

1995

# Automated image inspection using wavelet decomposition and fuzzy rule-based classifier

Zhong Zhang  
Iowa State University

Follow this and additional works at: <https://lib.dr.iastate.edu/rtd>

 Part of the [Artificial Intelligence and Robotics Commons](#), and the [Electrical and Electronics Commons](#)

## Recommended Citation

Zhang, Zhong, "Automated image inspection using wavelet decomposition and fuzzy rule-based classifier " (1995). *Retrospective Theses and Dissertations*. 10746.  
<https://lib.dr.iastate.edu/rtd/10746>

This Dissertation is brought to you for free and open access by the Iowa State University Capstones, Theses and Dissertations at Iowa State University Digital Repository. It has been accepted for inclusion in Retrospective Theses and Dissertations by an authorized administrator of Iowa State University Digital Repository. For more information, please contact [digirep@iastate.edu](mailto:digirep@iastate.edu).

## **INFORMATION TO USERS**

**This manuscript has been reproduced from the microfilm master. UMI films the text directly from the original or copy submitted. Thus, some thesis and dissertation copies are in typewriter face, while others may be from any type of computer printer.**

**The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleedthrough, substandard margins, and improper alignment can adversely affect reproduction.**

**In the unlikely event that the author did not send UMI a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.**

**Oversize materials (e.g., maps, drawings, charts) are reproduced by sectioning the original, beginning at the upper left-hand corner and continuing from left to right in equal sections with small overlaps. Each original is also photographed in one exposure and is included in reduced form at the back of the book.**

**Photographs included in the original manuscript have been reproduced xerographically in this copy. Higher quality 6" x 9" black and white photographic prints are available for any photographs or illustrations appearing in this copy for an additional charge. Contact UMI directly to order.**

# **UMI**

A Bell & Howell Information Company  
300 North Zeeb Road, Ann Arbor, MI 48106-1346 USA  
313/761-4700 800/521-0600



**Automated image inspection using wavelet decomposition  
and fuzzy rule-based classifier**

by

**Zhong Zhang**

**A Dissertation Submitted to the  
Graduate Faculty in Partial Fulfillment of the  
Requirements for the Degree of  
DOCTOR OF PHILOSOPHY**

**Department: Electrical and Computer Engineering  
Major: Electrical Engineering (Communications and Signal Processing)**

**Approved:**

Signature was redacted for privacy.

**In Charge of Major Work**

Signature was redacted for privacy.

**For the Major Department**

Signature was redacted for privacy.

**For the Graduate College**

**Iowa State University  
Ames, Iowa**

**1995**

**UMI Number: 9531813**

---

**UMI Microform 9531813**

**Copyright 1995, by UMI Company. All rights reserved.**

**This microform edition is protected against unauthorized  
copying under Title 17, United States Code.**

---

**UMI**

**300 North Zeeb Road  
Ann Arbor, MI 48103**

---

**TABLE OF CONTENTS**

<b>ACKNOWLEDGMENTS</b> .....	ix
<b>CHAPTER 1. INTRODUCTION</b> .....	1
Automated Flaw Detection .....	3
Research Objective .....	6
Approach .....	6
Overview of Chapters .....	8
<b>CHAPTER 2. COMPUTERIZED IMAGE INSPECTION</b> .....	10
Image Local Feature Extraction .....	11
Localized Image Classification .....	20
<b>CHAPTER 3. WAVELET DECOMPOSITION</b> .....	22
Fundamentals of the Wavelet Transform .....	23
Haar Transform .....	27
Compactly Supported Wavelet Transform .....	32
<b>CHAPTER 4. WAVELET BASED LOCAL FEATURE EXTRACTION</b> .....	49
Two Dimensional Wavelet Transform .....	49
Local Feature Extraction from Wavelet Decomposition .....	55

---

<b>CHAPTER 5. FUZZY RULE-BASED CLASSIFICATION .....</b>	<b>65</b>
Fuzzy Logic and Fuzzy System .....	66
Adaptive Fuzzy Classification .....	70
Optimized Fuzzy Classification .....	79
<b>CHAPTER 6. INTERACTIVE HUMAN-MACHINE INTERFACE .....</b>	<b>94</b>
Graphical User Interface .....	94
Generation of Training Prototypes .....	97
Machine Training and Image Inspection .....	98
<b>CHAPTER 7. EXPERIMENTAL RESULTS .....</b>	<b>102</b>
The Detection of Flaws in Aluminum Castings .....	103
The Detection of Flaws in Welds .....	111
The Detection of Flaws in an Air Conditioner Part .....	124
Performance Measure for Image Classification .....	128
<b>CHAPTER 8. SUMMARY AND CONCLUSIONS .....</b>	<b>135</b>
<b>BIBLIOGRAPHY .....</b>	<b>142</b>

