image has a size of 100x200(row x column) pixels and it is stretched to 100x220. The method is to lay an imaginary 100x220 grid over the original image. Obviously, the spacing (row direction) between the grids would be less than one pixel because it needs to fit into a smaller image. So the stretched image pixels have a fractional coordinates after mapping to the original image.

Linear interpolation is used to calculate the gray value of pixels of the stretched image. Usually only the integer part of the pixel coordinates is used. With the linear interpolation, the fraction part of the pixel coordinate is used to calculate a weight factor for two adjacent pixels in the original image (row direction) so that the final gray value of the pixel on the stretched image is a mix of the two adjacent pixels in the original image.



Original image size: 1x3;
Coordinate (0,1) (0,2) (0,3).

Stretched image size: 1x4;
Mapping coordinate (0,1) (0,2/3) (0,4/3) (0,2).

Figure 5.2 Coordinates Mapping for Stretched Image

Figure 5.2 is a simple example to show how to stretch the image in the row direction, where Oi donates the gray values of original image pixels; Sj denotes the gray values of the stretched image pixels. The stretched image pixel gray values are given by the following equations.

$$
\begin{aligned}
&S1 = O1 \\
&S2 = [1\text{-modf}(2/3)]\text{x}O1 + \text{modf}(2/3)\text{x}O2 = 1/3\text{x}O1 + 2/3\text{x}O2 \\
&S3 = [1\text{-modf}(4/3)]\text{x}O2 + \text{modf}(4/3)\text{x}O3 = 2/3\text{x}O1 + 1/3\text{x}O3 \\
&S4 = O3
\end{aligned}
\qquad (5.1)
$$

where *modf* is a function of getting the fractional part of a floating-point value. For example, modf(0.5) = 0.5; and modf(3.7) = 0.7.

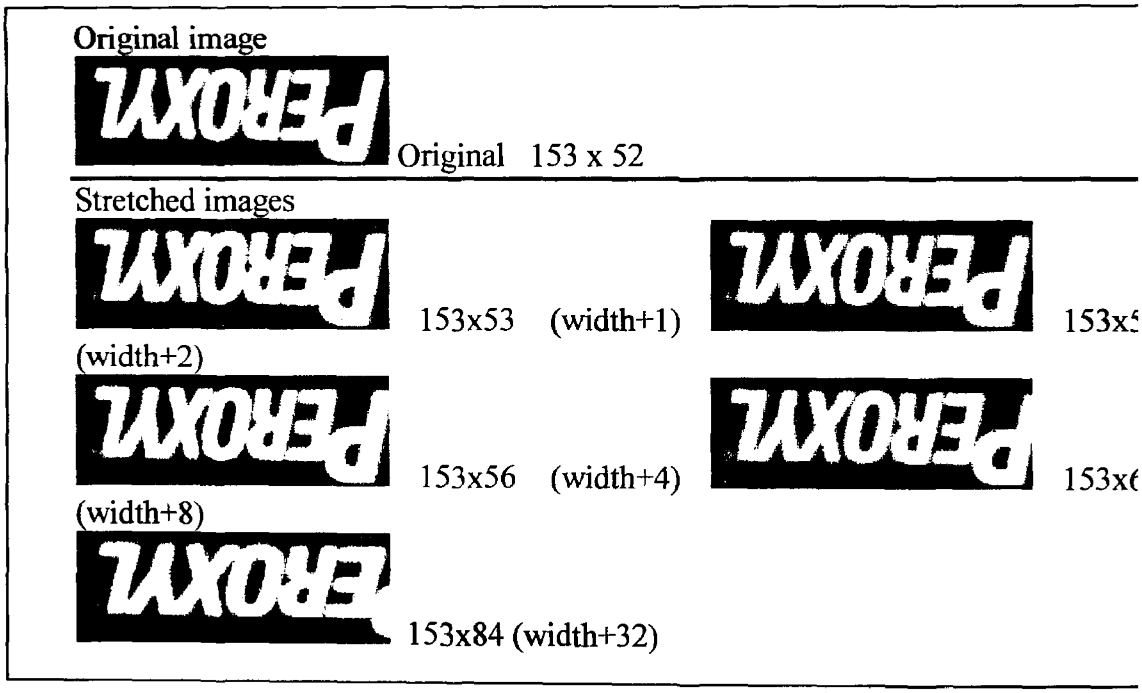


Figure 5.3 Stretched Images

Figure 5.3 shows the sample pictures used to simulate the stretch effect, where the image is stretched from left to right on the row direction, and the section beyond the original size is chopped off.

According to the simulation of the stretched images shown in Figure 5.3, Table 5.1 gives the testing results. All the stretched cases are detected by the neural network method because every pixel of the stretched image is recalculated through this simulation method.

Table 5.1 Inspection Result For Stretch Effect

Qualified Range	Width+1	Width+2	Width+4	Width+8
Row: 7.5818 - 12.9197	5.6607, 10.7293	2.3947, 10.7297	-1.7181, 10.7250	-5.3963, 10.7322
Column: 9.2172 - 10.9397	8.4942, 10.9408	6.2702, 10.9408	4.2154, 10.9410	0.9413, 10.9406
	Defect	Defect	Defect	Defect

5.2.2 Shift Effect

The shift effect is simply simulated by moving all the pixels of the image in the row or column direction, for example, up or down, left or right. Figure 5.4 shows the sample pictures that simulate the shift effect, where the images are shifted down 1, 2 and 5 pixels respectively. It could also be shifted in the other directions such as up, left, or right.

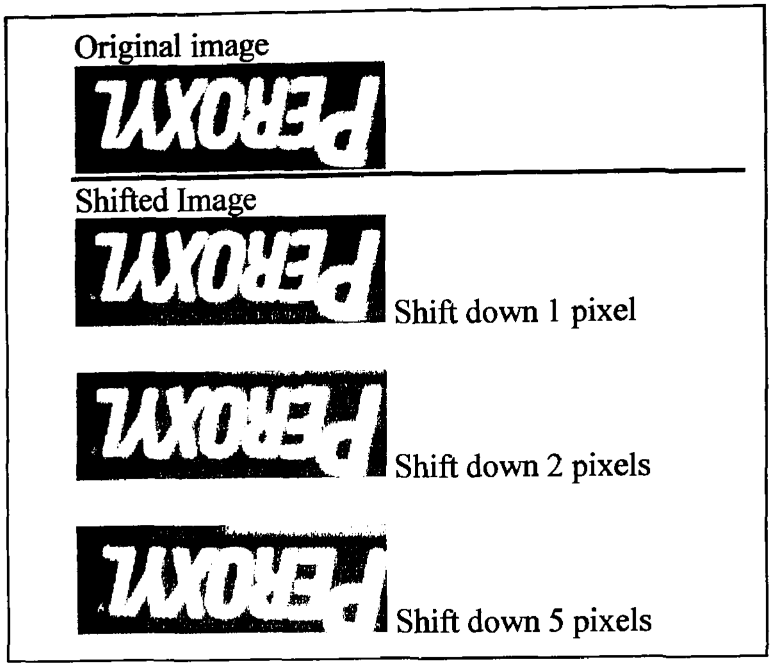


Figure 5.4 Shifted Images

According to the simulation of the shifted images shown in Figure 5.4, the testing results are given in Table 5.2. From the results, it can be seen that the Neural Network method is very sensitive to the image shift.

Table 5.2 Inspection Result For Shift Effect

Image size: 153x52

Qualified Range	Shift 1 pixel	Shift 2 pixels	Shift 5 pixels
Row: 7.5818 - 12.9197	4.5195, 10.6912	0.5986, 10.7130	-4.0021, 8.5674
Column: 9.2172 - 10.9397	7.2525, 10.9311	6.3621, 10.9379	1.8138, 9.2642
	Defect	Defect	Defect

5.2.3 Print Pressure Variation

Print pressure variation could be simulated by adding or subtracting a certain gray-level value to all the pixels of the image. The detail calculation equation is given below:

$$G_{ij} = G_{ij} \pm g \tag{5.2}$$

Where G_{ij} denotes the pixel gray value. Subscript i and j denote the coordinate of the pixel: i denotes the ith row, j denotes the jth column. g is a certain gray value from 0 to 255 that represents the variation of print pressure. If $G_{ij} + g$, the image is getting lighter. If $G_{ij} - g$, the image is getting darker.

