5-1991

An approach to fast detection of defects in metal surfaces

Iftekhar Hussain

To the Graduate Council:

I am submitting herewith a thesis written by Iftekhar Hussain entitled "An approach to fast detection of defects in metal surfaces." I have examined the final electronic copy of this thesis for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Master of Science, with a major in Electrical Engineering.

Dragana Brzakovic, Major Professor

We have read this thesis and recommend its acceptance:

D. W. Bouldin, M. O. Pace

Accepted for the Council:
Carolyn R. Hodges

Vice Provost and Dean of the Graduate School

(Original signatures are on file with official student records.)

To the Graduate Council:

I am submitting herewith a thesis written by Iftekhar Hussain entitled "AN APPROACH TO FAST DETECTION OF DEFECTS IN METAL SURFACES." I have examined the final copy of this thesis for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Master of Science, with a major in Electrical Engineering.

Dragana Brzakovic, Major Professor

We have read this thesis
and recommend its acceptance:

Marshall Pace

Donald W. Bouldin

Accepted for the Council:

CwMinkel

Vice Provost
and Dean of The Graduate School

STATEMENT OF PERMISSION TO USE

In presenting this thesis in partial fulfillment of the requirements for a Master's degree at The University of Tennessee, Knoxville, I agree that the Library shall make it available to borrowers under rules of the Library. Brief quotations from this thesis are allowable without special permission, provided that accurate acknowledgment of the source is made.

Permission for extensive quotation from or reproduction of this thesis may be granted by my major professor, or in his absence, by the Head of Interlibrary Services when, in the opinion of either, the proposed use of the material is for scholarly purposes. Any copying or use of the material in this thesis for financial gain shall not be allowed without my written permission.

Signature Hussain

Date 3/19/91

AN APPROACH TO FAST DETECTION OF DEFECTS IN METAL SURFACES

A Thesis

Presented for the

Master of Science

Degree

The University of Tennessee, Knoxville

Iftekhar Hussain

May 1991

DEDICATION

This thesis is dedicated to my mother and brothers.

ACKNOWLEDGEMENTS

I wish to express my respect and gratitude to my major advisor, Dr. D. Brzakovic, for her support, advice, encouragement, and patience during the course of this research. I am also indebted to Dr. D. W. Bouldin and Dr. M. O. Pace for serving on my committee and reviewing this thesis.

I would also like to express my appreciation to a number of individuals, for their help and encouragement. In particular, I am very thankful to Hal E. Beck, Hamed Sari-Sarraf, Chandra Tan, Paul B. Davis, Mohammad Abdulghafoor, Mourad B. Takla and Hrishikesh Gadagkar who helped me every time I found myself in a difficulty. I also wish to thank Bassam Awad, Wassim Hossari, Dongmei Wang, Don Mize, Ming Xu, and Siva Uppalapati for their encouragement. Special thanks go to J. L. Daniel and Mary Neitling for proofreading and helping with the various aspects of this thesis.

In addition, I express my appreciation to the Center for Measurement and Control at The University of Tennessee for sponsoring this research.

Finally, I thank my mother, my brothers, and other family members for showing their continuous support and standing by me during this period of my graduate study.

ABSTRACT

In this thesis an approach for the fast detection and classification of defects in digitized images of surfaces with known intensity distributions is presented. The reflectance response of specular planar surfaces is modeled. It is shown that by proper selection of lighting, non-linear intensity distributions of metal surfaces can be approximated by linear intensity distributions. Two kinds of defects, streaks and speckles, are considered for detection and classification. The approach presented in this thesis has two parts: defect extraction and defect classification. The defect extraction unit generates defect signatures that are input to the classifier. A multilayer perceptron applying the back-propagation rule is utilized for classification of the defect signatures. When applying the proposed approach, more than 95% of the defects are detected and 90% of the detected defects are classified correctly.

TABLE OF CONTENTS

CHAPTER PAGE

CHAPTER PAGE

LIST OF FIGURES

FIGURE **PAGE**

FIGURE PAGE

LIST OF TABLES

CHAPTER 1

Introduction

With the technological advancements in the image processing/pattern recognition area, automated visual inspection systems for online inspection have received increased attention [2,3,5,6,10,15,17,26,28]. The reasons for the increased interests are the following:

- the need to reduce manufacturing costs and make the inspection process more robust and,
- the ability to make online decisions about the quality of the product.

These developments in image processing and computer technologies have made it possible to implement automated visual inspection systems. Moreover, mass production of economical microprocessors and the rapid growth of VLSI technology have made visual inspection systems feasible for many ordinary factories.

Many products such as metals (e.g., aluminum), magnetic tapes, films, paper, etc., are manufactured at very high speeds, usually 600 feet or more per minute [10]. Most of the defects in such products are closely related to the parameters of the manufacturing process. The quality of the product can be improved if defects can be detected online and this information can be fed back to control the process parameters. Furthermore, with online inspection, the defective materials can be marked while being manufactured. The ability to mark defective materials online helps to develop strategies and systems that minimize waste, improve quality, and reduce the cost of manufactured products.

However, one of the problems in the application of image processing techniques to automize the visual inspection tasks is processing speed. Online automated visual inspection requires very fast computations for real-time applications.

This thesis presents an approach for the fast detection and classification of defects in digitized images of surfaces with known intensity distributions. The reflectance response of specular planar surfaces is modeled using Phong's reflectance model. Simulations of specular metal surfaces such as aluminum and nickle show that the metal images have non-linear intensity distributions. However, the non-linear intensity distributions of the metal surfaces can be approximated by the linear intensity distributions, provided lighting conditions are selected appropriately (Chapter 2). In this thesis, we are going to consider only the metal images with linear intensity distributions. Two kinds of defects, namely streaks and speckles, are considered for detection and classification. The approach described in this thesis for fast defect detection and classification has two stages: defect extraction and defect classification.

The purpose of the defect extraction stage is to identify regions of abrupt intensity changes in the input image and to generate defect signatures. The defect signature generation involves subdividing the image into non-overlapping regions called windows, generating horizontal and vertical signatures of the image by adding the row and column intensities within each window respectively, and analyzing these signatures for abrupt changes. Horizontal and vertical signatures are smooth for an image with uniform intensity. However, if the image contains defects, there is a spike in the signature at the location of the defect. Thus, the presence of defects is detected by observing these abrupt changes in the image signatures. Next, the defect signatures are obtained by setting to zero those elements of the horizontal and vertical signatures that are below a certain threshold. The threshold is based on the average intensity of the window. Thus, the defect signatures contain only the abrupt intensity changes. The defect signatures are then input to the classifier.

Defect classification is accomplished by using a multilayer perceptron network that uses the back-propagation error learning rule. The classifier has two modes of operation: training and testing. The training mode involves presenting to the network the pattern vectors (defect signatures) that are representative of defect signatures of streak and speckle. The correct class label for the each input pattern vector is provided externally by the teacher to the network to adjust its weights in such a way that the mean square error between the actual classifier output and the desired output is minimized. After the network has learned the representative pattern vectors of streak and speckle, the network is used in the testing mode. During the testing mode, pattern vectors of streak and speckle that are not seen by the network are input to the network, and the network provides a class label for each of the inputs.

The major advantage of the proposed approach for defect detection and classification relative to the traditional approaches is the effective use of one-dimensional signatures of a two-dimensional image that leads to less computations, hence, less processing time. An image is processed for defect detection using multiple window sizes. Thus, the presence of defects can be verified under multiple window sizes before defect signatures are passed on to the classifier that minimizes the false alarms. The computational efficiency, i.e., processing time, is improved by reusing the computations of the previous window size in the subsequent larger windows.

The remainder of this thesis is organized as follows. Chapter 2 describes the reflectance modeling of specular surfaces. The material in this chapter provides an insight into the reflectance response of planar illuminated specular surfaces. Chapter 3 describes the functional units of the defect detection and classification system. Furthermore, this chapter illustrates this system's performance by considering applications for synthetic and real images. Chapter 4 describes the multilayer perceptron classifier that is used for the classification of defects. Finally, concluding remarks are presented in Chapter 5.

CHAPTER 2

Reflectance Modeling of Specular Planar Surfaces

Our purpose for modeling a reflective surface (such as aluminum) is to determine the specular behavior of the surface under different lighting conditions. Typically, the specular behavior of a surface depends on the nature (diffuse light, spotlight) and position of the light source(s), and the material characteristics.

2.1 Surface Illumination

Illumination of a surface depends on the reflection and transmission of light as it interacts with the boundaries between materials (such as air and aluminum). A fundamental requirement for a valid model of an illuminated surface is the maintenance of energy equilibrium. The energy reflected from a surface plus the energy transmitted through the surface boundary must be equal to the energy that illuminates the surface. This relationship is given by the following equation:

$$\Phi_i = \Phi_r + \Phi_t, \tag{2.1}$$

where Φ_i is the incidence energy flux, Φ_r is the energy flux reflected from the surface, and Φ_t is the energy flux transmitted. Reflection and transmission are broken into two components, *a coherent* component and *an incoherent* component or scattered component. Detailed derivation of coherent and incoherent components can be found in [18,32,33]. The next section describes the reflectance models and the remainder of the chapter deals with development, implementation, and application of Phong's reflectance model.

2.2 Reflectance Models

Given a light source, a surface, and an observer, a reflectance model describes the intensity and the spectral composition of reflected light reaching the eye. The intensity of the reflected light is determined by the intensity of the light source and by the reflecting ability and surface properties of the material. Reflectance models used in practice differ greatly from those treated theoreticaly in [1,7,8,11,12,22,32,33]. Reflectance models can be categorized :

- Empirical models
- Transitional models, and
- Analytical models.

The shading techniques that evolved with these models can be classified as:

- Incremental
- Ray tracing
- Radiocity.

The shading technique determines how the model is applied to compute intensity at a point [14,16].

Early reflectance models were empirical in nature. They were evaluated after the geometry had been transformed into the perspective space. Transitional models require the use of Euclidean geometry prior to perspective transformation so that reflections, refractions, and shadows are geometrically correct. In analytical models, in addition to maintaining the true geometry, the movement of light energy through the environment must be modeled. In this work we will talk only about Phong's model. Phong's model is computationally tractable but is not very accurate. If more accurate and realistic results are needed, other models can be better candidates, but of course, they add increased complexity and more computations.

2.2.1 Phong's Reflectance Model

Phong's model is an empirical model that uses an incremental shading technique [27]. Phong's model is based on ambient, diffuse, and specular terms as follows:

$$I(\lambda) \quad = \quad K_a(\lambda) + K_d(\lambda)(\vec{N} \cdot \vec{L}) + K_s(\vec{R} \cdot \vec{L})^{N_s}, \qquad (2.2)$$

where

$K_{a,d,s}$ = ambient, diffuse, and specular reflectance, respectively,

$\vec{N}$ = surface normal,

$\vec{L}$ = light vector (surface to light),

λ = wavelength, and

N_s = specular exponent.

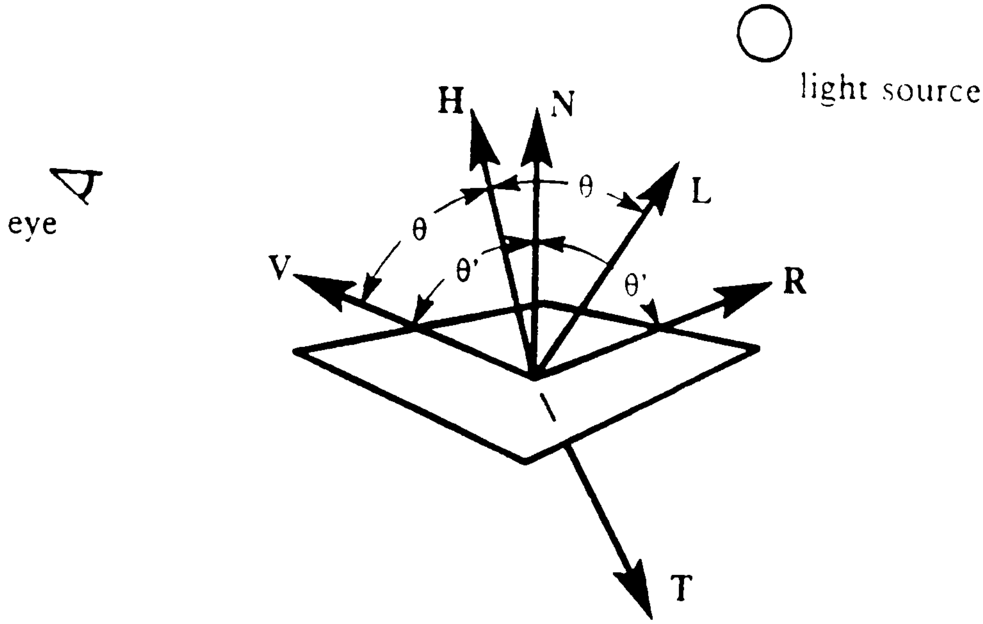
The ambient component represents light that is assumed to be uniformly incident from the environment and is reflected equally in all directions. The diffuse component represents light that is scattered equally in all directions. The specular component represents the highlights, light that is concentrated in the mirror direction. Note that in Phong's model, K_s is not a wavelength dependent function, thus, the highlights resulting from the application of Phong's model are always white. The geometry of Phong's model is given in Figure 2.1.

2.2.2 Improved Phong's Model

An alternate form of the specular reflection function proposed by Blinn [9] is given in the following expression:

$$\begin{aligned} I(\lambda) &= K_a(\lambda)I_a(\lambda) + K_d(\lambda)\sum_{n=1}^{l_s}(\vec{N}\cdot\vec{L_n})I_n + \\ & K_s(\lambda)\sum_{n=1}^{l_s}(\vec{N}\cdot\vec{H_n})^{N_s}I_n. \end{aligned} \tag{2.3}$$

Blinn's modification uses the vector bisector $\vec{H}$, between $\vec{V}$ and $\vec{L}$, instead of the reflection vector $\vec{R}$. The rationale is that $\vec{H}$ represents the surface normal producing mirror reflection from light to the eye. Phong's function maintains the same profile relative to the reflected direction regardless of the direction of the angle of incidence. Blinn's function maintains the same profile relative to the reflected direction, but it gets narrower as the angle of incidence approaches



$\vec{V}$: The View Vector
$\vec{L}$: The Light Vector
$\vec{N}$: The Surface Normal
$\vec{R}$: The Reflection Vector
$\vec{T}$: The Transmission Vector
$\vec{H}$: The Bisector of $\vec{L}$ and $\vec{V}$ Vectors

Figure 2.1: *Geometry of the Phong's model.*

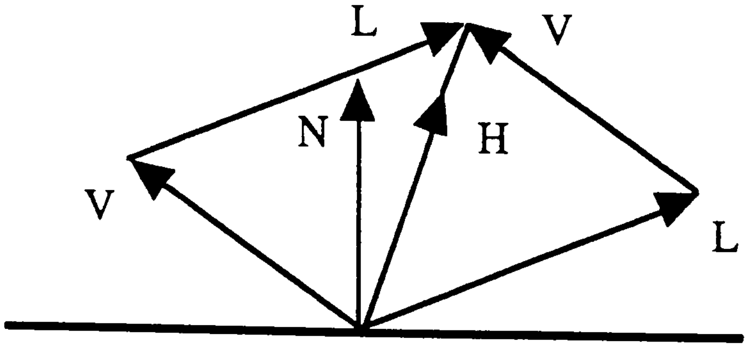
grazing. These functions are identical when the incident light is coincident with the normal, with the exception that a higher specular exponent, N_s, is required for Blinn's function to produce the same result. The modified version of the model accounts for the *color* (wavelength) of both the light source and the material and for multiple light sources placed anywhere in the environment. The selection of modeling parameters for material and light sources is detailed in Section 2.3.

2.2.3 Computation of $\vec{H}$ Vector

The $\vec{H}$ vector is a bisector between two vectors, an incident light vector, $\vec{L}$, and an outgoing vector, $\vec{V}$. The $\vec{H}$ vector represents the orientation of the surface normal that is required for mirror reflection between the $\vec{L}$ and $\vec{V}$ vectors. Since $\vec{L}$ and $\vec{V}$ are equal length vectors (both are unit vectors), their vector sum is a vector that bisects the angle between $\vec{L}$ and $\vec{V}$. The $\vec{H}$ vector is found by normalizing the vector sum:

$$\vec{H} = \frac{\vec{L}+\vec{V}}{\left|\vec{L}+\vec{V}\right|}. \tag{2.4}$$

See Figure 2.2 for vector diagram. The process of selecting material and lighting parameters to attain a realistic appearance is described in Section 2.3.