**LIST OF TABLES**

Table 3.1:	The filter parameters for Daubechies wavelet transform (part 1) .....	40
Table 3.2:	The filter parameters for Daubechies wavelet transform (part 2) .....	41
Table 3.3:	The filter parameters for Daubechies wavelet transform (part 3) .....	42
Table 3.4:	The filter parameters for Daubechies wavelet transform (part 4) .....	43
Table 3.5:	The normalization factors for Daubechies wavelet decomposition .....	47
Table 5.1:	Prototype feature vectors and the desired output values .....	75
Table 5.2:	Simulated training data and the desired output classes .....	87
Table 7.1:	The detection of various flaws in weld .....	123

---

## LIST OF FIGURES

Figure 2.1:	The block diagram of an automated image inspection system .....	12
Figure 2.2:	Extracting image features inside a local window .....	14
Figure 3.1:	A vector is represented by its orthogonal basis .....	28
Figure 3.2:	A vector is represented by another orthogonal basis .....	29
Figure 3.3:	The Haar pyramid for a vector of size 4 .....	31
Figure 3.4:	The compactly supported Daubechies wavelets $W(t)$ .....	34
Figure 3.5:	The scaling function, $\Phi(t)$ , of DAUB4 wavelet transform .....	39
Figure 3.6:	The wavelet function, $W(t)$ , of DAUB4 wavelet transform .....	39
Figure 3.7:	The pyramid computation of DAUB4 wavelet transform .....	44
Figure 3.8:	A one dimensional signal with differently scaled wavelets .....	45
Figure 3.9:	The regular wavelet transform of the signal in Figure 3.8 .....	46
Figure 3.10:	The amplitude normalized wavelet transform of the signal in Figure 3.8 .....	48
Figure 4.1:	A test image .....	51
Figure 4.2:	The first level wavelet decomposition of Figure 4.1 .....	52
Figure 4.3:	The second level wavelet decomposition of Figure 4.1 .....	53
Figure 4.4:	The sketch of a 2-dimensional image .....	57
Figure 4.5:	The sketch of the first level of decomposition .....	57

---

Figure 4.6:	The sketch of the second level of decomposition .....	58
Figure 4.7:	A weld X-ray image .....	62
Figure 4.8:	The smoothed feature extracted from the second level of decomposition of the image in Figure 4.7 .....	62
Figure 4.9:	The decomposed feature extracted from the second level of decomposition of the image in Figure 4.7 .....	63
Figure 4.10:	The decomposed feature extracted from the third level of decomposition of the image in Figure 4.7 .....	63
Figure 5.1:	The tallness membership function as a function of height .....	67
Figure 5.2:	The block diagram of a fuzzy system .....	69
Figure 5.3:	A sketch of Gaussian membership functions .....	73
Figure 5.4:	Selectively establish the fuzzy rule base from the training data .....	78
Figure 5.5:	The asymmetric trapezoidal fuzzy membership function .....	80
Figure 5.6:	The optimized training from the training data .....	84
Figure 5.7:	Establish the optimal fuzzy rule base from the training data .....	88
Figure 5.8:	The feature distance in a 3-D fuzzy hyper cube .....	90
Figure 6.1:	HAPXM graphical user interface .....	96
Figure 6.2:	Creating the training data from training images .....	99
Figure 6.3:	The dialog box for selecting training parameters .....	101
Figure 7.1:	The X-ray image of an aluminum casting .....	104
Figure 7.2:	A void flaw in the aluminum casting of Figure 7.1 .....	104
Figure 7.3:	The four features extracted from wavelet decompositions of Figure 7.2 .....	105
Figure 7.4:	The image segmentation result of Figure 7.2 .....	107
Figure 7.5:	The detection of a smaller flaw .....	108

Figure 7.6:	The detection of a flaw at another location .....	109
Figure 7.7:	The detection of different types of flaws .....	110
Figure 7.8:	A gas hole flaw in weld, example 1 .....	112
Figure 7.9:	A gas hole flaw in weld, example 2 .....	112
Figure 7.10:	A gas hole flaw in weld, example 3 .....	113
Figure 7.11:	The optimized classification of Figure 7.8 .....	113
Figure 7.12:	The optimized classification of Figure 7.9 .....	114
Figure 7.13:	The optimized classification of Figure 7.10 .....	114
Figure 7.14:	The adaptive fuzzy classification of Figure 7.8 .....	116
Figure 7.15:	The adaptive fuzzy classification of Figure 7.9 .....	116
Figure 7.16:	The adaptive fuzzy classification of Figure 7.10 .....	117
Figure 7.17:	The adaptive fuzzy classification using 25% of the training data .....	118
Figure 7.18:	An inclusion flaw in weld .....	119
Figure 7.19:	The optimal classification of Figure 7.18 .....	119
Figure 7.20:	A crack flaw in weld .....	121
Figure 7.21:	The optimal classification of Figure 7.20 .....	121
Figure 7.22:	An under-cut flaw in weld .....	122
Figure 7.23:	The optimal classification of Figure 7.22 .....	122
Figure 7.24:	The X-ray image of an air conditioner part .....	125
Figure 7.25:	The flaw detection result of Figure 7.24 .....	125
Figure 7.26:	The intensity stretching of Figure 7.24 .....	126
Figure 7.27:	The transfer function for intensity stretching .....	127
Figure 7.28:	The flaw detection result of Figure 7.26 .....	128

---

Figure 7.29:	The classification time for different fuzzy classifiers .....	129
Figure 7.30:	The training time for different fuzzy classifiers .....	130
Figure 7.31:	The casting image corrupted by Poisson noise .....	131
Figure 7.32:	Classification without noise training .....	132
Figure 7.33:	Classification with noise training .....	133

## ACKNOWLEDGMENTS

I would like to express my sincere thanks to my major professor Dr. John P. Basart for his guidance and encouragement throughout my research.