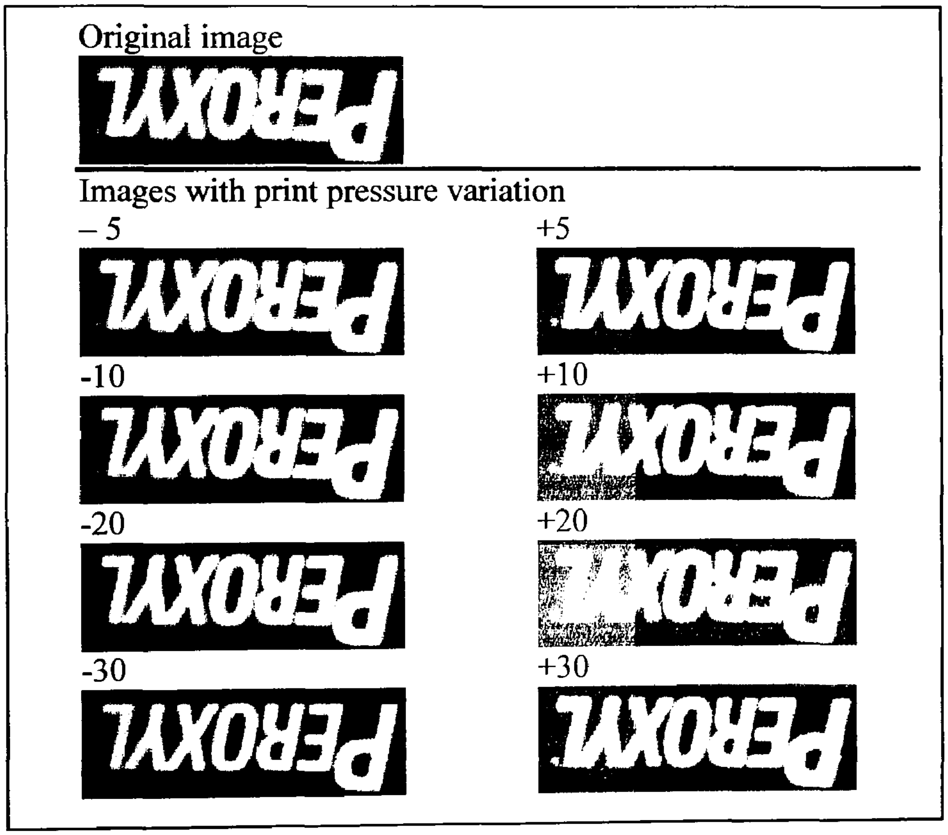


Figure 5.5 Print Pressure Variation

Figure 5.5 shows the sample pictures that simulate the print pressure variation. The gray values of the image pixels are subtracted or added by 5, 10, 15, 20 gray-level, respectively.

Table 5.3 Inspection Result for Print Pressure Variation

(darker)

Qualified Range	-5	-10	-20	-30
Row: 7.5818 - 12.9197	7.5065, 10.5148	6.8501, 9.3818	5.5209, 8.6564	3.9583, 8.5432
Column: 9.2172 - 10.9397	9.2050, 10.7391	9.1190, 9.5439	8.3408, 9.2978	6.8165, 9.2733
	Defect	Defect	Defect	Defect

(lighter)

Qualified Range	+5	+10	+20	+30
Row: 7.5818 - 12.9197	8.1875, 9.2557	8.1746, 11.9618	7.4327, 10.1979	6.7787, 9.1860
Column: 9.2172 - 10.9397	5.8627, 10.6697	9.2549, 10.5763	9.1942, 10.4085	9.0909, 9.4407
	Pass	Pass	Defect	Defect

According to the simulation of the images with print pressure variation as shown in Figure 5.5, the testing results are given in Table 5.3. The result implies that sensitivity of inspection also depends on the original print pressure of the image. If the main content of the image is white, it will be more sensitive than darker.

5.2.4 Ink Rubbing Effect

It assumes that it happens along the width direction, because that is the running direction of the web. For the pixels in a row, the first pixel gray value are kept same; the second pixel gray value will be changed according to itself and the first pixel gray value; the third pixel gray value will be changed according to itself and the second pixel gray value. The rest of pixels in the row could be changed in this way. These processes will continue until all the rows done. The detail calculation equation is given below if calculation starts from the right side of the image:

$$G_{ij} = G_{i(j+1)} * percent + G_{ij} * (1 - percent) \tag{5.3}$$

Where G_{ij} denotes the pixel gray value. Subscript i and j denote the coordinate of the pixel: i denotes the ith row, j denotes the jth column. The *percent* is a certain value from 0% to 100% that represents the strength of ink rubbing.

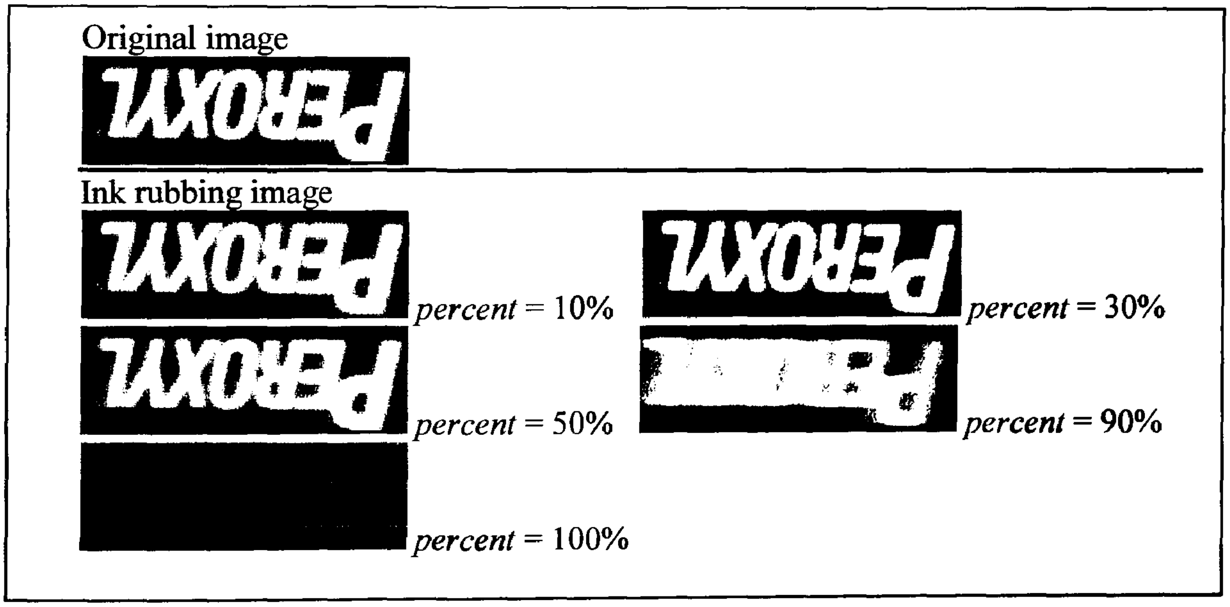


Figure 5.6 Images With Ink Rubbing

Figure 5.6 shows the sample pictures that simulate the ink rubbing effect. The inspection results are given in Table 5.4. It can be seen that if the rubbing effect happens on entire image surface, it will be easily recognized by the Neural Network method.

Table 5.4 Inspection Result For Ink Rubbing

Image size: 153x52

Qualified Range	*percent* = 10%	*percent* = 30%	*percent* = 50%	*percent* = 90%
Row: 7.5818 - 12.9197	8.3738, 10.7250	8.6674, 10.7249	9.0913, 10.7318	2.7462, 10.6861
Column: 9.2172 - 10.9397	9.2655, 10.9410	9.3001, 10.9410	9.3952, 10.9401	6.2491, 10.9215
	Defect	Defect	Defect	Defect

5.2.5 Background Variation

This simulation is carried out with the same defects on the different image backgrounds, including pure white background and pure black background. To simplify the simulation, Photoshop CS is used to change the image background.