$\vec{L}$: The Light Vector
$\vec{V}$: The View Vector
$\vec{N}$: The Surface Normal
$\vec{H}$: The Bisector of $\vec{L}$ and $\vec{V}$ Vectors

Figure 2.2: *Vector diagram for computation of the $\vec{H}$ vector.*

2.3 Parameters Selection

This section examines the process of selecting material and lighting parameters to achieve a realistic appearance. There are two kinds of parameters involved in the Phong's improved model:

- material parameters, and
- light parameters.

Material parameters control the surface type (e.g., metal, plastic) and finish of the material. Light parameters model light sources such as ambient illumination and point light sources. Surface type can be broadly subdivided into that of dielectrics, conductors (metals), and composite materials.

2.3.1 Dielectric Materials

The electrons in the dielectric materials are not free to resonate or to be excited by the light wave when the light wave strikes the surface. The result is that the light wave passes through the media interface with little change. The reflectivity of dielectric surfaces is generally low and is uniform across the visible spectrum (see Figure 2.3). Dielectrics have the following characteristics:

- Dielectrics are not very reflective.
- Dielectrics are transparent.
- The spectral curves are identical for ambient, diffuse, and specular color.

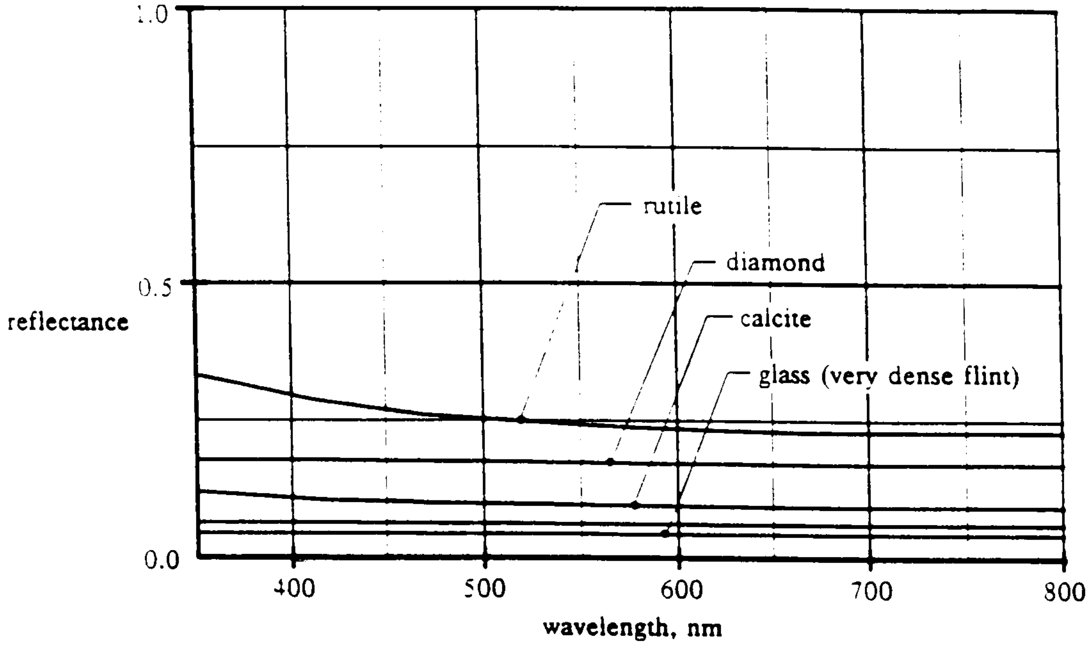


Figure 2.3: *Typical spectral curves for dielectric materials.*

2.3.2 Conductive Materials

The electrons in the conductive materials are free to move. Metals like aluminum, steel or nickle have a large number of free electrons that result in nearly equal reflection of all wavelengths. This results in the reflective grayish colors of these metals. The reflective characteristics of other metals are affected by electrons that are not free to move. This lack of freedom results in the selective absorption of light at frequencies where oscillation of the electrons is suppressed. This results in the color of metals such as gold, copper, or bronze. Briefly, the metals have the following characteristics:

- They are reflective.
- The spectral curves for ambient, diffuse, and specular color are the same.

Typical spectral curves for metals are shown in Figure 2.4.

2.3.3 Composite Materials

Briefly, the behavior of the composite materials is summarized by these observations:

- Their specular properties are those of dielectrics.
- Their diffuse properties are those of metals.
- The spectral curves for ambient, diffuse, and specular color are representative of the different materials in the composite.

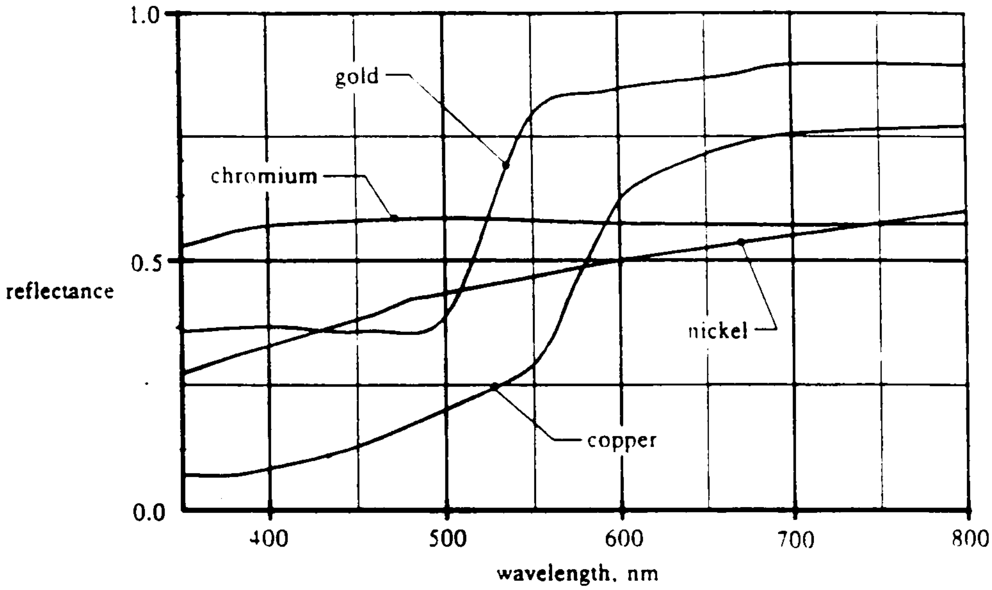


Figure 2.4: *Typical spectral curves for metals [34].*

2.3.4 Selection of K_a, K_d, K_s

In considering the values that should be used for material properties, it is helpful to cast the reflectance model so that terms such as K_a, K_d, and K_s are the product of a color and a multiplier. Phong's improved model, after extending it for multiple light sources and casting in terms of multipliers, can be expressed as:

$$\begin{aligned} I(\lambda) &= C_{K_a}K_a(\lambda)C_{I_a}I_a(\lambda) + C_{K_d}K_d(\lambda)\sum_{n=1}^{l_s}(\vec{N}\cdot\vec{L_n})C_{I_n}I_n(\lambda) \\ &\quad + C_{K_s}K_s(\lambda)\sum_{n=1}^{l_s}(\vec{N}\cdot\vec{H_n})^{N_s}C_{I_n}I_n(\lambda). \end{aligned} \tag{2.5}$$

Note that both the light sources and the materials are cast in terms of a spectral component, $K(\lambda)$, and a multiplier, C. Ambient illumination multipliers that work well are in the range:

$$0.0 \leq C_{I_a} \leq 0.3.$$

Low values simulate space or night conditions. Mid values simulate daylight on clear days when there is ambient light from a blue sky, or simulate interiors that have reflective finishes. High values simulate overcast daylight.

While using empirical models, the spectral curves are used directly for K_a, K_d, and K_s when the reflectance model is cast in terms of the multipliers and the spectral information.

2.3.5 Selection of C_{K_a}, C_{K_d}, C_{K_s}

In empirical models the rules of thumb for the selection of C_{K_a}, C_{K_d} and C_{K_s} are

- the diffuse and ambient multiplier are same, $C_{K_a} = C_{K_d}$
- the sum of the diffuse and specular multipliers should be roughly aqual to one i.e., $C_{K_d} + C_{K_s} \approx 1$.

The first rule of thumb comes from the reciprocity relationship. The ambient multiplier relates the intensity reflected in a given direction to the intensity incident from the all directions. The diffuse multiplier relates the intensity reflected in all directions to the energy incident from a given direction. The second rule helps to assure that the computed intensity is not outside the displayable color gamut. The next section describes the implementation of Phong's model.

2.4 Implementation of Phong's Improved Model

Phong's improved model is given in Equation (2.5). The extended form of Phong's model is implemented and utilized to simulate the reflectance response of the illuminated surfaces.

The geometry of the model is provided by describing the location of the surface normal, the location of the viewer, and the location of the light sources.

The view vector is assumed to be directed from the surface to the observer. Each light source is loaded into the model. The first light source is always assumed to be the ambient illumination. The remaining lights are considered to be point light sources.

Material information is provided by selecting the appropriate values for K_a, K_d, and K_s. Each material coefficient is described by a set of spectral samples and a scale factor. The spectral samples are obtained from the spectral curves of the materials (see Figures 2.3, 2.4). The scale factor determines the contribution of the illumination component. Section 2.3.5 describes the selection of scale factors. Typically, the ambient and diffuse spectral curves and scale factors are the same. If the material is ***homogeneous***, then the spectral curve is identical to the diffuse curve. If the material is a composite, the spectral curve is different. The relative magnitude of the diffuse and specular scale factor in combination with the surface roughness controls the surface character.

Light sources are described by the location (center) of the light, the sampled spectral curve, and an intensity scale. The first light is always considered to be an ambient illumination and its center is ignored. The spectral data for modeling the light sources can be obtained from [21,24].

2.5 Simulation of Metal Images Using Phong's Model

This section describes the application of Phong's improved reflectance model to simulate the illumination response of metals. The main consideration is the homogeneity of the material. Because we are concerned only with planar surfaces, surface normal is always orthogonal to the surface plane. This eliminates the need for computing the surface normal at each point on the surface. However, the model is not restricted only to planar surfaces and can be applied to any surface.

The spectral curve for film-coated aluminum is given in Figure 2.5. The parameters used in a simulation that models a film-coated aluminum sheet are given in Table 2.1. In this simulation, four light sources are used. The first light, as described in Section 2.4, is always an ambient light source while the remaining three lights are considered point light sources. The point light sources are spaced 120^0 apart. The purpose of this simulation is to investigate the effect of varying parameters such as N_s, C_{Ka}, C_{Kd}, and C_{Ks}, on the specular response of the surface illuminated. The ambient illumination multiplier, C_{Ia}, is chosen that simulates interiors with reflective finishes. Figures 2.6, 2.7, and 2.8 show the images resulting from the simulation. It is obvious from these images that the highlight or specular component is dependent upon the specular exponent N_s. Higher values result in narrower highlights, and smaller values of N_s produce broader highlights (see Figure 2.8). The intensities of the highlights are a function of C_{Ks} and K_s. For a given material, increasing the C_{Ks} increases the intensities of the highlights. The ambient illumination behavior is controlled by C_{Ka}. The values of C_{Ka} and C_{Kd} are selected as described in Section 2.3.5. Figure 2.5 shows the spectral curve of the evaporated film coated aluminum. It can be seen from the spectral curve that reflectivity of the aluminum is uniform

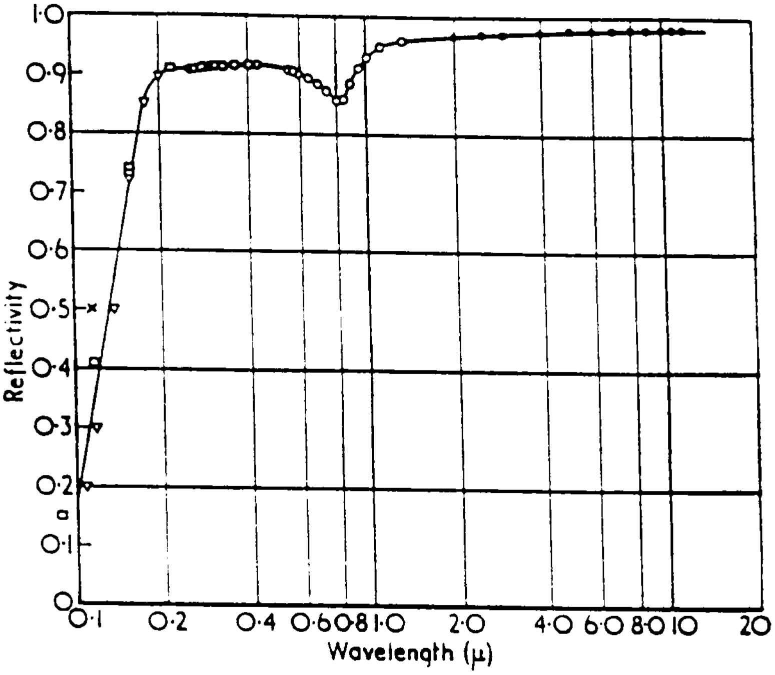


Figure 2.5: *Spectral curve for film-coated aluminum.*

Table 2.1: *Parameters used by the Phong's improved model to generate synthetic images for a film-coated aluminum planar sheet. The effects of these parameters are depicted by the resulting images.*

Wavelength	*Spectral*			*Spectral*	*Ambient Illumin.*	*Scale Factors*			*Resulting Image*
nm	*Parameters*			*Exponent*	*Factor*				
λ	K_a	K_d	K_s	N_s	C_{Ia}	C_{Ka}	C_{Kd}	C_{Ks}	
600	0.90	0.90	0.90	350.0	0.20	0.35	0.35	0.90	Figure 2.6
550	0.90	0.90	0.90	350.0	0.20	0.35	0.35	0.90	
450	0.92	0.92	0.92	350.0	0.20	0.35	0.35	0.90	
600	0.90	0.90	0.90	45.0	0.20	0.38	0.38	0.65	Figure 2.7
550	0.90	0.90	0.90	45.0	0.20	0.38	0.38	0.65	
450	0.92	0.92	0.92	45.0	0.20	0.38	0.38	0.65	
600	0.90	0.90	0.90	5.0	0.20	0.50	0.50	0.25	Figure 2.8
550	0.90	0.90	0.90	5.0	0.20	0.50	0.50	0.25	
450	0.92	0.92	0.92	5.0	0.20	0.50	0.50	0.25	

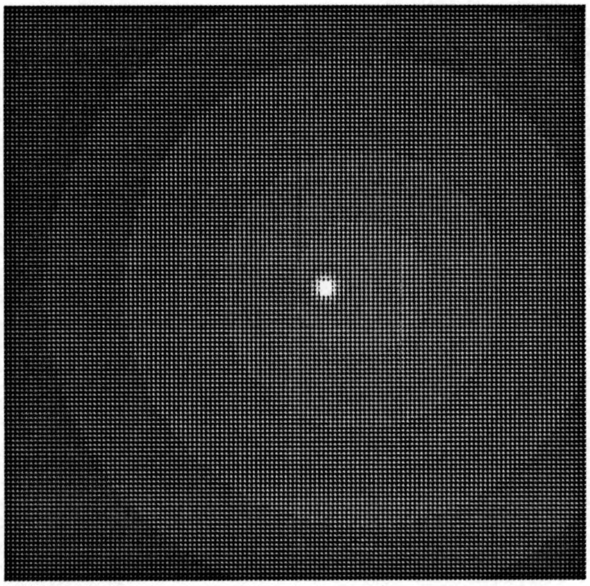


Figure 2.6: *Synthetic image for a film-coated aluminum planar sheet using Phong's improved model. Highlight is shown by the bright spot in the image. Parameters used:* $N_s = 350.0$, $C_{Ka} = C_{Kd} = 0.35$, *and* $C_{Ks} = 0.90$.