Testing for pure black background

Shown in Figures 5.7 are five sample images with backgrounds changed to **pure black.** These are recognized as qualified images and have been registered already. They are used as golden images for live image inspection.

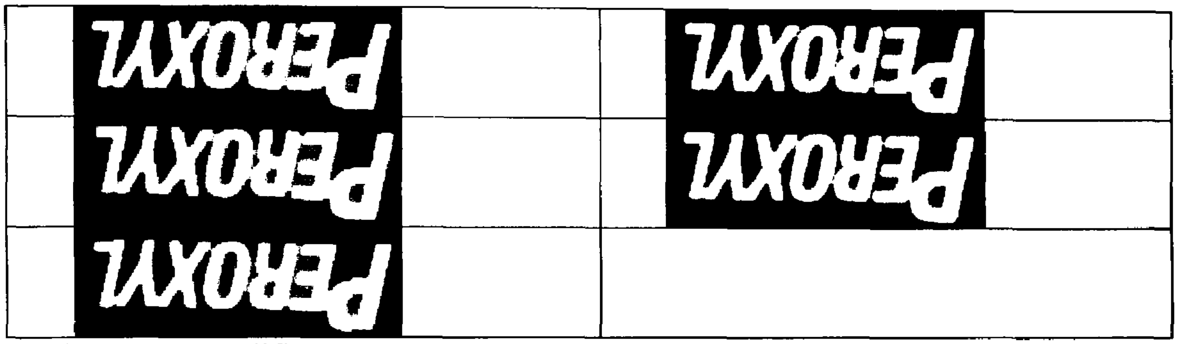


Figure 5.7 Qualified Sample Images For Inspection (Pure Black Background)

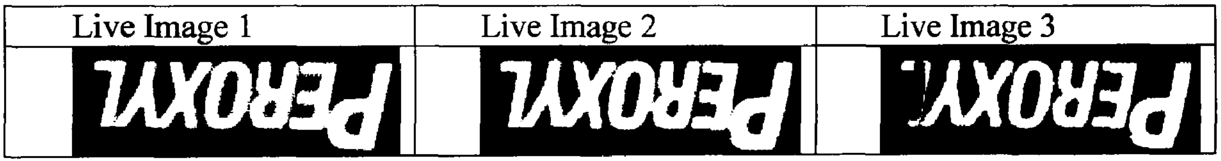


Figure 5.8 Live Images To Be Inspected (Pure Black Background)

Figure 5.8 shows three live images under inspection with backgrounds changed to pure black:

- Live Image 1 is a perfect image that is deemed as a qualified image;
- Live Image 2 is dirty image that is deemed as a defective image (a little dirty on the area of white letter).
- Live Image 3 is modified artificially that is a defective image.

The testing result is given in Table 5.5:

Table 5.5 Inspection Result For Pure Black Background

Qualified Range	Live Image 1	Live Image 2	Live Image 3
Row: 9.1769 - 44.1790	16.9674, 44.1790	19.1686, 44.1790	7.2418, 44.1790
Column: 3.9447 - 45.0815	11.9068, 45.0815	16.1089, 45.0815	2.8321 45.0815
	Pass	Pass	Defect

Testing for pure white background

Figure 5.9 shows five sample images with backgrounds changed to pure white. They are recognized as qualified images and have been registered already. They are used as the golden images for live image inspection.

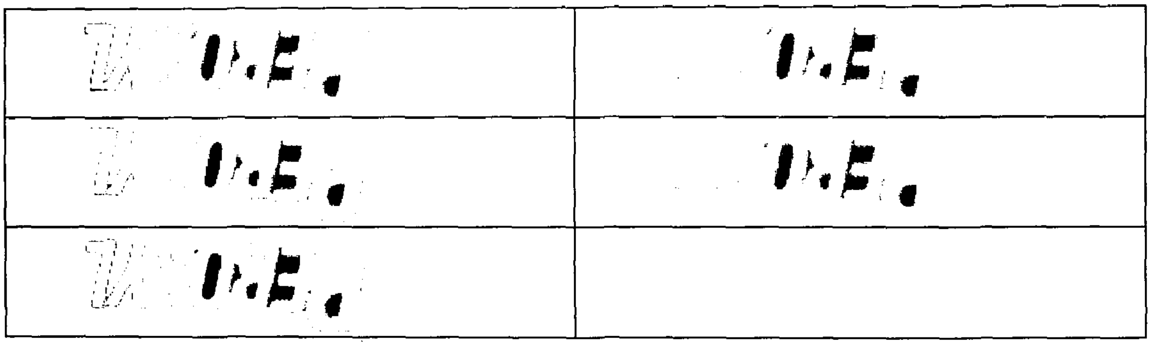


Figure 5.9 Qualified Sample Images For Inspection (Pure White Background)

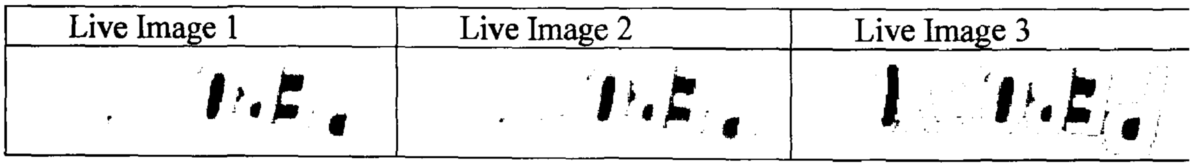


Figure 5.10 Live Images To Be Inspected (Pure White Background)

Figure 5.10 shows three live images under inspection with backgrounds changed to pure white:

- Live Image 1 is a perfect image that is deemed as a qualified image;
- Live Image 2 is dirty image that is deemed as a defect image (a little dirty on the area of white letter).
- Live Image 3 is modified artificially that is deemed as a defect image.

The testing result is given in Table 5.6.

Table 5.6 Inspection Result For Pure White Background

Qualified Range	Live Image 1	Live Image 2	Live Image 3
Row: 8.9011 - 18.9456	8.7836 , 18.9454	7.5718 18.9454	-9.3406, 18.9454
Column: 8.5918 - 18.3780	8.4702, 18.3777	7.6082, 18.3777	-8.8541 18.3777
	Defect	Defect	Defect

5.2.6 Computational Time on Two Indices

Table 5.7 shows the computational time for calculating the two image Indices. The testing result is based on a desktop computer: CPU: AMD Althlon 2400+; Memory: 512M.

Table 5.7 Indices Computation Time

Image Size (pixel)	128x128	256x256	512x512	1024x1024
Row index calculation time	< 1ms	< 1 ms	10 ms	20 ms
Column index calculation time	< 1ms	10 ms	70 ms	290 ms

The gray-values of the pixels are saved in the memory row by row. The column index calculation is processed through the column direction so that it needs more time to seek for the memory location. That is why the row index calculation is faster than the column index calculation

5.3 Accuracy Verification

The images used for testing were captured on line using the SC. Totally 14 images (shown in Figure 5.11) were used for the Neural Networks training, and 58 images were used for inspection tests, among which 19 are with defects. Figure 5.12 shows the inspection results. The arrow signs point out the false calls.

From the test results, it can be seen that 18 defect images are detected as defective, with only 1 missed because the defect portion is very tiny and located in the black background area, as illustrated in Figure 5.13(a). Also, there is a false qualification, as shown in Figure 5.13 (b). In total, there are 2 false calls out of 58, and the accuracy is around 96%.

Based on the same image samples, the inspections are implemented again with pure statistic method. Figure 5.14 shows the results. The arrow signs point out the false calls.

It can be seen that 4 qualified images are detected as defective, and 1 defective image is not been detected. Totally, there are 5 false calls out of 58, and the accuracy is around 91%.