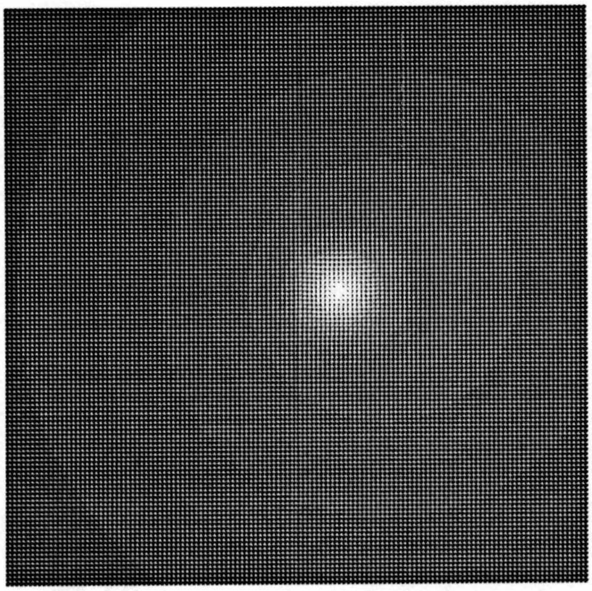


Figure 2.7: *Synthetic image for a film-coated aluminum planar sheet using Phong's improved model. Parameters used:* $N_s = 45.0$, $C_{Ka} = C_{Kd} = 0.38$, *and* $C_{Ks} = 0.65$.

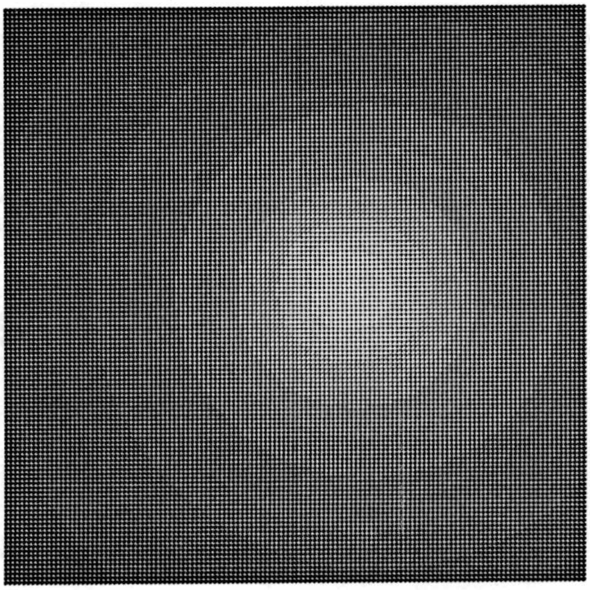


Figure 2.8: *Synthetic image for a film-coated aluminum planar sheet using Phong's improved model. The highlight is broader as compared to image shown in Figure 2.9. Parameters used:* $N_s = 5.0$, $C_{Ka} = C_{Kd} = 0.50$, *and* $C_{Ks} = 0.25$.

in the visible spectrum range that is why the image of aluminum has a grayish appearance. Because the Phong's improved model is a function of the wavelength, the model is evaluated three times at each point on the surface to compute three pixel intensity values, one for each wavelength corresponding to $\lambda = 600, 550$ and 450 nm. This results in three image planes, namely red, green, and blue, at the corresponding wavelengths respectively. These image planes are combined to generate a color (RGB) image for the illuminated surface. The role of color in the defect detection will be discussed in the next section.

The results of the simulation are in accordance with theoretical expectations. However, as the model is empirical in nature, to achieve an appearance that is realistic is a matter of adjusting and testing various values for the parameters. The results of the simulation show that Phong's extended model can be applied to simulate a reflectance response for any metal by choosing proper modeling parameters for material and light sources.

The location and number of light sources are provided while simulating the illumination response of a given surface. Thus, the arrangement of optimum light sources can be determined empirically. The term, optimum, refers to light sources placement that will produce the least specular reflectance response. As another example, Table 2.2 gives the parameters used for simulation of a planar copper sheet. In this simulation five light sources are used and placed asymmetically in the surroundings. The images resulting from this simulation are depicted in Figures 2.9 and 2.10. The intensity distributions produced by the application of

Table 2.2: *Effect of varying the modeling parameters using improved Phong's model on the simulated images of a copper planar sheet.*

Wavelength nm	*Spectral Parameters*			*Spectral Exponent*	*Ambient Illumin. Factor*	*Scale Factors*			*Resulting Image*
λ	K_a	K_d	K_s	N_s	C_{Ia}	C_{Ka}	C_{Kd}	C_{Ks}	
600	0.63	0.63	0.63	80.0	0.20	0.35	0.35	0.80	Figure 2.9
550	0.32	0.32	0.32	80.0	0.20	0.35	0.35	0.80	
450	0.14	0.14	0.14	80.0	0.20	0.35	0.35	0.80	
600	0.63	0.63	0.63	150.0	0.20	0.36	0.36	0.75	Figure 2.10
550	0.32	0.32	0.32	150.0	0.20	0.36	0.36	0.75	
450	0.14	0.14	0.14	150.0	0.20	0.36	0.36	0.75	

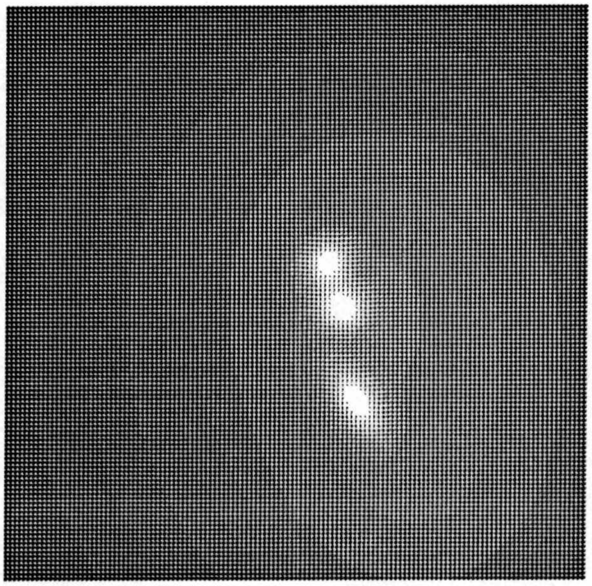


Figure 2.9: *Synthetic image for a copper planar sheet using Phong's improved model for asymmetrical light sources positionings. Parameters used:* $N_s = 80.0$, $C_{Ka} = C_{Kd} = 0.35$, *and* $C_{Ks} = 0.80$.

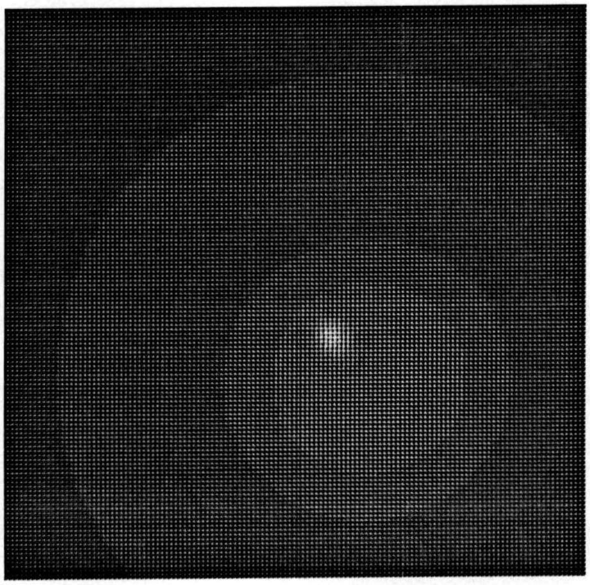


Figure 2.10: *Synthetic image for a copper planar sheet using Phong's model for asymmetrical light sources positionings. Parameters used:* $N_s = 150.0$, $C_{Ka} = C_{Kd} = 0.36$, *and* $C_{Ks} = 0.75$.

Phong's improved model are non-linear. This should be expected because the specular function used in Phong's model to simulate the specular reflectance of the illuminated surface is a non-linear function. The impact of this non-linearity on the design of the vision system is discussed in Chapter 3.

2.6 Role of Color in Defect Detection

The process of metal surface reflectance simulation is described in the previous section. An image is generated synthetically by evaluating Phong's improved reflectance model at three different wavelengths ($600, 550$, and 450 nm) corresponding to red, green, and blue. The spectral response of a surface depends on the surface character, finish, and roughness. The spectral response of a surface with defects is different from a defect free surface. As the surface character determines the spectral response, a defect on a smooth surface can exhibit itself more predominantly under certain wavelengths. A similar spectral response of a simulated metal surface is shown in Figure 2.11. A defect is modeled using a flaw generator and then superimposed on the simulated image of the metal surface [5]. It is obvious from Figure 2.11 that the defect is more visible in the blue image plane ($\lambda = 450$ nm) than in the red or green image planes. This implies that if we can determine that a defect is more dominant for certain wavelengths, then only the corresponding image plane need be processed for defect detection.

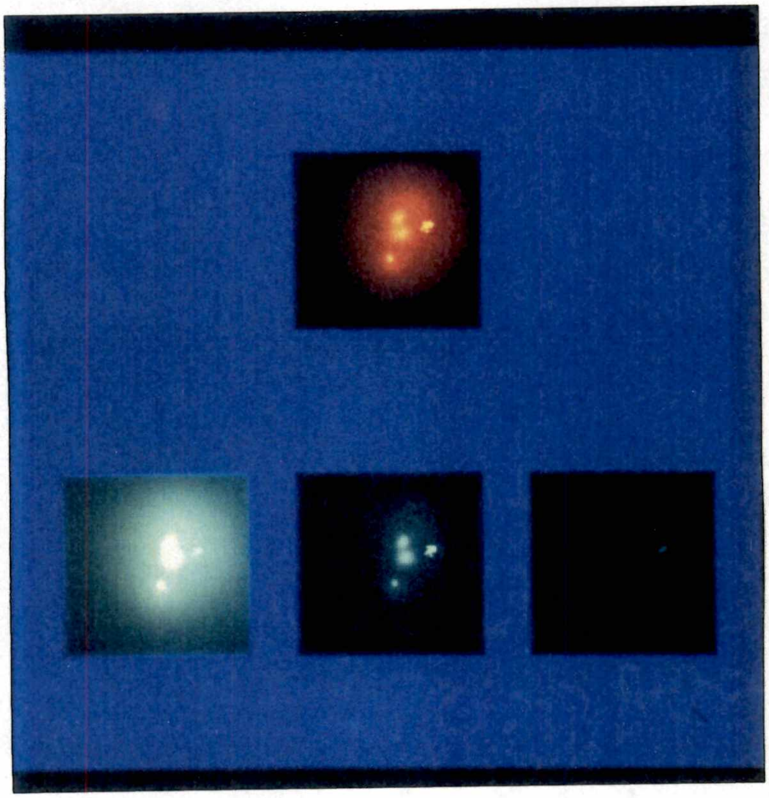


Figure 2.11: *(a): Color image for a copper planar sheet; (b): Red image plane; (c): Green image plane; (d): Blue image plane.*

2.7 Summary

The purpose of this chapter is to investigate the reflectance response of specular surfaces and to use this knowledge to develop a vision system for defect detection. Phong's extended model (Equation (2.5)) is implemented and used to simulate the reflectance response of metals. By the appropriate selection of the modeling parameters for material and light sources, a realistic appearance can be obtained. The results of this chapter provide an insight into the reflectance behavior of polished metal surfaces. The intensity distributions resulting from the application of Phong's improved model to an illuminated surface are non-linear. Chapter 3 describes the design and development of the vision system for detection of defects in the digitized images of metal surfaces.

CHAPTER 3

Fast Defect Detection System

3.1 Introduction

This chapter describes a method for detecting defects such as streaks and speckles in the digitized images of metal surfaces of known intensity distributions (e.g., aluminum). Defect detection involves two steps: defect extraction and defect classification. The process of defect extraction is described in this chapter. Defect classification is accomplished using a *back-propagation* multilayer perceptron classifier. The classifier is discussed in more detail in Chapter 4.

It was illustrated in Chapter 2 that the images resulting from the illumination of specular surfaces contain specularities (highlights) and are characterized by gradual changes in intensities. Furthermore, it was shown that the intensity distributions are non-linear and are a function of material characteristics such as K_a, K_d, K_s, and surface finish, roughness, etc. Consequently, traditional image segmentation methods (e.g., thresholding) cannot handle various reflectances present in an image and typically cannot successfully segment an image of a metal surface into regions that correspond to different reflectances. This chapter describes an approach to detect regions with abrupt intensity changes in the digitized images of metal surfaces. These regions of abrupt changes correspond to possible defects in an image. Before the detection system and various processing steps required for defect detection are detailed, it is important to elaborate what we mean by the term *defect(s)*. The term defect(s) refers to one or all of the following:

- Streak - an elongated elliptical region with a different intensity from that of the background, and
- Speckles - a cluster of small elliptical regions.

The remainder of the chapter is dedicated to the design, development, and application of the detection system for defect detection.

3.2 Defect Detection System

The block diagram showing the functional elements of the detection and classification system is depicted in Figure 3.1. This system can be divided into two stages: defect extraction and defect classification. This chapter deals with the first stage; the second stage is dealt with in Chapter 4.

3.3 Defect Extraction

The task of the defect detection unit is to extract features of a possible defect and generate a pattern vector containing the defect features that adequately distinguish the class of each defect. The features may be specific measurements

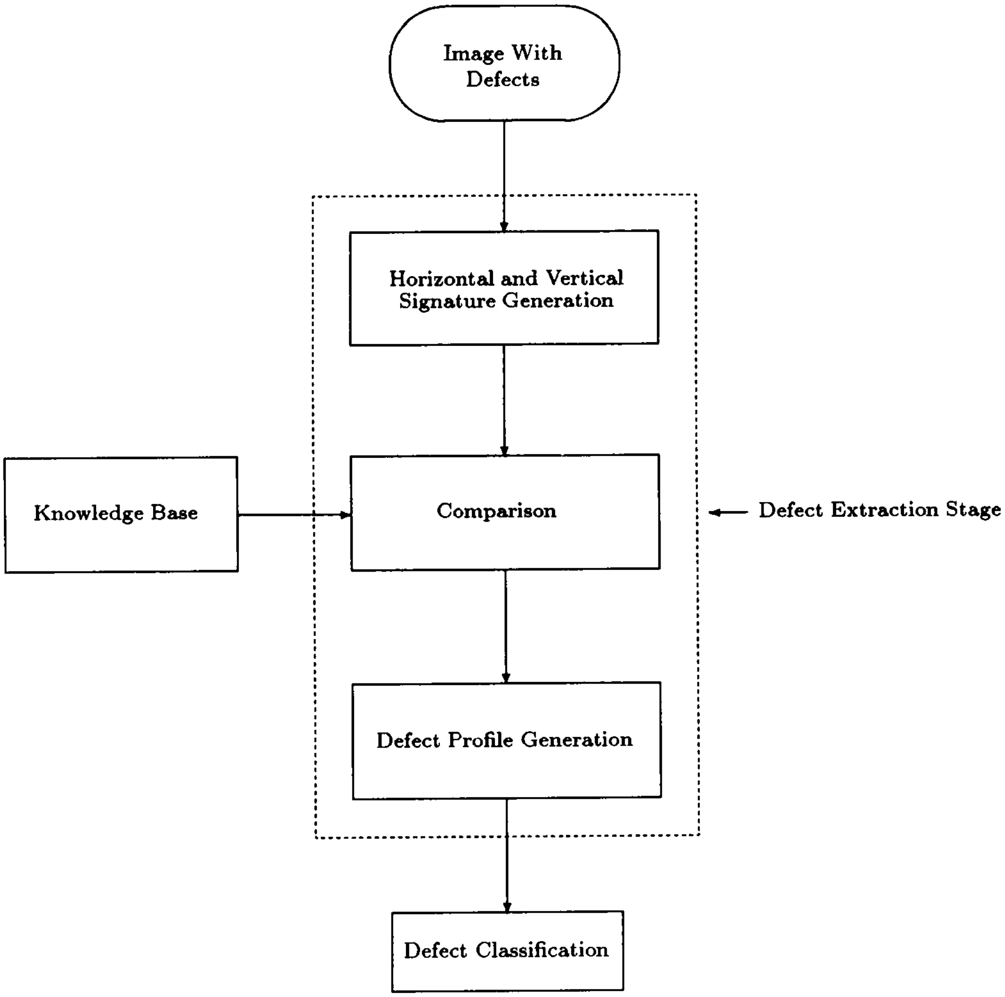


Figure 3.1: *Functional diagram of the vision system for defect detection and classification.*

of defect characteristics, or they may be profiles of the defect as contained in the 1–D signatures of the digitized image.

3.3.1 Signature Generation

Possible defects are detected by partitioning the image into non-overlapping regions called windows. Then each window is processed by

- generating horizontal and vertical signatures, and
- analyzing these signatures for sharp changes.

Each of the windows is processed individually, and for each window two kinds of profiles, namely horizontal and vertical profiles, are generated (see Figure 3.2). The terms signature and profile are used interchangeably. Horizontal signatures are obtained by summing the pixel intensities of the rows, and vertical signatures are obtained by adding the pixel intensities of the columns of the image within each window. Thus, for a window of size $M \times N$ pixels the two signatures, $S(i)$ and $S(j)$, are given by the following equations:

$$S(i) = \sum_{j=1}^{M} F(i,j) \; i = 1, 2, ..., N, \tag{3.1}$$

and

$$S(j) = \sum_{i=1}^{N} F(i,j) \; j = 1, 2, ..., M. \tag{3.2}$$

Examples of the signatures generated by the application of Equations (3.1) and (3.2) to an image are shown in Figure 3.3. The horizontal and vertical signatures compress two-dimensional information into one-dimensional information. This

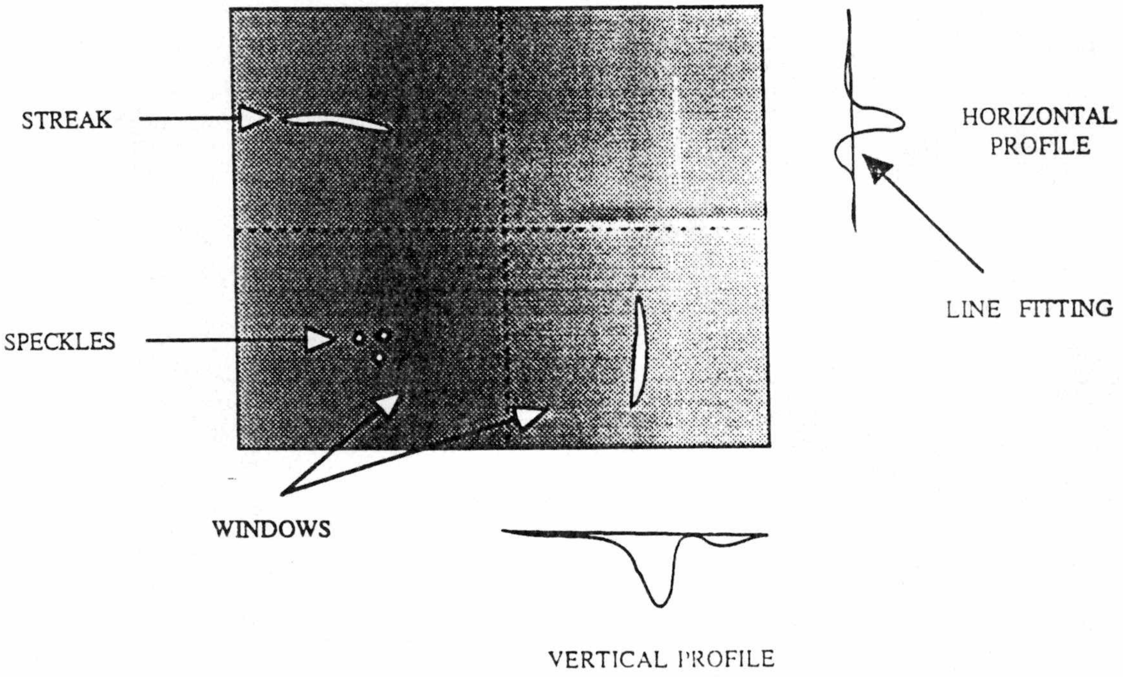


Figure 3.2: *Diagram showing defects, partitioning of an image into windows, and generation of horizontal and vertical signatures.*

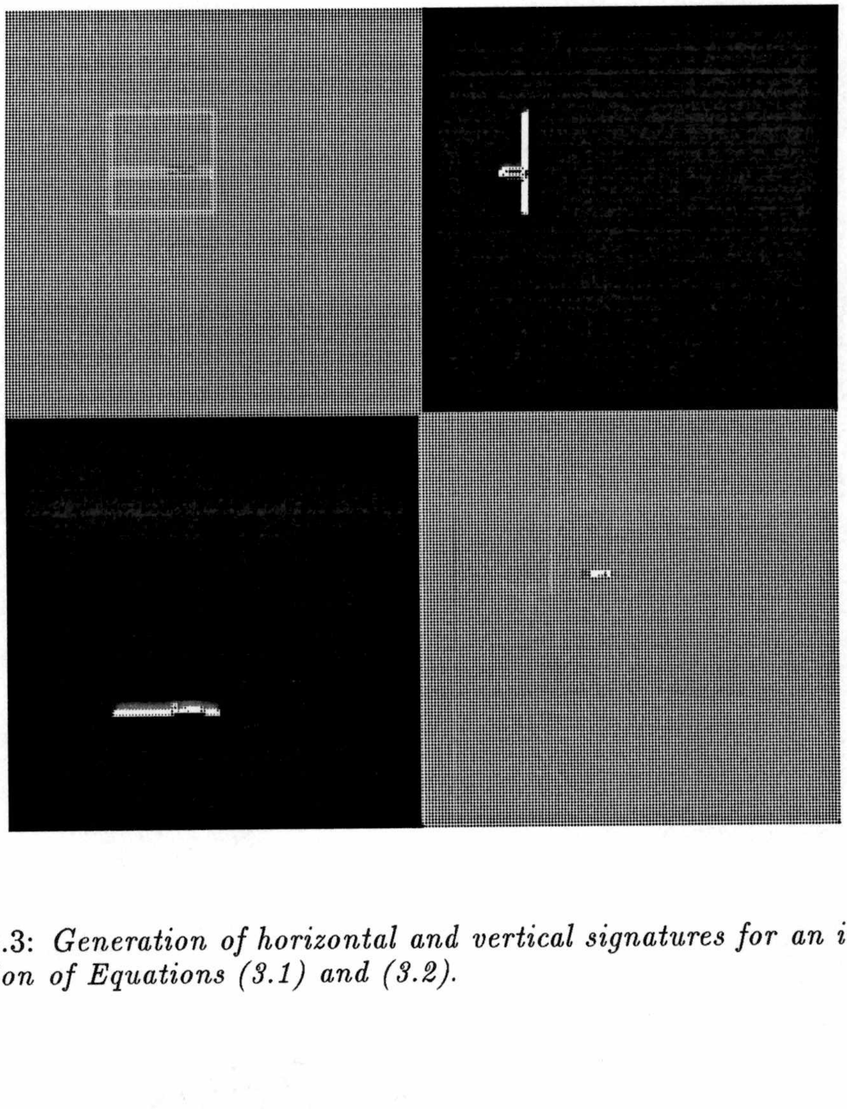


Figure 3.3: *Generation of horizontal and vertical signatures for an image by the application of Equations (3.1) and (3.2).*

means that a 2–D image is processed using its two 1–D signatures, which offers a significant computational advantage over 2–D processing of the image. The analysis of the horizontal and vertical signatures of an image with possible defects involves a comparison of these signatures to the signatures of an image that does not contain defects. The objective of this comparison is to determine the discrepancies between the signatures of the two images. For example, the horizontal and vertical signatures of a homogeneous surface with uniform intensity are fairly smooth if a defect is not included in the summation, in which case profiles will show spike or abrupt change at the location of each defect. The next section describes the steps required to determine these discrepancies.

3.3.2 Defect Profiles

Discrepancies between the signatures of an image with defects and a defect free image are interpreted as possible defects. To determine these discrepancies, first, all elements of horizontal and vertical profiles are fitted to a line using *least-square* fitting. Second, elements of the horizontal and vertical profiles that are below a certain threshold (distance) based on the average intensity value in the window from the line, are set to zero. The resulting profiles of each window show only regions of abrupt changes in the window. These profiles which show only the regions of abrupt changes are termed *defect profiles.* A typical example of defect profiles for streaks and speckles is shown in Figure 3.4.

Only information reflecting abrupt changes is passed on to the classifier. Classification is accomplished by using a multilayer perceptron classifier that utilizes

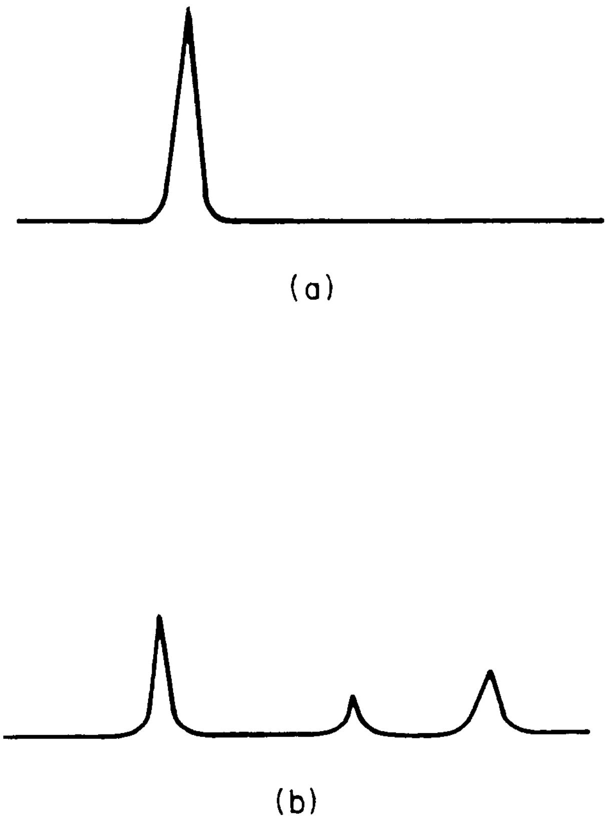


Figure 3.4: *Typical vertical defect profiles for (a) streak; (b) speckles.*

the back-propagation training algorithm. The classifier is described in detail in Chapter 4. By the generation of horizontal and vertical signatures, two-dimensional information is lost by compressing a two-dimensional signal to a one-dimensional signal. The impact of this on the classifier is described in [5]. The main steps involved in defect detection are described below.

Step 1. Input the image with possible defects.

Step 2. Partition the image into windows and process each window individually.

Step 3. Generate horizontal and vertical profiles for each window by adding the pixel intensities of the columns and rows using the Equations (3.1) and (3.2).

Step 4. Fit all elements of the horizontal and vertical profiles to line using least-square fitting.

Step 5. Set all elements of the profile to zero that are below a threshold distance. Generate pattern vectors containing the defect features.

Step 6. Input the defect profile obtained as a result of Step 5 to the classifier.

3.3.3 Effect of Window Size on Detection of Defects

As described earlier, the defect detection process involves: subdividing the image into windows, generating horizontal and vertical signatures, and analyzing these signatures for abrupt changes. This section details how the abrupt changes in the signatures produced by the defects are affected by changing window size.

To determine the effect of window size on defect detection we need to study more carefully the mechanism of signature generation. Figure 3.5 shows only one row of an image having a uniform intensity of 160. The contribution of the defect in the k^{th} row within the window is exhibited by pixels with intensity values of 180. To generate the k^{th} element of the horizontal signature for a window with N columns, all pixels for the k^{th} row inside this window are first added and then divided by the number of columns, N, i.e.,

$$S(k) = \frac{\sum_{j=1}^{N} F(k,j)}{N}.$$

Let us apply this expression to calculate the k^{th} element of the signature for a window with $N = 8$:

$$S(k) = \frac{\overbrace{(160 + 160 + 160 + \overbrace{180 + 180 + 180}^{defect\ pixels} + 160 + 160)}^{image\ pixels}}{8} = 167.50.$$

Now if the window size is increased to $N = 12$, the value of the k^{th} element of the horizontal signature for this window will be

$$\begin{aligned} S(k) &= \frac{(160 + 160 + 160 + 180 + 180 + 180 + 160 + 160 + 160 + 160 + 160 + 160)}{12} \\ &= 165.0. \end{aligned}$$

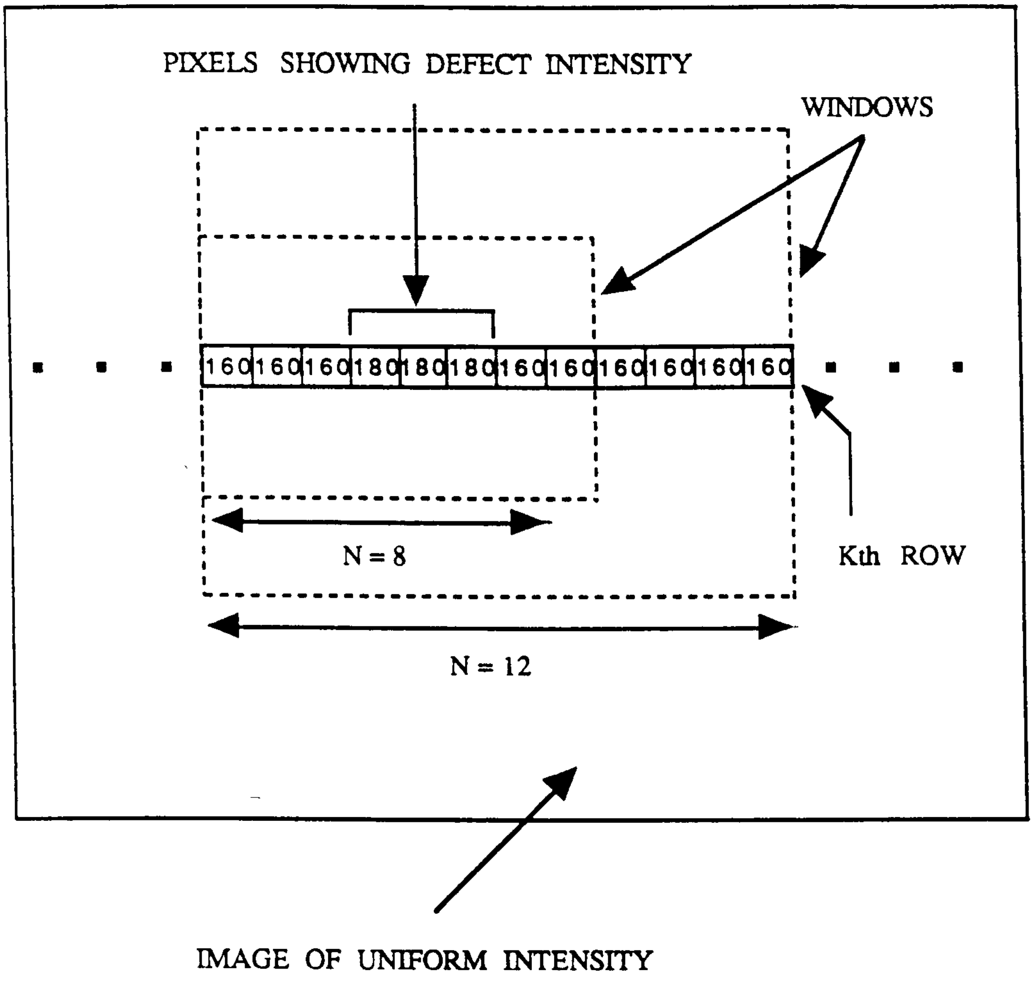


Figure 3.5: *Diagram showing the effect of window size on the k^{th} element of the horizontal profile for an image containing defects.*

From the illustrations given for the k^{th} row of the image, the following observations can be made:

a. Averaging the intensity for each row smoothes the abrupt change in the horizontal signatures.

b. A larger size defect will show itself as an abrupt change in the signatures for a larger or smaller window.

c. A small defect will not have a significant effect on the profile for a window much larger in size than the defect.

d. The larger the defect size (elongation), the greater the number of defect pixels contributing to a given signature element.

e. The larger the difference in intensities (contrast) of the background and the defects pixels, the greater the deviations in the signature elements from the mean value.

The same illustrations can be used for vertical signature generations by interchanging the rows and columns. These observations suggest that larger size windows can be utilized when detection of larger defects is the main concern. For smaller defect detection, windows of a smaller size must be used. This is due to the fact that the defect contribution to the signature elements increases as the window size decreases.