Summary

In this chapter, the new method proposed in this thesis has been verified on the industrial inspection machine. Based on the same inspection samples, the results show that the new method has higher inspection accuracy (96%) than the accuracy of traditional statistic method (91%).

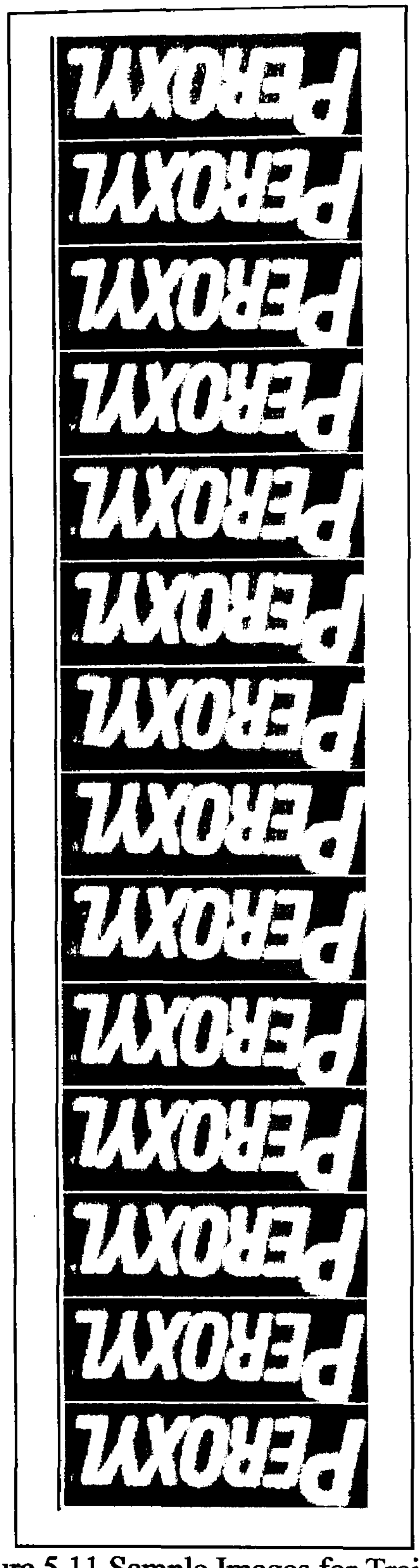


Figure 5.11 Sample Images for Training

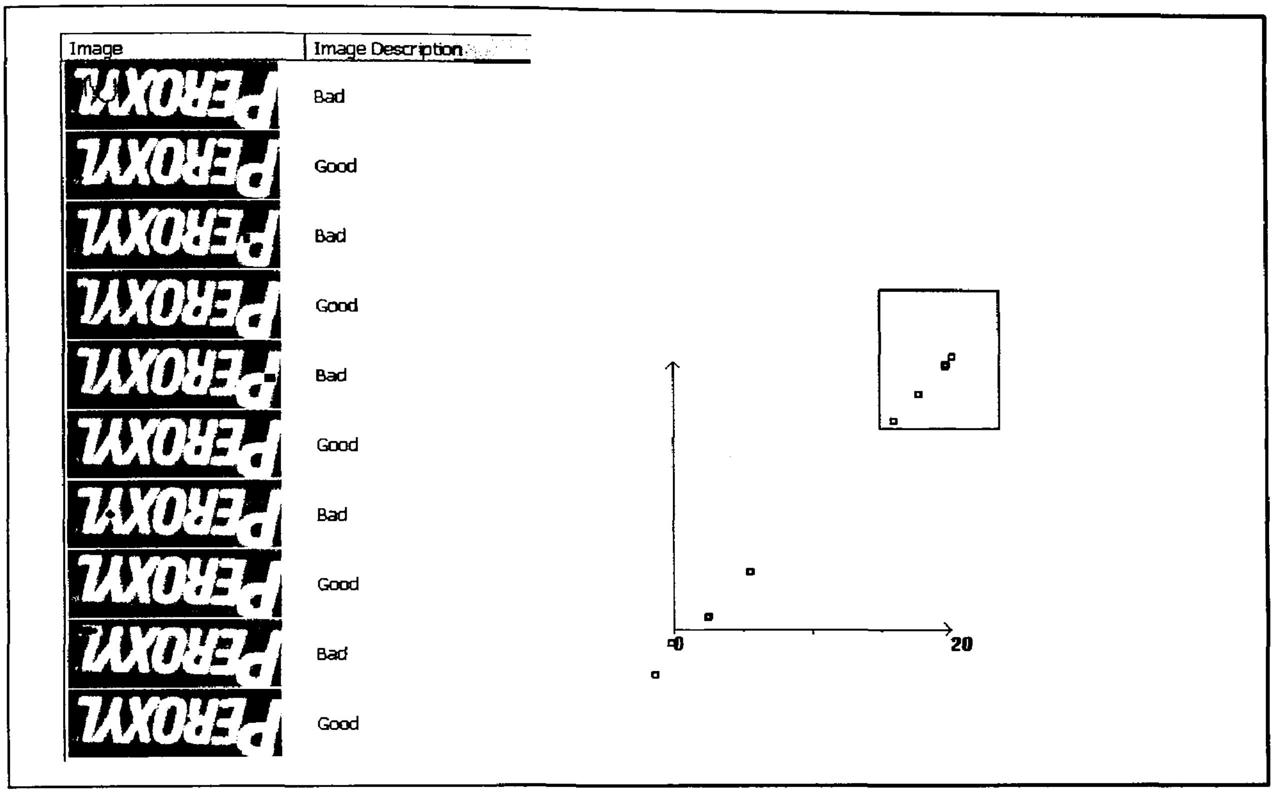


Image	Image Description
	Bad
	Good
	Bad
	Good
	Bad
	Good
	Bad
	Good
	Bad
	Good