Table 3.1 shows the results of an experiment performed to determine the effect of contrast on defect detection. All images in this experiment are of uniform

Table 3.1: *Results of experiment performed to determine the effect of contrast on detection of a constant size defect for a given window size.*

Image Size	*Image Intensity*	*Defect Size*	*Defect Intensity*	*Defect Location*	*Absolute Inten. Difference*	*Window Size*	*Defect Detected ?*
256 x 256	170	5 x 3	167	(20, 20)	3	32 x 32	*No*
256 x 256	170	5 x 3	165	(20, 20)	5	32 x 32	*No*
256 x 256	170	5 x 3	160	(20, 20)	10	32 x 32	*Yes*
256 x 256	170	5 x 3	150	(20, 20)	20	32 x 32	*Yes*
256 x 256	170	8 x 3	167	(20, 20)	3	32 x 32	*No*
256 x 256	170	8 x 3	165	(20, 20)	5	32 x 32	*No*
256 x 256	170	8 x 3	160	(20, 20)	10	32 x 32	*Yes*
256 x 256	170	8 x 3	150	(20, 20)	20	32 x 32	*Yes*

intensity. Defect and window size are constant. It follows from the results in Table 3.1 that for a given window size the defect is not detected for small contrast values. However, increasing the contrast causes the defect to be detected. The results of another experiment are given in Table 3.2. The objectives of this experiment are to determine:

- the minimum defect size the system can detect, and
- the system behavior for a defect under multiple window size.

In this case, defect intensity is fixed but the size is changed and the experiment is repeated for multiple window sizes. It can be seen from these results that the minimum detectable defect size is 2×2 for the given contrast. The special case of defects not lying in a single window is given in Table 3.3. Results show that a defect bisecting the window boundaries is detected in various window sizes.

From the results of these experiments it is obvious that there are various factors involved in the detection of defects. Defect size, contrast, window size, and the threshold all affect the detection process. First, a larger defect (elongated) is easier to detect than a smaller defect. A smaller defect requires smaller windows. Second, the larger the difference in intensities (contrast), the greater the likelihood of a defect to be detected. The impact of the threshold choice on the defect detection is described in the Section 3.5. These test results are based on experimentation for images with uniform intensity. Real images do not necessarily have uniform intensity distributions, which is the case here. The applications of the detection system to the real images are described in Section 3.5.

Table 3.2: *Results of an experimentation performed to determine the minimum size of defect that can be detected. Intensity difference (contrast) is constant and image is processed for multiple window sizes.*

Image Size	*Image Intensity*	*Defect Size*	*Defect Intensity*	*Window Size*	*Defect Detected ?*
256 x 256	170	2 x 2	160	32 x 32	*Yes*
256 x 256	170	2 x 2	160	64 x 64	*Yes*
256 x 256	170	2 x 2	160	128 x 128	*Yes*
256 x 256	170	3 x 2	160	32 x 32	*Yes*
256 x 256	170	3 x 2	160	64 x 64	*Yes*
256 x 256	170	3 x 2	160	128 x 128	*Yes*
256 x 256	170	4 x 2	160	32 x 32	*Yes*
256 x 256	170	4 x 2	160	64 x 64	*Yes*
256 x 256	170	4 x 2	160	128 x 128	*Yes*

Table 3.3: *Results showing the detection system behavior for defects that are not completely contained in one window.*

Image Size	*Image Intensity*	*Defect Size*	*Defect Intensity*	*Window Size*	*Defect on the Boundary ?*	*Defect Detected ?*
256 x 256	170	8 x 3	160	16 x 16	*Yes*	*Yes*
256 x 256	170	8 x 3	160	32 x 32	*No*	*Yes*
256 x 256	170	8 x 3	160	60 x 60	*Yes*	*Yes*
256 x 256	170	8 x 3	160	80 x 80	*No*	*Yes*
256 x 256	170	8 x 3	160	110 x 110	*Yes*	*Yes*
256 x 256	170	8 x 3	160	128 x 128	*No*	*Yes*

3.4 Fast Version of The Defect Detection Algorithm

The main steps involved in defect extraction and detection are described in Section 3.3. From the experimental results of the previous section it follows that to have consistent flaw detection and classification, an image should be processed using multiple window sizes. As given in Equations (3.1) and (3.2), profile generation involves the addition of row and column intensities for a given window size. These operations are performed for every window of the image containing possible defects. This means that the same operations are carried out redundantly from one window size to another. If the results obtained by the intensity summation of a given window size can be utilized while processing the image for a larger window size, this can considerably reduce the computations. The reduction in computations means less CPU time and a faster algorithm.

The main driving force behind the new version is to improve the computing time by reducing the number of operations. The number of operations is reduced by eliminating the redundant computations. For example, to elaborate on how the computations performed for a given window size are utilized by a window of larger size, let us consider a 128×128 image which is to be processed using 32×32 and 64×64 windows. The image is first partitioned into 32×32 windows and for each window horizontal and vertical profiles are generated by adding the rows and columns pixels. The summations computed for every 32×32 window are preserved for use by windows of progressively larger size. Now if the same image is to be processed using 64×64 windows, then instead of adding all the pixels of a 64×64 window, only the pixels not included in the previous 32×32 window are added. The processing of the 64×64 window is completed by adding the summation of the corresponding 32×32 window to the the summation obtained by adding the pixels, which lie outside the 32×32 but inside the 64×64 window. Similarly, 64×64 window summations can be saved if the same image is to be processed using larger windows. This means that at each processing stage the results from the previous stage are utilized, which reduces processing for each next step. Hence considerable speed-up can be obtained by avoiding the repetitious operations.

In the modified version, first an image to be processed is partitioned into windows of the smallest size. Then for each following window size the summations computed in the previous stage are used by adding only pixels lying outside the previous window but inside the present window. See for example Figure 3.6, where an image is subdivided into the smallest $n \times n$ window (e.g., 8×8). Then the image is processed for the smallest window size, and profiles are generated for each window. If the image is to be processed using the next larger window, then the results of the smallest window computations are utilized. Thus, results of the present window are utilized by the next larger window and so forth. As a consequence to reduction in operations, computational efficiency increases as the window size increases.

Let us consider an image of size $2^n \times 2^n$ and the smallest window of size $n \times n$. If q is the total number of $n \times n$ windows into which the image is subdivided, then the total number of operations (summations) required for horizontal or vertical

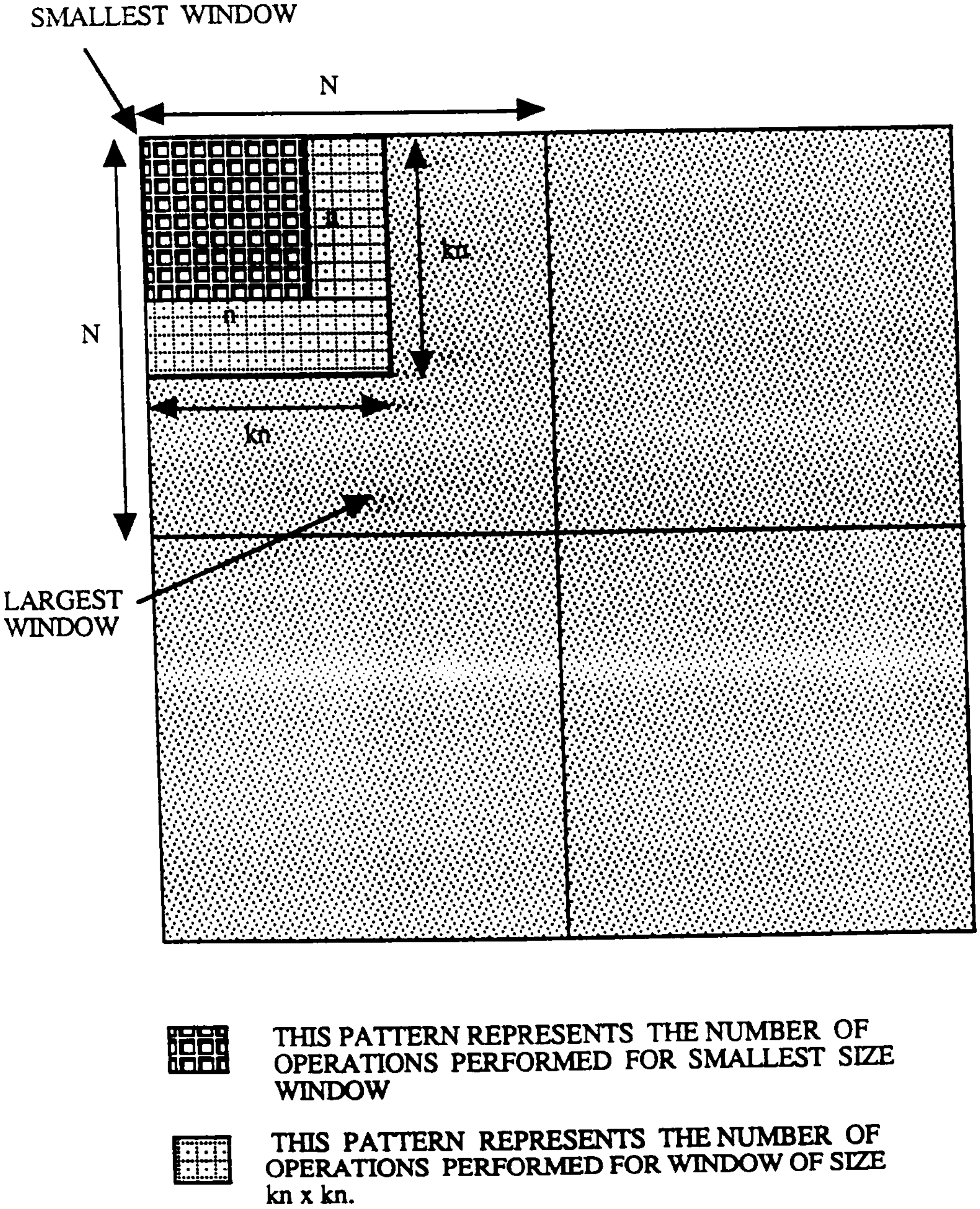

SMALLEST WINDOW
N
N
kn
n
n
kn
LARGEST WINDOW
THIS PATTERN REPRESENTS THE NUMBER OF OPERATIONS PERFORMED FOR SMALLEST SIZE WINDOW
THIS PATTERN REPRESENTS THE NUMBER OF OPERATIONS PERFORMED FOR WINDOW OF SIZE kn x kn.

Figure 3.6: *Diagram showing functioning of the fast defect detection algorithm.* $n \times n$,*is the smallest window size and* $N \times N$, *the largest window size.*

profiles is given by $q(n^2 - n)$. If the window size for the next stage is k times the smallest window size and there are m such windows in the image, then the total number of operations required for the next stage is $m(k^2n^2 - kn)$. By reusing the summations of the previous stage, the actual number of operations required is reduced to mkn. This computational efficiency is illustrated in Table 3.4, where the smallest window size of 8 × 8 and the largest of 128 × 128 pixels are used for an image of size 256 × 256. The main steps involved in the defect extraction and detection in the modified version of the algorithm are outlined in Section 3.4.1.

Table 3.4: *Reduction in number of operations by eliminating redundant computations for multiple window size processing for an image of* 256 × 256. *Size of the smallest window is* 8 × 8 *and the largest* 128 × 128.

Window Size	*Total Windows*	*Operations Required* O_r	*Operations Performed* O_p	*Reduction in Operations* $\frac{(O_r - O_p)}{O_r} \times 100$
8 × 8	1024	57344	–	–
16 × 16	256	61440	4096	93.33 %
32 × 32	64	63488	2048	96.77 %
64 × 64	16	64512	1024	98.41 %
128 × 128	4	65024	512	99.21 %

3.4.1 Fast Algorithm for Defect Extraction and Detection

The mechanism of the detection algorithm is described in Section 3.4. The main steps involved in the defect extraction and detection algorithm are described below. Flow chart showing the logic of the algorithm is presented on the next page.

Step 1. Input an image with possible defects.

Step 2. Partition the image into windows and process each window individually. Start with the smallest window.

Step 3. Generate horizontal and vertical profiles for each window by adding the pixel intensities of the columns and rows.

Step 4. Fit all elements of the horizontal and vertical profiles to line using least-square fitting.

Step 5. If the image is to be processed for another window size go to Step 6. Otherwise go to Step 7.

Step 6. Add intensity values of those pixels not contained in the previous window. Combine result of the present window with that of the previous window and generate the horizontal and vertical profiles. Go to step 4.

Step 7. Set all elements of the profile to zero that are below a threshold distance. Generate pattern vectors containing the defect features.

Step 8. Input the defect profile obtained as a result of Step 7 to the classifier.

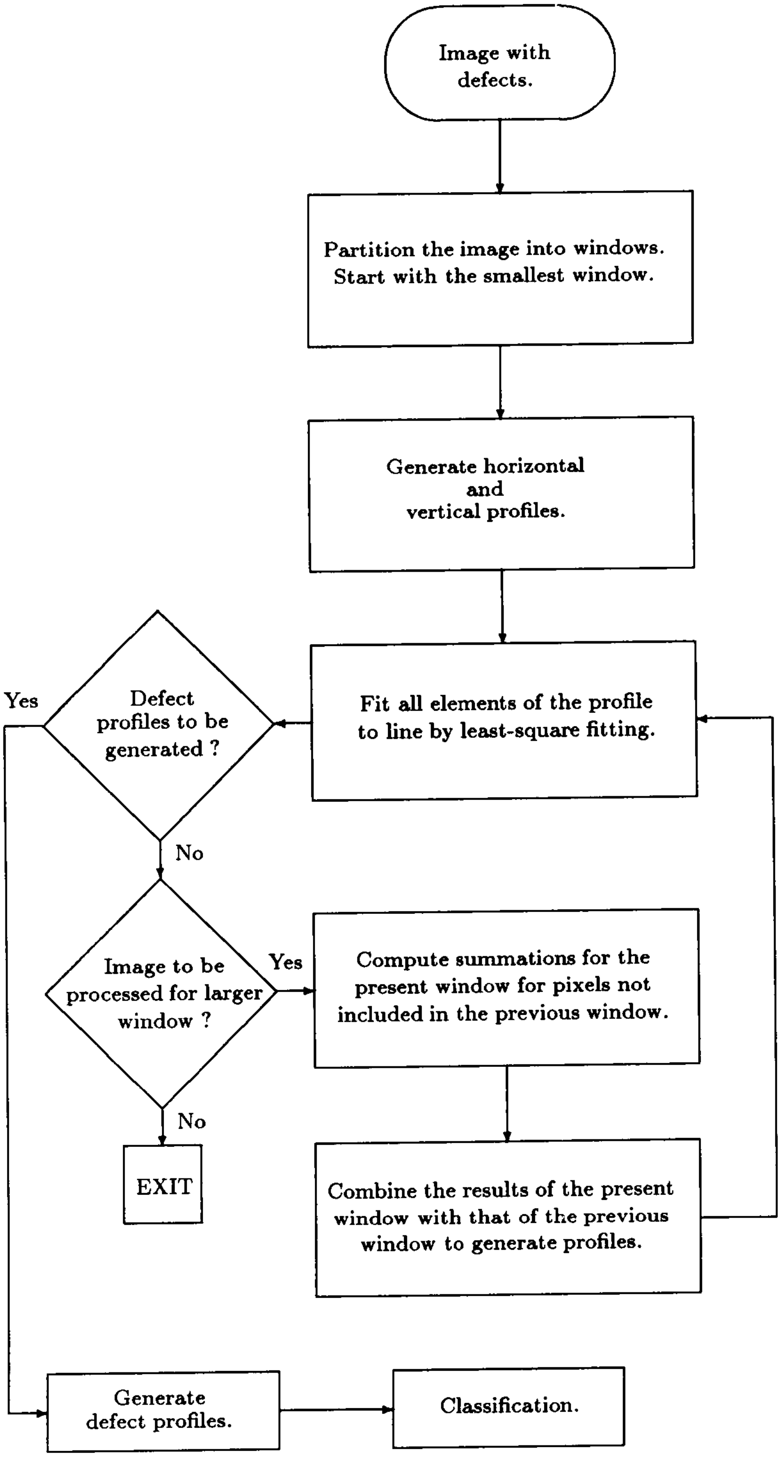


Image with defects.
Partition the image into windows. Start with the smallest window.
Generate horizontal and vertical profiles.
Fit all elements of the profile to line by least-square fitting.
Defect profiles to be generated ?
Yes
No
Image to be processed for larger window ?
Yes
No
Compute summations for the present window for pixels not included in the previous window.
Combine the results of the present window with that of the previous window to generate profiles.
EXIT
Generate defect profiles.
Classification.