Figure 5.12 (a) Inspection of 1st set of 10 images

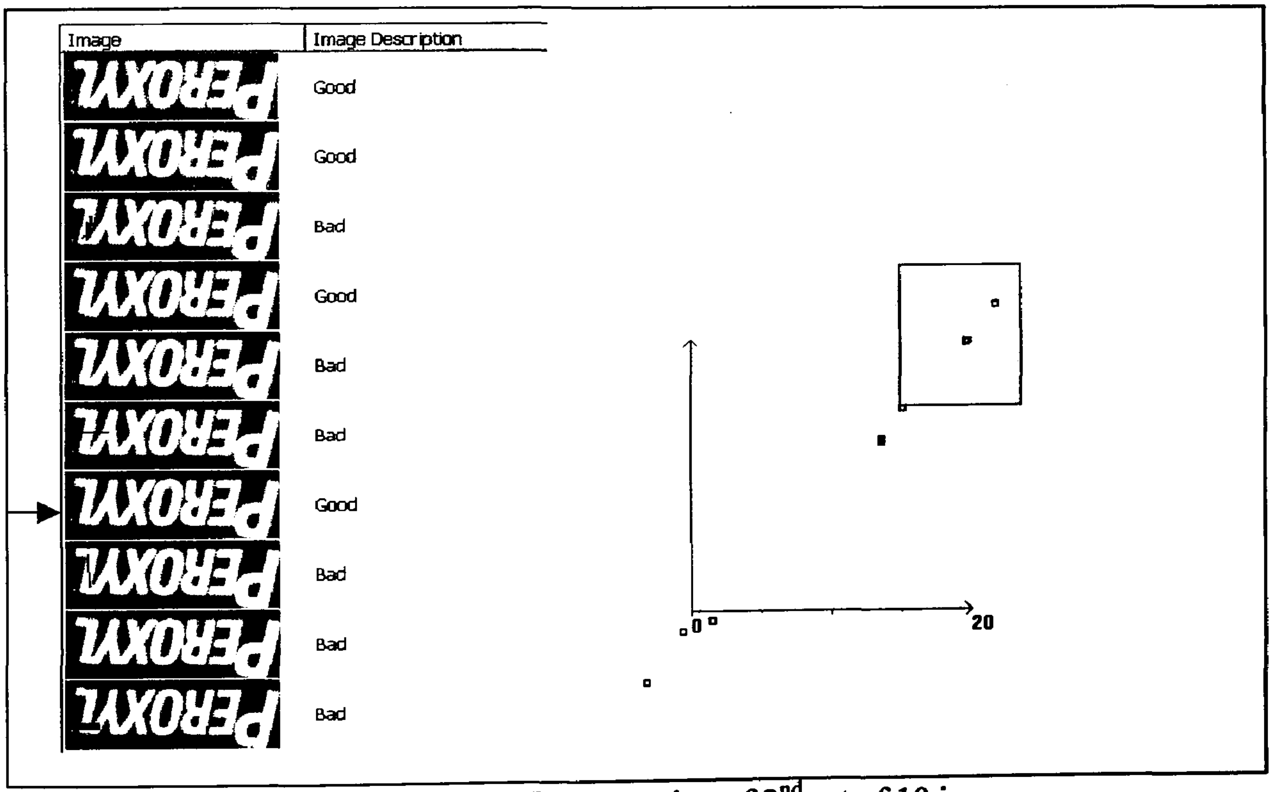


Image	Image Description
	Good
	Good
	Bad
	Good
	Bad
	Bad
	Good
	Bad
	Bad
	Bad

Figure 5.12 (b) Inspection of 2nd set of 10 images

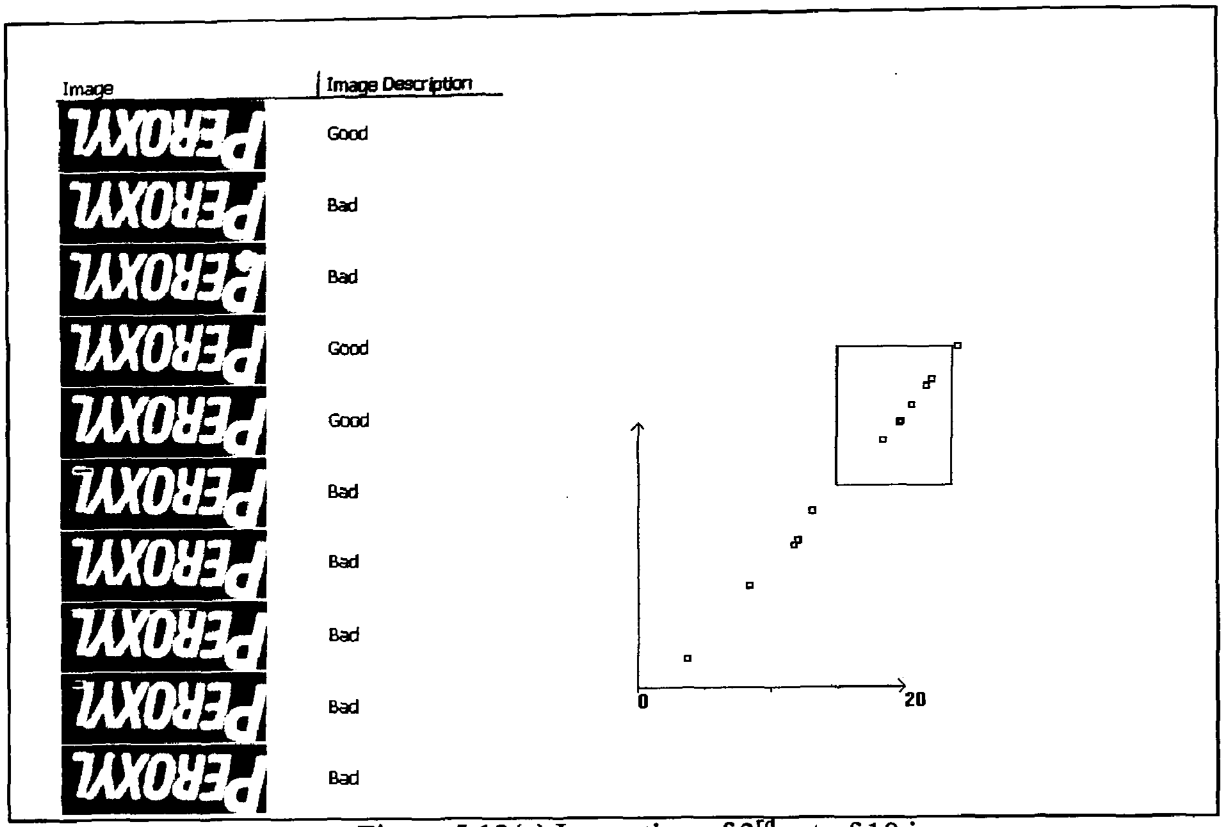


Image	Image Description
	Good
	Bad
	Bad
	Good
	Good
	Bad
	Bad
	Bad
	Bad
	Bad

Figure 5.12(c) Inspection of 3^{rd} set of 10 images.

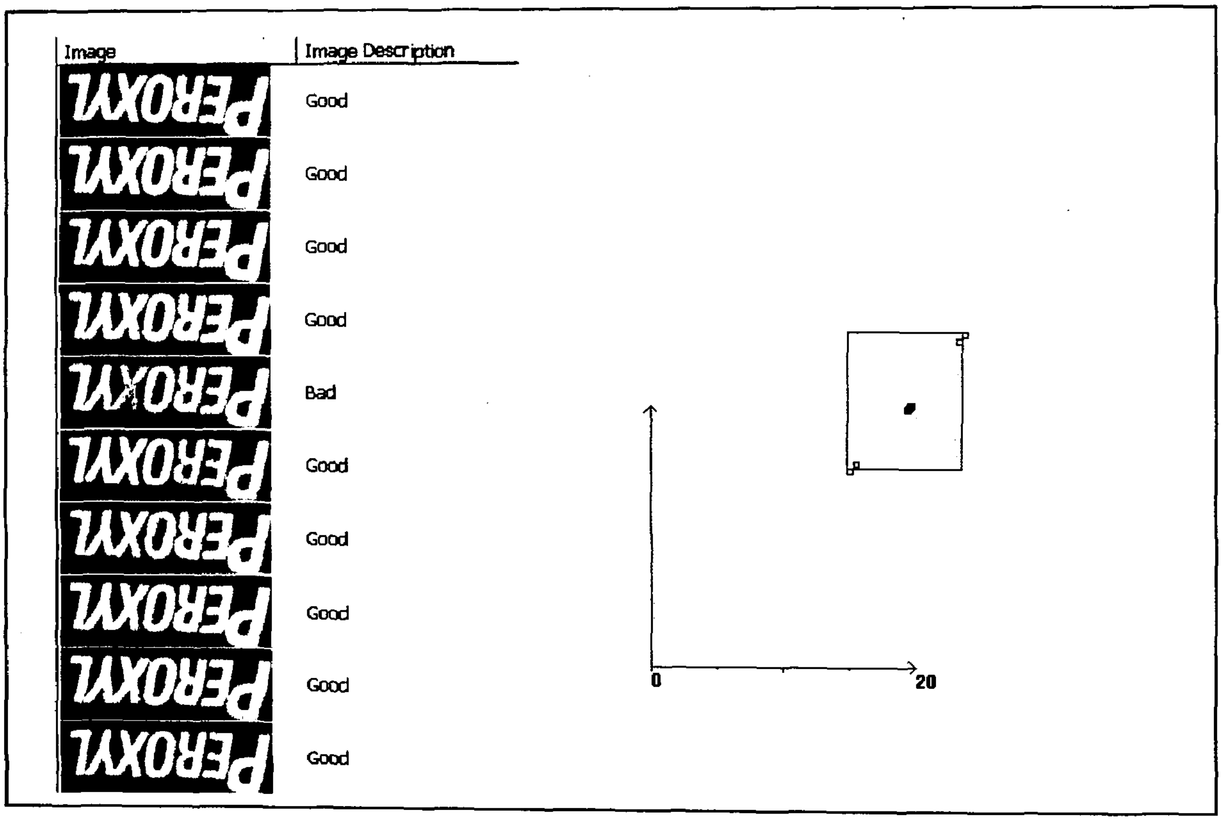


Image	Image Description
	Good
	Good
	Good
	Good
	Bad
	Good
	Good
	Good
	Good
	Good