Figure 3.7: *Flow chart showing the logic of the defect detection algorithm.*

3.5 Applications

This section describes applications of the detection system for defect extraction and detection in digitized images of the metals. The detection system is first applied to synthetic images with known intensity distributions; then real images of metal surfaces (i.e., aluminum) are considered. In the case of synthetic images, a flaw generator is used to generate deformed ellipses to model the defects [5]. The length of the major and minor axes of the ellipses can be varied to model streaks or speckles. A more realistic defect may appear when both shape and intensity of the ellipse are deformed. The intensity deformation is accomplished by adding random noise to the ellipse. The shape of the ellipse is deformed, along both x and y, by adding or eroding pixels at the ellipse border.

Images with linear intensity variations along the x and y axes are shown in Figures 3.8 and 3.9, respectively. After the addition of defects, these images are processed for defect detection. Figure 3.10 shows an example of defect detection using a 128×128 window for the image having intensity variations along the x axis. Figure 3.11 depicts the detection of defects for the image with intensity variations along the y axis. Detection of a defect is indicated by marking horizontal and vertical lines across the windows. The results of defect detection for synthetic images under multiple window sizes are tabulated in Tables 3.5 and 3.6.

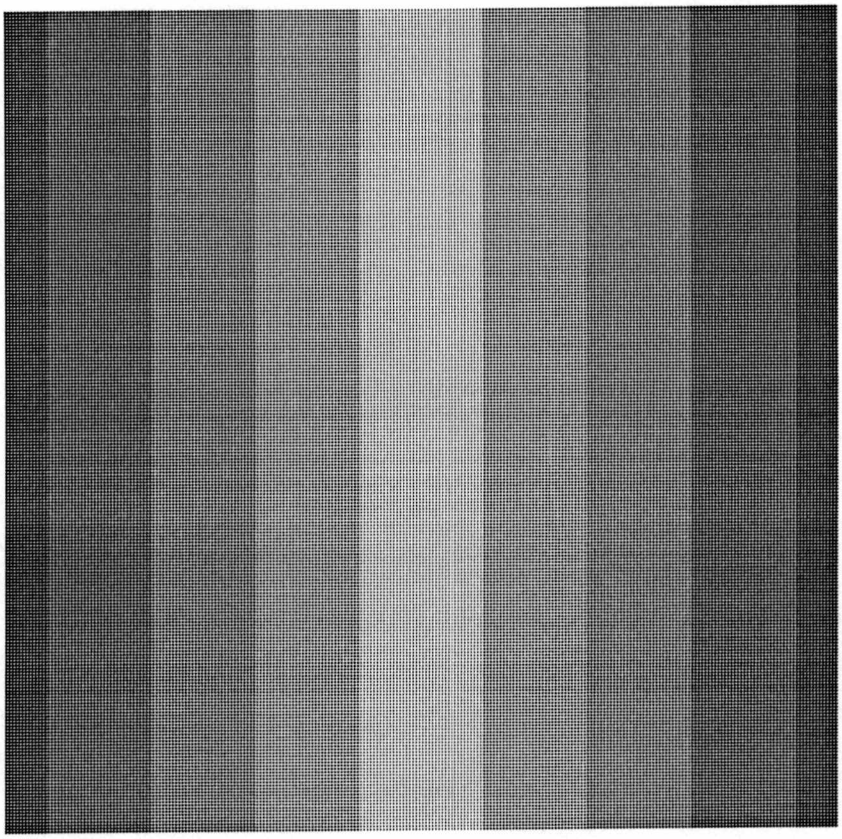


Figure 3.8: *A* 256 × 256 *synthetic image with linear intensity variations along the x axis.*

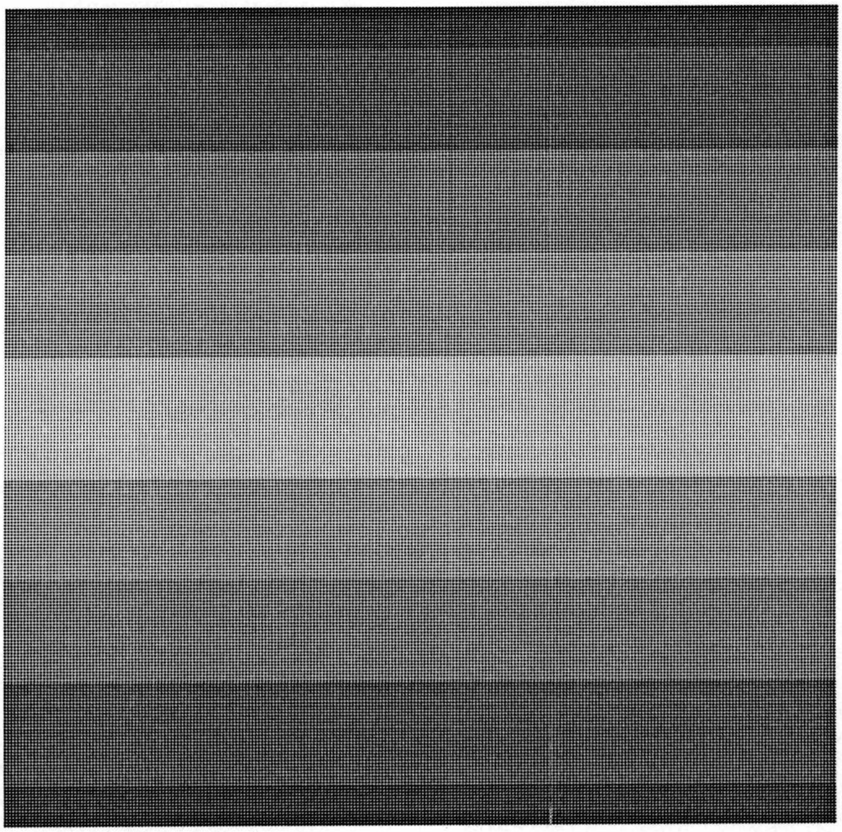


Figure 3.9: *A* 256 × 256 *synthetic image with linear intensity variations along the y axis.*

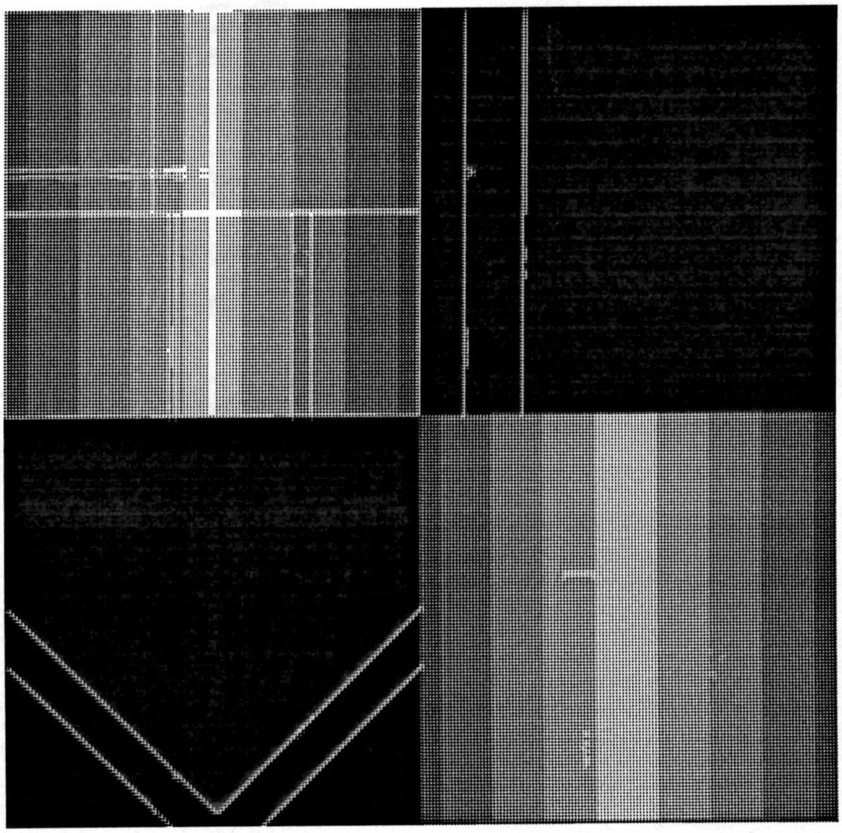


Figure 3.10: *Detection of defects for a* 256×256 *synthetic image with linear intensity variations along the x axis. Window size* 128×128 *pixels. Detection of a defect is indicated by marking horizontal and vertical lines across the windows.*

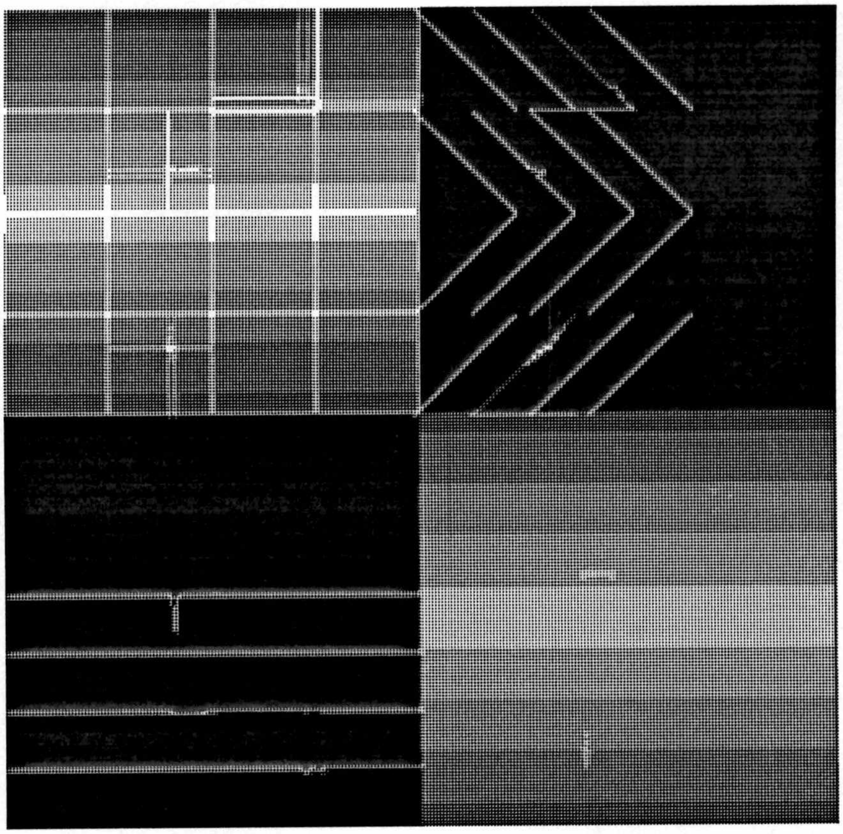


Figure 3.11: *Detection of defects for a* 256 × 256 *synthetic image with linear intensity variations along the y axis. Window size* 64 × 64. *Detection of a defect is indicated by marking horizontal and vertical lines across the windows.*

Table 3.5: *Results showing the detection system behavior for defect detection in synthetic images with linear intensity variation along x axis.*

Image	*Window*	*Defects Detected ?*	
Size	*Size*	*speckles*	*streaks*
256 x 256	32 x 32	*Yes*	*Yes*
256 x 256	64 x 64	*Yes*	*Yes*
256 x 256	128 x 128	*Yes*	*Yes*

Table 3.6: *Results showing the detection system behavior for defect detection in synthetic images with linear intensity variation along y axis.*

Image	*Window*	*Defects Detected ?*	
Size	*Size*	*speckles*	*streaks*
256 x 256	32 x 32	*Yes*	*Yes*
256 x 256	64 x 64	*Yes*	*Yes*
256 x 256	128 x 128	*Yes*	*Yes*

A real image for aluminum surface is depicted in Figure 3.12. Blemishes representing the streaks and speckles can be seen in the image shown in Figure 3.12. These defects represent surface irregularities present on aluminum surface when the given sample of aluminum was digitized. Figure 3.12 shows a 3-D surface plot representing intensity variations in the image. High peaks in this figure show defect intensities. Figure 3.14 illustrates the defect detection using a 128 × 128 window. Table 3.7 gives the results of multiple window size processing. In the case of a real image, all speckles are not detected for larger window, and as the window size is reduced the number of speckles that are detected increases. Streaks are detected for all window sizes considered, however, there are some speckles that are not detected under any window size. These results suggest that:

(i) the detection system is very sensitive to streaks,

(ii) smaller windows detect finer defects,

(iii) some defects are not detected at all by the system, especially small defects that are not elongated.

The sensitivity of the system to streaks can be explained by the fact that for streaks there are more pixels contributing to cause abrupt changes in profiles. The threshold value determines the minimum deviation in signatures that can be detected. Because the defect detection is based on analyzing the abrupt changes in the profile elements, a smaller defect with a low contrast may be unable to cause considerable deviations in the profiles to be detected for the given threshold.

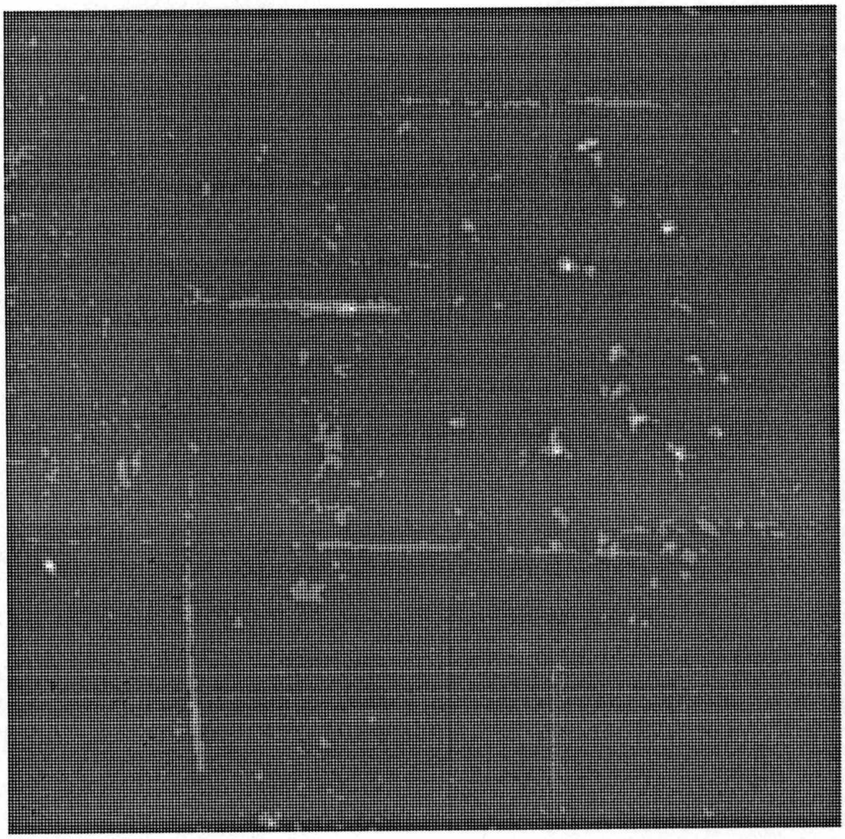


Figure 3.12: *A real image for aluminum surface. Defects can be identified as bright blemishes in the image. Image size* 256 × 256.