Figure 5.12 (d) Inspection of 4^{th} set of 10 images

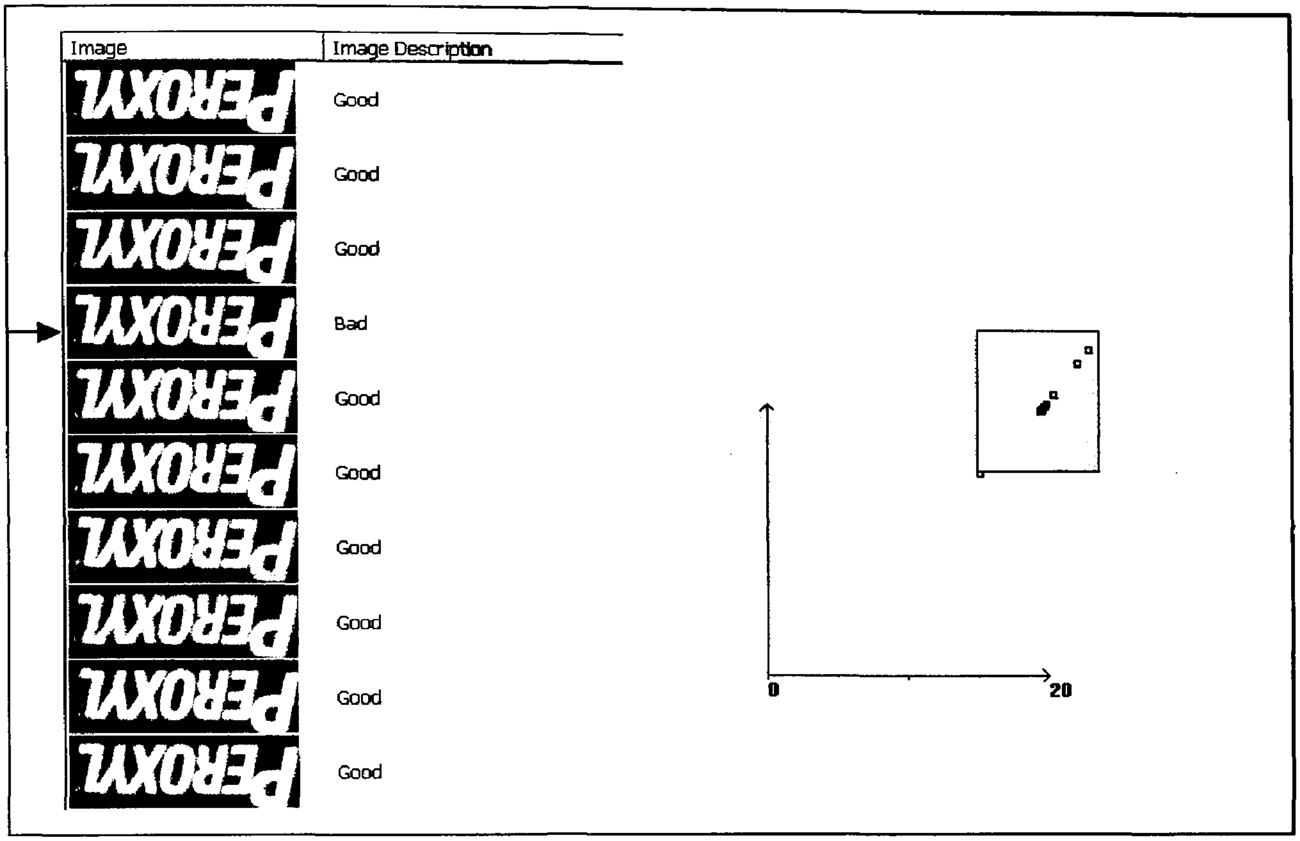


Image	Image Description
	Good
	Good
	Good
	Bad
	Good
	Good
	Good
	Good
	Good
	Good

Figure 5.12 (e) Inspection of 5th set of 10 images

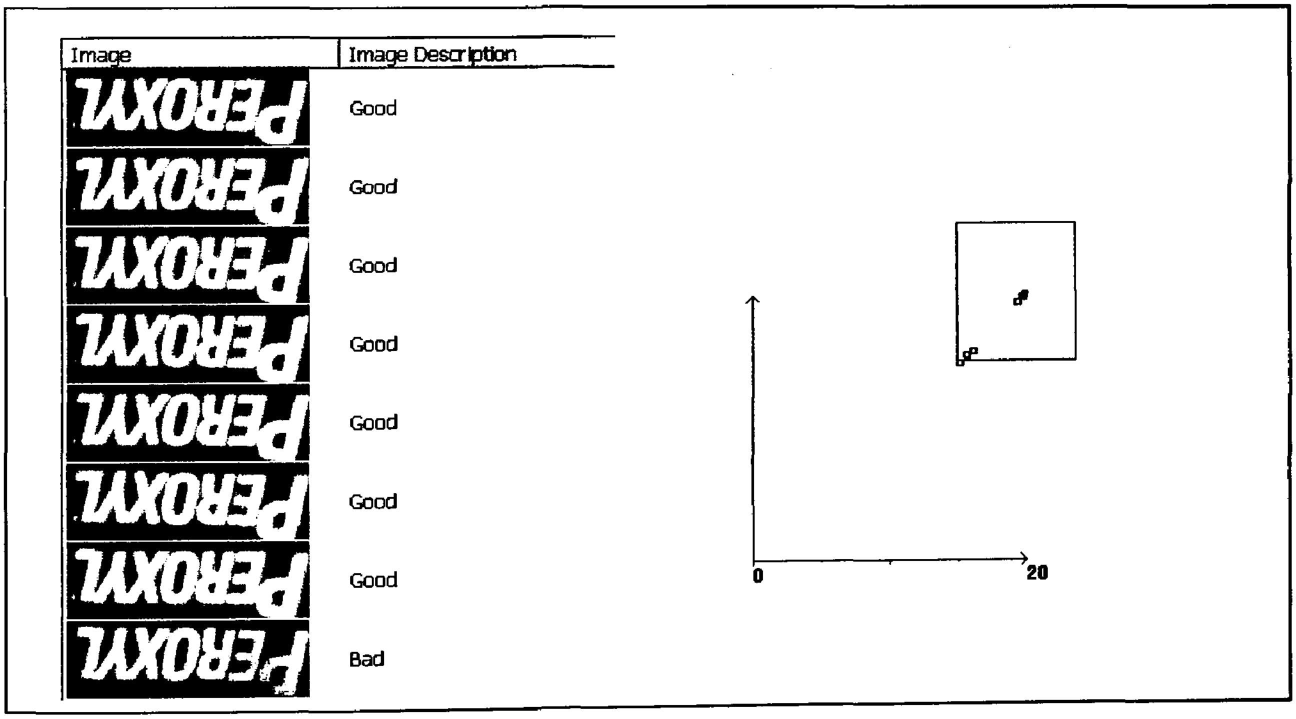


Image	Image Description
	Good
	Good
	Good
	Good
	Good
	Good
	Good
	Bad

Figure 5.12 (f) Inspection of 6th set of 8 images

Figure 5.12 Live Image Inspection Results (Statistic Based Neural Network Method)