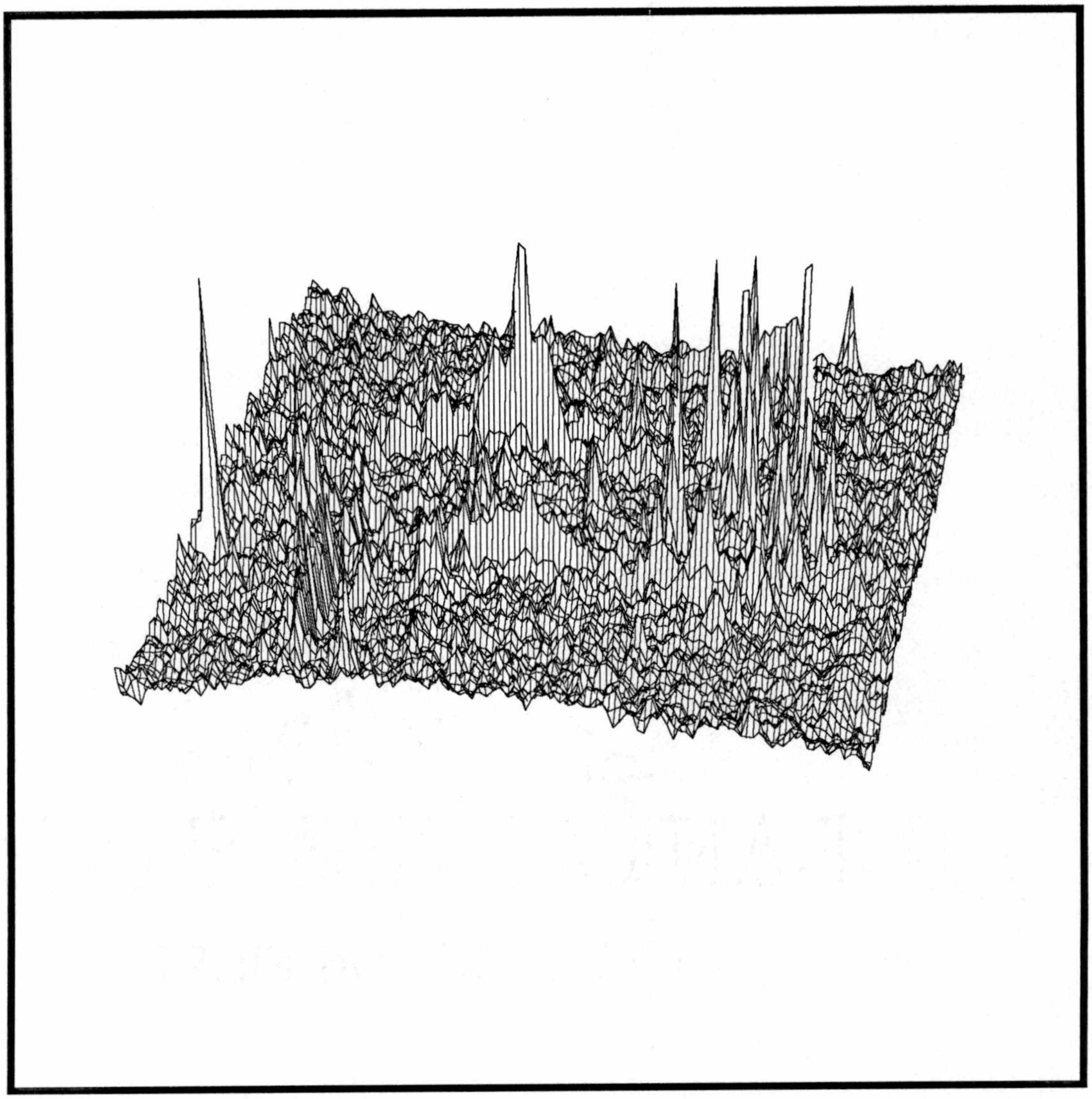


Figure 3.13: *A 3-D plot of the aluminum image given in Figure 3.12. This plot shows the intensity variations in the image, high peaks in the plot correspond to the defects pixel intensities.*

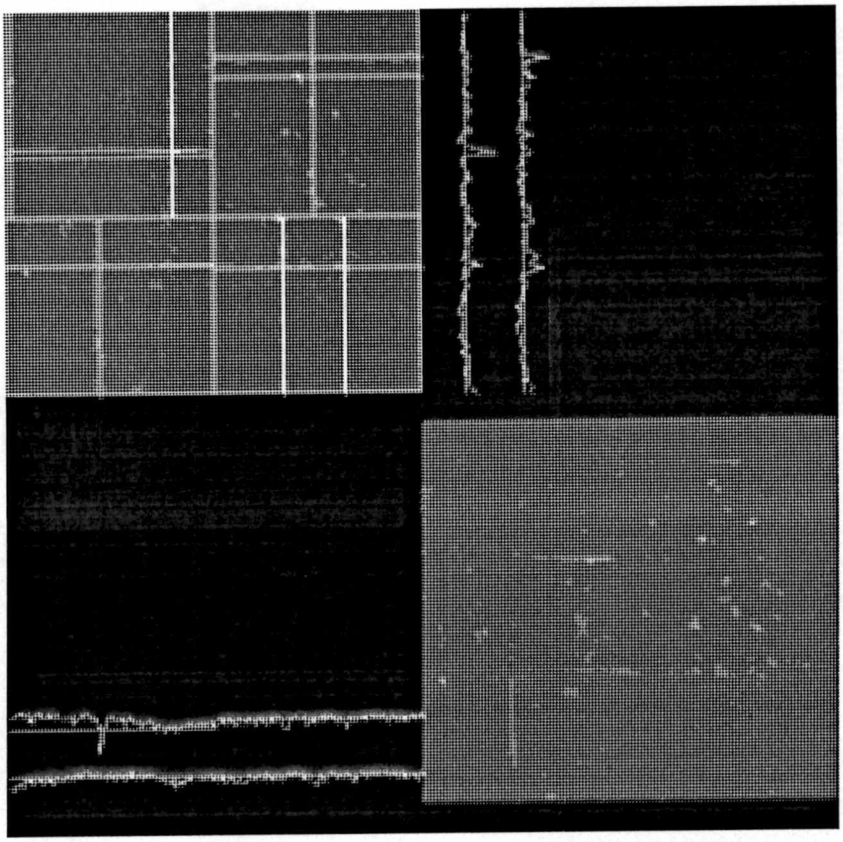


Figure 3.14: *Defect detection in the real image of aluminum surface. Window size* 128 × 128. *Detection of a defect is indicated by marking horizontal and vertical lines across the windows.*

Table 3.7: *System performance for defects detection in a real image of aluminum. Image is processed for multiple window sizes.*

Image Size	*Window Size*	*Defects Detected ?*		*Remarks*
		speckles	*streaks*	
256 x 256	128 x 128	*Yes*	*Yes*	*Not all speckles are detected*
256 x 256	64 x 64	*Yes*	*Yes*	
256 x 256	32 x 32	*Yes*	*Yes*	

Thus, defects falling under this category may not be detected. Therefore in addition to other factors, such as window size and defect size, the performance of the system for defect detection is dependent on the threshold value used. There is a trade-off between the smallest threshold that can be used and false detection of defects. Hence, suitable threshold for defect detection unit is selected empirically. In particular, threshold values that appear to work well are in the range:

$$3.5 \leq \ threshold \ \leq 5.0.$$

From experimental results for the images with known intensity distributions and real metal images, it is obvious that the detection system has industrial applications. Especially if the defects are elongated, as is the case in many industrial environments, this system can be used for visual inspection applications. After the defects are detected, the next important stage is to classify the defects. The results and performance of the classifier are described in detail in Chapter 4.

3.6 Summary

This chapter describes the functional elements of the defect detection system. Basic concepts underlying the detection system are presented with many illustrations. Also, various factors contributing to the system's performance were discussed. Finally, the applications of the detection system to synthetic and real images were presented. The results showed that the system has a potential for real world defect detection applications. However, the system's applications are useful to some specific kinds of defects. Classification of the defects is accomplished by a multilayer perceptron classifier. The next chapter describes the classifier in more detail.

CHAPTER 4

Multilayer Perceptron Classifier

This chapter introduces basic concepts about neural networks and describes applications of the multilayer perceptron network to defect classification. The classifier utilizes the *back-propagation* learning rule for training. Before the architecture and operation of the classifier are described, the common terminology used in connection with neural networks is reviewed.

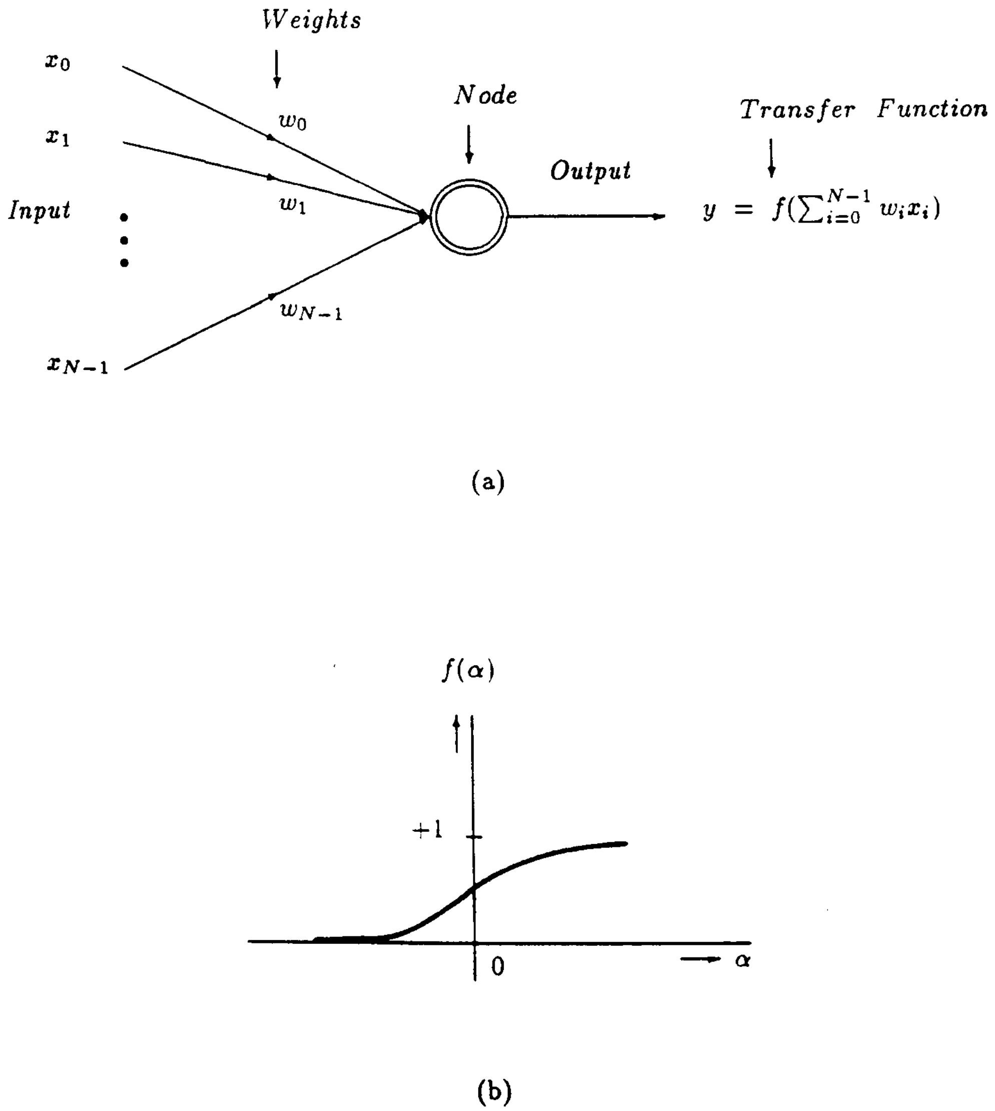
4.1 Basic Concepts

A neural network approximates or models a neuron or an ensemble of neurons. The neuron is the fundamental building block of the brain and nervous system. For detailed information about neural structure, functionality, and signal transmission mechanism in the nervous system refer to [4,20,25,31].

A neural network consists of computational elements or nodes, the links between the nodes, and the transfer function. Nodes are located in three layers: the input layer, the output layer, and the hidden layer. The information to be processed is presented to the input layer. The output layer produces class labels in the case of pattern classification applications. The hidden layer contains hidden units or nodes that are not directly connected to both the input and output nodes. A link is a unique line of communication from a sending node to a receiving node. There are two types of connections to a node: excitory and inhibitory. Inhibitory connections tend to prevent firing of the node, whereas, excitory connections tend to cause firing of the node. On each connection at the input of a node, there is a weight which is analogous to a synapse in a neuron. A weight controls the strength of the incoming signal to the node. Just as a neuron has a variety of structures and functions, a node, which is a mathematical approximation of a neuron, has a number of types and functions. A node adds N weighted inputs and passes the results through a transfer or threshold function (see Figure 4.2 (a)). The relationship between inputs and outputs at any instant is specified by the transfer function. There are many kinds of transfer functions, such as hard limiter, threshold logic, and sigmoid function. A particularly useful transfer function is the sigmoid function [23]. The *sigmoid function* has a high and a low saturation limit, and a smooth variation in between. The sigmoid function is defined as:

$$f(\alpha) = \frac{1}{1+e^{-\alpha}}. \tag{4.1}$$

The sigmoid function is 1 for a large positive value of α, 0 for a large negative value of α, and has a smooth transition for other α values. The parameter, α, is the node value at a certain time after the summation of the weighted inputs. Figure 4.2 (b) shows a sigmoid function. Parameter α is passed through a transfer function which produces the actual output for that time. The following analogies can be drawn between the artificial and biological neural networks.



(a)

(b)

Figure 4.1: *(a): A computational element which computes the weighted sum of N inputs; (b): A sigmoid function where $f(\alpha)$ is the transfer function, and α is the node value for a certain time after the addition of weighted inputs.*

- A node is the buliding block of artificial neural networks just as a neuron is the buliding block of biological neural networks.
- Weights in artificial neural networks are analogous to synapses in biological neural networks.

The next section describes the architecture of the multilayer perceptron network.

4.2 Multilayer Perceptron Network

The multilayer perceptron network is shown in Figure 4.2. The classifier used in this work is a three-layer perceptron with two hidden layers. It has been shown that a three-layer perceptron is capable of forming arbitrary decision regions [23,29,30].

Neural networks are specified in terms of topology of the network, node characteristics, and training rules. The training rules specify how to initialize the weights and modify those weights during operation of the network to improve performance. The topology of a three-layer perceptron network is shown in Figure 4.2, and the back-propagation training algorithm is described in the following section.

4.3 Back-Propagation Training Algorithm

Associated with each network is a learning rule, which specifies the way to initialize the weights and to modify these weights during the operation of the

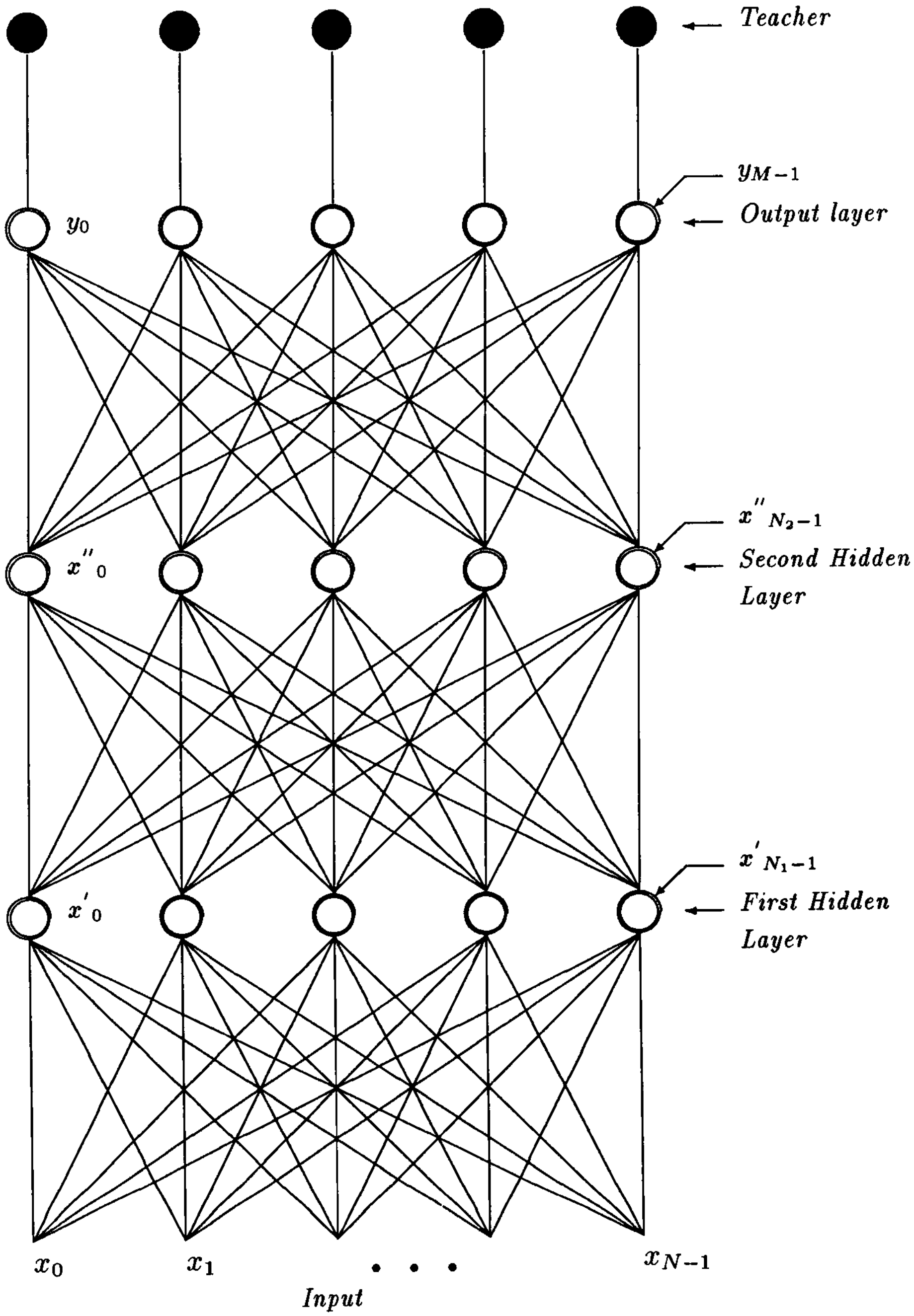


Figure 4.2: *A 3-layer perceptron network with two hidden layers. M is the number of outputs, N is the number of inputs, x' and x'' are the outputs of nodes in the first and second layer respectively. The network used for classification has $M = 2$, $N_2 = 35$, $N_1 = 30$, and $N = 17$.*

network. The learning rule is the very heart of a neural network. The perceptron network utilizes the back-propagation training rule. It is an iterative gradient algorithm designed to minimize the mean square error between the actual output of a multilayer perceptron network and the desired output. Back-propagation is a supervised learning method, which means the network is provided with labels that specify the correct class for each input pattern vector during training. Training involves presenting to the network the representative pattern vectors of each class, providing the correct class labels in such a way as to reduce the error towards the desired output [5,23]. The teacher specifies the correct class labels during training to enable the network to adjust its weights so that the error between actual output and desired output is minimized. The main steps involved in the back-propagation algorithm are outlined below:

Step 1. Initialize all the weights to small random numbers.

Step 2. Input vector $x_0, x_1, \ldots x_{N-1}$ and provide the desired outputs $d_0, d_1, \ldots d_{M-1}$. The samples from a training set are presented randomly with each sample having equal likelihood until weights stabilize.

Step 3. Calculate outputs $y_0, y_1, \ldots y_{M-1}$, as follows:

$$\begin{aligned} y_m &= f(\sum_{k=0}^{N_2-1} w''_{km} x''_k) && 0 \geq m \leq (M-1), \\ x''_k &= f(\sum_{j=0}^{N_1-1} w'_{jk} x'_j) && 0 \geq k \leq (N_2-1), and \\ x'_j &= f(\sum_{i=0}^{N-1} w_{ij} x_i) && 0 \geq j \leq (N_1-1), \end{aligned}$$

where x_i is the input, and x'_j and x''_k are the outputs of nodes in the first and second hidden layers. The connection strength from the input to the first hidden layer is w_{ij}, and w'_{jk} and w''_{km} are the connection strengths between the first and second and between the second and the output layers, respectively.

Step 4. Starting at the output nodes and working back down to the first hidden layer, adjust weights by

$$w_{ij}(t+1) = w_{ij}(t) + \eta\delta_j x'_j.$$

In this equation $w_{ij}(t)$ is the weight from hidden node i or from an input to node j at time t; x'_j is either an output of node i or is an input; η is a gain term; and δ_j is an error term for node j. The error term, δ, is calculated differently for the top level cells and lower level cells. If node j is an output node, then

$$\delta_j = y_j(1-y_j)(d_j-y_j),$$

where d_j is the desired output of node j (teacher value) and y_j is the actual output. The values of the teacher are either high (0.9) or low (0.1) . The high value is the m^{th} entry of the teacher corresponding to class m. If node j is an internal hidden node (i.e., lower level node), then

$$\delta_j = x'_j(1-x'_j)\sum_k \delta_k w_{jk},$$

where k is over all nodes in the layers above node j.

Step 5. Go to step 2.

The next section describes the training and testing of the perceptron network.

4.4 Training and Testing of The Network

The defect signature generation process is described in Chapter 3. After the defect signatures are generated by the defect detection stage, they are ready for input to the classifier. The network has two modes of operations: training and testing. In the training mode, the representative pattern vectors for defect signatures of streak and speckle are input to the network. The term defect signature and pattern vector is used interchangeably. Because back-propagation is a supervised learning method, a teacher is used that provides the correct class label for each of the inputs. The network uses this information to propagate an error from the output layer to the first hidden layer. Error for each layer is calculated according to the back-propagation rule described in Section 4.3. The purpose of error propagation is to modify the weights in such a way that the mean square error between the actual classifier output and the desired output is minimized. The network is considered to have learned the inputs if it can correctly classify these inputs. Thus, training is an iterative process and it is continued until the network has learned the inputs. During the testing mode, pattern vectors of streak and speckle that are not seen by the network are input to the network and the network provides a class label for each of the inputs.

The training set is a list of matched inputs and class labels. Our training set consists of equal numbers of streak and speckle pattern vectors. Each pattern vector is normalized and augmented. Normalization ensures that equal weight is given to every element of the pattern vector and that larger elements do not dominate smaller elements. The network is randomly presented with the training pattern vectors iteratively until it correctly classifies the training patterns. At this point, the network is tested by presenting streak and speckle pattern vectors that are not known to the network. The class label corresponding to each testing pattern vector is given by the output layer of the network. As we are dealing with a two-class problem, there are two outputs, one for each class. The class of an input pattern vector is interpreted by looking at the *high* value of the output layer.

Because real metal image samples containing a wide variety of streaks and speckles are not available, training and testing data is generated by modeling streaks and speckles with ellipses. As described in Chapter 3, the length of the major and minor axes of the ellipse can be changed to model a variety of streaks and defects. In addition, deformation of both intensity and shape can be used to produce a realistic defect appearance.

The defect signatures are input as 17-element pattern vectors to the network. As described in Chapter 3, to a generate defect signature those elements of the horizontal and vertical signatures that are below a certain threshold are set to zero. Thus, each defect signature may have elements equal to zero. Before the defect signatures are input to the classifier, each defect signature is rearranged as a 16-element pattern vector in such a way that all leading and trailing zero elements of the defect signatures are ignored. Then the resulting 16-element pattern vectors are normalized between 0 to 1 and augmented by the 17 element. To normalize, divide the pattern vectors by window size then linearlly scale between 0 and 1 using the minimum intensity value of 0 and the maximum intensity value of 255, i.e.,

$$scaled \ \ element = \frac{(unscaled \ \ element) - \ min}{(max \ - \ min)}.$$

The 17-element pattern vectors are then input to the classifier.

In our testing of the network, 20 pattern vectors are used both in the training and testing data sets. The network dimensions and the gain term used for the testing of the network are as follows:

- input size = 17,
- first layer = 30,
- second layer = 35,
- output layer = 2,
- gain term = 0.2.

The results of training the network indicate that the network learns the defect signatures and is well stabilized by 300 iterations. Two pattern vectors of the testing set belonging to speckles were misclassified as streaks. The network correctly classifies the streak and speckle signatures with an accuracy of 90%.

4.5 Summary

This chapter describes the multilayer perceptron network that uses backpropagation as a learning rule. The network has two modes of operation: training and testing. The training set consists of streak and defect signatures. The network correctly classifies the streak and speckle signatures with a high accuracy.

CHAPTER 5

Conclusions

An approach for defect detection and classification in the digitized images of metals with known intensity distributions has been presented. The reflectance response of the specular surface has been investigated that has helped in understanding the specular behavior of metal surfaces such as aluminum, nickle, copper, etc. A fast flaw detection algorithm has been presented that improves computational efficiency (processing time) by reusing computations of the previous window size in the subsequent larger windows. A multilayer perceptron network using the back-propagation of the error learning rule has been utilized for classification of defect signatures. The vision system that has been developed for defect detection and classification has the following features:

- A two-dimensional image is processed by generating its two one-dimensional signatures. The processing of one-dimensional signals reduces computing time.
- An image with defects is processed for multiple window sizes, and the computations performed for a given window size are utilized by the next window size, considerably reducing the number of operations, which leads to a faster algorithm.

- The detection system is found to be more sensitive to streaks than speckles.
- The signatures of streaks and speckles are input to the multilayer layer perceptron network for classification.
- The network correctly classifies the streak and speckle signatures with an accuracy of 90%.

It has been shown that the spectral response of a defect free surface is different from the response of a surface with defects. Moreover, defects are more dominant at certain wavelentghs and only the corresponding image plane need be processed.

BIBLIOGRAPHY

BIBLIOGRAPHY

[1] E. Bahar, "Full-Wave Solutions for the Scattered Radiation Fields from Rough Surfaces with Arbitrary Slope and Frequency," *IEEE Transactions on Antennas and Propagation*, vol. AP-28, no. 1, pp. 11–21, 1980.

[2] M. L. Baird, "A Computer Vision System for Automated IC Chip Manufacture," *IEEE Trans. Syst. Man Cyber.*, vol. SMC-8, pp. 133–139, 1978.

[3] B. G. Batchelor, "Iteracting Image Analysis as a Tool for Industrial Inspection," *IEEE J. Comput. Digital Tech.*, vol. 2, pp. 61–70, 1979.

[4] H. E. Beck, *Simulation of Physiological Signal Processing Mechanisms in the Visual Tract.* Master's thesis, The University of Tennessee, Knoxville, TN, August 1990.

[5] H. E. Beck, D. Mcdonald and D. Brzakovic, "A Self-training Visual Inspection System with a Neural Network Classifier ," in *Proc. of the International Joint Conference on Neural Networks*, (Washington, DC), vol. 1, pp. 307–312, June 1989.

[6] H. E. Beck, D. Mcdonald and D. Brzakovic, "A Self-training Inspection System for the On-line Inspection of Printed Material," in *Proc. of the SPIE OE/Aerospace Sensing Symposium*, (Orlando, FL), April 1990.

[7] P. Beckmann and A. Spizzichino, *The Scattering of Electromagnetic Waves from Rough Surfaces.* Oxford, England: Pergamon Press, 1963.

[8] H. E. Bennett and J. O. Porteus, "Relation Between Surface Roughness and Specular Reflectance at Normal Incidence," *Journal of the Optical Society of America*, vol. 51, pp. 123–129, 1961.

[9] J. F. Blinn, "Models of Light Reflection for Computer Synthesized Pictures," *ACM Computer Graphics (SIGGRAPH 77)*, vol. 11, no. 2, pp. 192–198, 1977.

[10] C. B. Chittineni, " Signal Classification for Automatic Industrial Inspection," in *Proc. of IEEE Pattern Recognition Image Processing*, pp. 330–335, 1981.

[11] R. L. Cook, *A Reflection Model for Realistic Image Synthesis.* Master's thesis, Cornell University, Ithaca, NY, 1981.

[12] R. L. Cook and K. E. Torrance, "A Reflection Model for Computer Graphics," *ACM Trans. on Graphics*, vol. 1, no. 1, pp. 9–24, January 1982.

[13] J. A. Feldman and D. H. Ballard, "Connectionist Models and Their Properties," *Cognitive Science*, vol. 6, pp. 205–254, 1982.

[14] J. D. Foley and V. Dam, *Fundamentals of Interactive Computer Graphics.* Reading, PA: Addison-Wesley Publishing Company, 1984.

[15] P. A. Godinez, "Inspection of Surface Flaws and Textures," *Sensors*, vol. 4, no. 6, pp. 27–32, 1987.

[16] H. Gouraud, "Continuous Shading of Curved Surfaces," *IEEE Trans. on Computers*, vol. c-20, no. 6, pp. 623–629, June 1971.

[17] E. L. Graminski and R. A. Kirsch, " Image Analysis in Paper Manufacturing," in *IEEE Conference on Pattern Recognition and Image processing*, pp. 137–143, 1977.

[18] R. Hall, *Illumination and Color in Computer Generated Imagery.* New York, NY: Springer-Verlag, 1989.

[19] J. J. Hopfield, "Neural Networks and Physical Systems with Emergent Collective Computational Abilities," in *Proc. of Natl. Acad. Sci. USA*, pp. 2554–2558, April 1982.

[20] J. J. Hopfield, "Neurons with Graded Response Have Collective Computaional Properties Like Those of Two-State Neurons," in *Proc. of Natl. Acad. Sci. USA*, pp. 3088–3092, May 1984.

[21] D. B. Judd and G. Wyszecki, *Color in Business, Science, and Industry.* New York, NY: John Wiley and Sons, 1975.

[22] K. J. Koestner, *A Wave Based Reflection Model for Realistic Image Synthesis.* Master's thesis, Cornell University, Ithaca, NY, 1986.

[23] R. P. Lippman, "An Introduction to Computing with Neural Nets," *IEEE ASSP Magzine*, vol. , pp. 4–22, April 1987.

[24] G. W. Meyer and D. P. Greenberg, "Perceptual Color Spaces for Computer Graphics," *Computer Graphics*, vol. 14, no. 3, pp. 254–261, 1980.

[25] M. M. Nelson and W. T. Illingworth, *A Practical Guide to Neural Nets.* Reading, MA: Addison-Wesley Publishing Company, 1990.

[26] Y. Okawa, "Automatic Inspection of the Surface Defects of Cast Metals," *Comput. Vision, Graphics, and Image Processing*, vol. 25, pp. 89–112, 1984.

[27] B. T. Phong, "Illumination for Computer Generated Pictures," *Communications of the ACM*, vol. 18, no. 8, pp. 311–317, 1975.

[28] G. B. Porter and J. L. Mundy, "Visual Inspection System Design," in *Proc. of IEEE IECON'82 Conf.*, (Palo Alto, CA), pp. 149–151, Nov. 1982.

[29] R. Rosenblatt, *Principles of Neurodynamics.* New York, NY: Spartan Books, 1959.

[30] D. E. Rumelhart and J. L. McClelland, *Parallel Distributed Processing: Exploration in the Microstructure of Cognition.* Cambridge, MA: MIT Press, 1986.

[31] J. Stanley, *Introduction to Neural Networks.* Sierra Madre, CA: California Scientific Software, 1990.

[32] K. E. Torrence and E. M. Sparrow, "Polarisation, Directional Distribution, and Off-Specular Peak Phenomena in Light Reflected from Roughened Surfaces," *Journal of the Optical Society of America*, vol. 56, no. 7, pp. 916–925, 1966.

[33] K. E. Torrence and E. M. Sparrow, "Theory of Off-Specular Reflection from Roughened Surfaces," *Journal of the Optical Society of America*, vol. 57, no. 9, pp. 1105–1114, 1967.

[34] P. University, "Thermophysical Properties of High Temperature Solid Materials," *Technical Report*, vol. 1, pp. 25–28, 1967.

VITA

Iftekhar Hussain was born in a small town Behlol, District Gujranwala, Pakistan on January 2, 1964. He attended Government Middle School Bhiri Khurd. He graduated from Government Middle School Bhiri Khurd in 1976 and joined Government Muslim Model High School, Lahore, Pakistan. He graduated from Government High School in 1978 with an overal third position in the School. He joined Government College Lahore, Pakistan in 1978 and graduated with high first division in 1980. He started his udergraduate study in Electrical Engineering at The University of Engineering and Technology, Lahore, Pakistan in February 1982. He graduated in September 1986 from The University of Engineering and Technology, Lahore, Pakistan obtaining B.Sc. degree in Electrical Engineering. From June 1987 to February 1988, he worked as a design engineer for Water and Power Development Authority, Lahore, Pakistan.

He joined the Department of Electrical Engineering at the University of Tennessee, Knoxville toward a Master of Science in Electrical Engineering in March 1988. He was a teaching assistant from Fall 1988 to Spring 1991 and a research assistant from Fall 1989 to Spring 1991. His research was concentrated in the area of Image Processing.