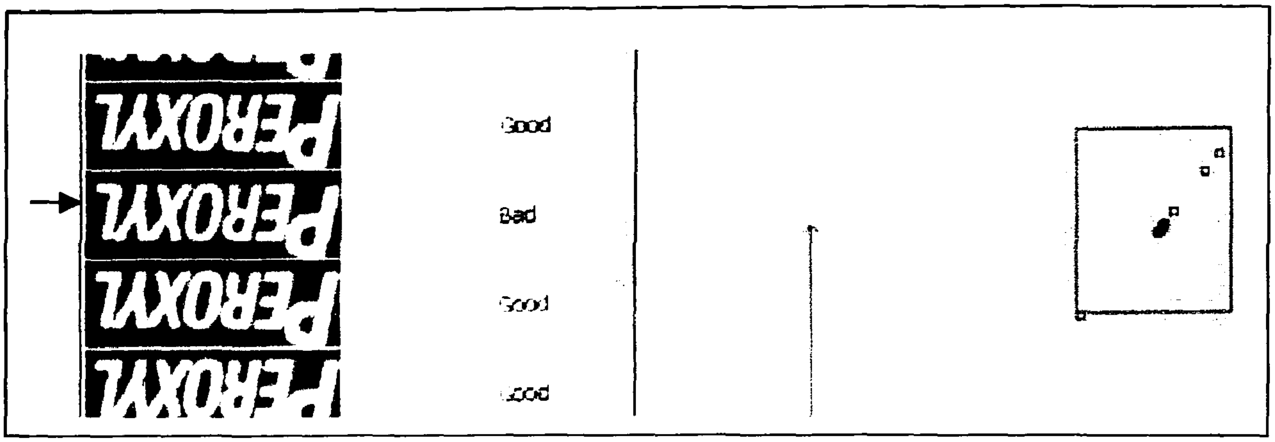


Figure 5.13 (a) Missing Call

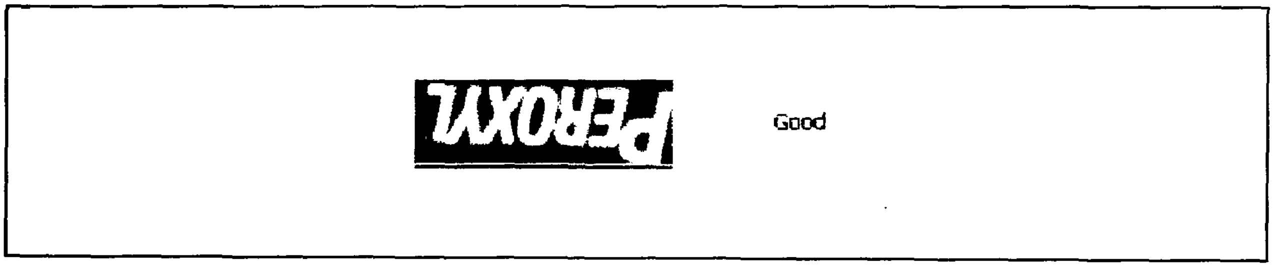


Figure 5.13 (b) False Call

Figure 5.13 Two Wrong Detections Figure

Image	Image Description
	Bad
	Bad
	Bad
	Good
	Bad
	Good
	Bad
	Good
	Bad
	Good

Image	Image Description
	Good
	Good
	Bad
	Good
	Good
	Bad
	Bad
	Bad
	Bad
	Bad

Image	Image Description
	Bad
	Bad
	Bad
	Good
	Good
	Bad
	Good
	Bad
	Bad
	Bad

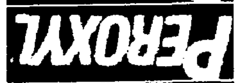
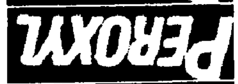
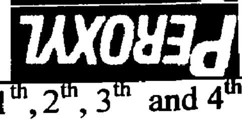
Image	Image Description
	Good
	Good
	Good
	Good
	Bad
	Good
	Good
	Good
	Good
	Good

Figure 5.14 (a) Inspection of 1th,2th, 3th and 4th set of 40 images

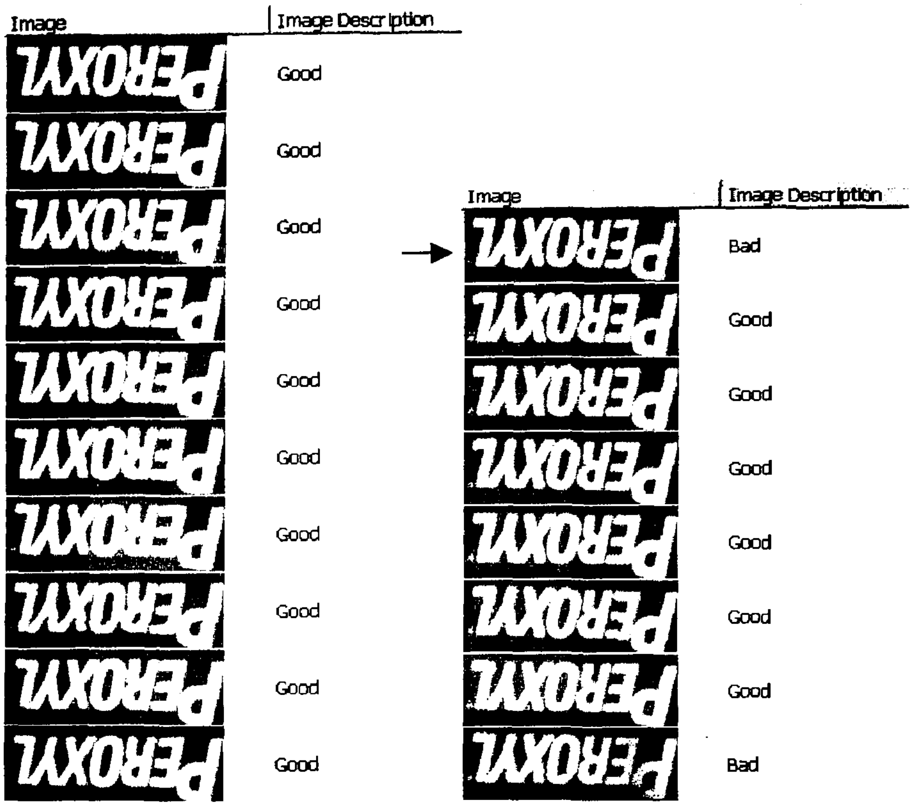


Figure 5.14 (b) Inspection of 5th and 6th set of 18 images

Figure 5.14 Live Image Inspection Results (Pure Statistic Method)

6. Conclusions

In this thesis, a statistic based Neural Network method is proposed for industrial image inspection. The whole algorithm has been implemented and verified on the real industrial inspection machine at Rotoflex Ltd. According to the result, the inspection accuracy is around 96% and higher than the accuracy of traditional statistic method (91%).

The contributions of new inspection method are summarized below:

1. Statistics Based NN Method

Most of AVI systems don not consider the real world uncertainties. The new method is to account for uncertainties that would affect the accuracy of the AVI system. To get rid of the effect of uncertainties, a hybrid method is proposed by combining a statistics method with a NN method. The statistics method is used to account for random uncertainties while used to compute the two variances for the establishment of a tolerance zone. The NN method is used to train the inspection system to account for the systematic uncertainties.

When inspecting images, first Neural Networks are trained using two indices from a set of qualified images in order to establish a tolerance zone. Then, the two indices of each inspection image are computed through trained Neural Networks and compared with the tolerance zone. A defective item is detected if either index falls out of the tolerance zone. The new system is very suit for the quality control in field of printing, and fabric graphic inspection

2. Two Indices (Row Variances and Column Variances)

The indices of image are defined and used for NN training and inspection instead of pixel or feather of image, which are the variances of the rows and columns of the image. Since variance measures the deviation from the mean value, it is selected to establish a tolerance zone that defines an allowable area within which the items under inspection are considered acceptable. Considering that images are 2-dimensional signals, the variances of the rows and the columns are selected as the two indices. The minimum and maximum values of the two indices provide four corner points that can form a rectangular tolerance zone in the 2-D plane. Comparing with traditional pixel-based or feather-based inspection method, indices-base method improves the speed of data processing and makes the online inspection possible.

3. Two Point Registration Method

As the image registration is the important pre-processing step of image inspection. A two-point based registration method is implemented. The concept of this method is a hybrid of landmark-based method and axes/moments based method. This method is based on computing the center positions of two letter images. By using two points each on the base image and registration image, two corresponding lines can be identified on the two images and used to register them both in position and orientation. It is very simple and computationally efficient.

4. Defect Simulation Algorithms and Methods

To verify the effect of image inspection method, large numbers of defect simulations have been made, including simulation of stretch, shift, printing pressure variation, ink rubbing, background variation, and different shape of smear effect. All the simulation algorithms and methods have been implement through c++ program.

5. Software Development of Inspection System

Based on the proposed method, a complete software package of AVI system has been developed and tested on the real industrial image inspection machine.

The functions of software package include: image acquisition from CCD camera, image registration, NN training and image inspection.

Extensive tests were carried out to verify the hybrid method, and a comparison was made to the pure statistics method. The results show that the proposed hybrid method has an accuracy of 96%, which is higher than the accuracy of traditional statistic method of 91%.

Future Work

Since the simulation results gave us strong evidence that new AVI system can obtain a good generalization performance and high accuracy. Some potential future work directions are listed as in the following:

1) More testing items should be done based on exhaustive and realistic images. Because this new system is proposed for industrial image inspection, the simulated defect images could not fully represent the uncertain conditions of real environment.

2) Index calculation could be implanted to FPGA to further increase computation speed.

3) To upgrade the qualified zone by training sample images that could include simulated uncertain effects such as stretch.

Bibliography

[1] D.T. Pham, and R.J. Alcock, Automated Visual Inspection of Birth Wood Boards, *Journal of Process Mechanical Engineering*, Vol.210, No.1, pp.45-52, 1996.

[2] P. Pemer, An Architecture For A Knowledge - Based Image Inspection System, *Proceedings of International Symposium on Speech, Image Processing and Neural Networks,* Vol.1, pp.65-68, 1994.

[3] D. Deridder, R.P.W. Duin, P.W. Verbeek, and L.J. Vanvliet., *The Applicability of Neural Networks to Non-linear Image Processing*, Springer-Verlag London Ltd, 1999.

[4] R. Chellappa, and K. Fukushima, *Guest Editorial,* Applications of Artificial Neural Networks to Image Processing, *IEEE Transactions on Image Processing*, Vol.7, No.8, pp.1093-1096, 1998.

[5] Y. Hara, R.G. Atkins, S.H. Yueh, R.T. Shin, and J.A. Kong, Application of Neural Networks to Radar Image Classification, *IEEE Transactions on Geoscience and Remote Sensing*, Vol.32, No.1, pp.100-109, 1994.

[6] A. Zaknich, Introduction to The Modified Probabilistic Neural Network for General Signal Processing Applications, *IEEE Transactions on Signal Processing*, Vol.46, No.7, pp.1980-1990, 1998.

[7] H.A. Rowley, S. Baluja, and T. Kanade, Neural Network-Based Face Detection, *Proceedings of CVPR '96 - IEEE Computer Society Conference*, pp.203-208, 1996

[8] H.S. Ranganath, and G. Kuntimad, Image Segmentation Using Pulse Coupled Neural Networks, *Proceedings of IEEE International Conference,* Vol.2, pp.1285-1290, 1994

[9] S.Y. Kung, Decision-Based Neural Networks with Signal/Image Classification Applications, *IEEE Transactions on Neural Networks*, Vol.6, No.1, pp.170-181, 1995

[10] H.B. Burke, D.B. Rosen, and P.H. Goodman, Comparing Artificial Neural Networks to Other Statistical Methods for Medical Outcome Prediction, *Proceedings of IEEE International Conference*, pp.2213-2216, 1994

[11] J.R. Bittencourt, and F.S. Osorio, Adaptive Filters for Image Processing Based on Artificial Neural Networks, *IEEE Computer Society*, pp.336-336, 2000

[12] M. Egmont-Petersena, D. Deridderb, and H. Handelsc, Image Processing with Neural Networks – A Review, *Journal of Pattern Recognition*, Vol.35, pp.2279-2301, 2002

[13] C. Bahlmann, G. Heidemann, and H. Ritter, Artificial Neural Networks for Automated Quality Control of Textile Seams, *Journal of Pattern Recognition*, Vol.32, No.6, pp.1049-1060, 1999

[14] E.N. Malamas, E.G.M. Petrakis, M. Zervakis, L. Pettit, J. Legat, Industrial Vision: Systems, Tools and Techniques, *Journal of Image and Vision Computing*, Vol.21, No.2, pp.171-188, 2003

[15] D.S. Sohi, and S.S. Devgan, Application to Enhance The Teaching And Understanding of Image Processing Techniques, *Proceedings of the IEEE International Conference*, pp.413-416, 2000

[16] J.L.C. Sanz, D. Petkovic, Machine Vision Algorithm for Automated Inspection of Thin-Film Disk Heads, IEEE Transaction on PAMI, Vol.10, pp.830-848, 1988.

[17] T. Torres, J.M. Sebastian, R. Aracil, L.M. Jimenez, and O. Reinoso, Automated Real-Time Visual Inspection System for High-Resolution Superimposed Printings, *Journal of Image and Vision Computing*, vol.16, pp.947-958, 1998.

[18] M. Moganti, F. Ercal, C.H. Dagli, S. Tsunekawa, Automatic PCB Inspection Algorithms: A Survey, *Journal of Computer Vision and Image Understanding*, Vol.63, No.2, pp.287-313, 1996

[19] R.T. Chin, C.A. Harlow, Automated Visual Inspection: A Survey, *IEEE Transactions on Pattern Analysis and Mechanical Intelligence*, Vol.4, No.6, pp.557-573, 1982

[20] A.M. Darwish, A.K. Jain, A Rule-based Approach for Visual Pattern Inspection, *IEEE Transactions on Pattern Analysis and Mechanical Intelligence*, Vol.10, No.1, pp.56-58, 1988

[21] H. Yoda, Y. Ohuchi, Y. Taniguchi, M. Ejiri, An Automatic Wafer Inspection System Using Pipelined Image Processing Techniques, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol.10, No.1, pp.4-16, 1988

[22] H. Freeman, M.Y. Chiu, D.D. Dreyfuss, I. Gorog, *Machine Vision for Inspection and Measurement*, Academic Press Inc., 1989

[23] S, Haykin, *Neural Networks*, 2nd Edition, Prentice-Hall Inc., 1999.

[24] W.K. Pratt, *Digital Image Processing*, A Wiley-Interscience Publication, 1978

[25] J. Heikkonen, J. Lampinen, Building Industrial Applications with Neural Networks, *Proceedings of European Symposium on Intelligent Techniques*, pp.19-20, 1999

[26] J. Lampinen, S. Smolander, M. Korhonen, Wood Surface Inspection System Based on Generic Visual Features, *Journal of Industrial Applications of Neural Networks*, pp.35-42, 1998

[27] M. Raus, O. Brenner, W. Ameling, Neural Quality Inspection in Industrial Compact Disc Print Stations, *Proceedings of Second International Conference on Intelligent Systems Engineering*, pp.154-158, 1994

[28] Y.A. Karayiannis, R. Stojanovic, P. Mitropoulos, C. Koulamas, T. Stouraitis, S. Koubias, G. Papadopoulos, Defect Detection and Classification on Web Textile Fabric Using Multi-resolution Decomposition and Neural Networks, *Proceedings of IEEE International Conference on Electronics, Circuits and Systems*, Vol.2, pp.765-768, 1999

[29] V. Sankaran, B. Chartrand, D.L.H. Lillard, M.J. Embrechts, R.P. Kraft, Automated Inspection of Solder Joints - A Neural Network Approach, *Proceedings of Electronics Manufacturing Technology Symposium*, pp.232-237, 1995

[30] G.P. Zhang, Neural Networks for Classification: A Survey, *IEEE Transactions on Systems, Man and Cybernetics*, Vol.30, No.4, pp.451-462, 2000

[31] J.B.A. Maintz, M.A. Viergever, An Overview of Medical Image Registration Methods, *Journal of Medical Image Analysis*, Vol.2, No.1, pp.1-36, 1998

[32] B. Zitova, J. Flusser, Image Registration Methods: A Survey, *Journal of Image and Vision Computing*, Vol.21, pp.977-1000, 2003

[33] D. Robinson, P. Milanfar, Fundamental Performance Limits in Image Registration, *IEEE Transactions on Image Processing*, Vol.13, No.9, pp1185-1199, 2004