I would also like to thank Dr. Noel A. C. Cressie, Dr. Jennifer L. Davidson, Dr. Joseph N. Gray, Dr. Hsien-sen Hung and Dr. Lalita Udpa for serving on my graduate committee.

Finally, special thanks to my wife, Fang, for her emotional support throughout these years.

This work was sponsored by the NSF Industry/University Center for NDE at Iowa State University.

---

## CHAPTER 1. INTRODUCTION

Automated image inspection is an artificial intelligent technique that enables a computer to understand images containing complex scenes. There are many computer vision systems developed for different applications, ranging from medical diagnosis, industrial robots, to intelligent weapon systems. The automated image inspection system discussed in this dissertation is a general purpose architecture that can be applied in most of image formations, while the objective of the system development is to create a general purpose computer vision system to automatically detect flaws contained in images generated for nondestructive evaluation applications.

The purpose of nondestructive evaluation (NDE) is to detect flaws or defects inside objects or device during their manufacturing processes or while they are still in service, without destroying their physical structures and functions. NDE is especially important in situations where destructive inspections are not feasible in nature or in cost, such as medical inspection and of human body, routine maintenance of aircraft and motor vehicles, and quality control of critical components of devices. Images are usually involved in today's NDE applications because techniques such as X ray, ultrasound, eddy current, or microwave provide ways to reach inside an object, and create pictures showing the inside structure of the objects to ease the vision limits of human beings. X-ray

---

radiography is the most direct imaging approach in NDE and is a widely used inspection technique in manufacturing industries and health care business. While in industry or health care business, most X-ray radiographs are currently inspected by specially trained personnel to recognize the irregular appearances in the films and therefore detect defects. The results of human inspections are not usually consistent because they may be affected by the working skill and the concentration of a particular person, the sensitivities of workers' eyes, and even the general stress and environment. These and other factors cause the inspection results to vary from time-to-time. On the other hand, a human's inspection and decision making abilities are much more superior than any other artificial means, thanks to our naturally sophisticated vision and brain systems.

With the appearance of high performance computers, it is possible to use computers to help people accomplish their work. Many image processing techniques such as noise reduction, edge detection, intensity manipulation and background trend removal, have been developed in computer programs to enhance the visual quality of image appearance and help people to see the targets. In some cases, the results of these image enhancement techniques may lead to ideal results in which flaws are successfully separated from the image background, and can therefore be detected automatically by the computers through simple thresholding. But in most of the applications, the objects to be inspected have complex shapes and structures. These structures result in complicated background intensity variations in the image and hence make the separation of flaws and background more difficult. Therefore, more sophisticated approaches must be used to detect flaws automatically in the presence of complicated image background and noises.

### Automated Flaw Detection

Compare to manual inspection of NDE images, computer integrated automated inspection system will be able to perform flaw detection task in a more consistent and cost effective way. A simple and straight forward approach to conduct quick detection is to use the difference between an image under test and a reference image of a good sample shot generated under the same conditions. The difference approach was widely used the past mainly due to its simplicity and the lack of more sophisticated approaches [Closier, 1981]. However, it requires excellent reproducible positioning and sizing of test and reference samples, which is very difficult to maintain during mass productions. Decker has applied a nonlinear image warping technique to overcome the alignment difficulty in the difference method [Decker, 1983], but the method is computationally expensive and the results are far from ideal.

For many years, a lot of effort has been made to develop sophisticated automated image inspection systems. The systems developed so far are mostly customized systems. The customized systems are usually specific systems that were built on specific distinctions in an image and were tailored for particular types of applications. They are built upon typical appearances of targets and backgrounds in *ad hoc* cases, and flaws are detected directly according to their distinct properties. Some examples of such systems are: a flaw inspection package for aluminum wheel castings [Boerner and Strecker, 1988], which is limited only to "bubble" type voids in wheel castings; an automatic flaw detection package for welds in space shuttle fuel tanks [Basart and Xu, 1991], which is designed for the structure of welds and various flaw types; a medical radiograph inspecting software that can only detect round nodules in chest X-rays [Gatot, 1988]; and the algo-

---

rithms developed for line and edge detection [Haralick, 1984; Nevatia and Babu, 1980, Sklansky, 1978] or shape matching [Davis, 1979; Bhanu and Faugeras, 1984; You and Jain, 1984]. The customized systems have the advantages of relatively fast speed and reasonably good liability, but once system is developed for a specific type of application, it can not be used in other applications without substantial changes in the program itself. In order to make a flaw detection system usable for most of the NDE images, the system should be a general purpose one and not limited by the type of object it is able to detect. Some general purpose detection schemes have been tested in the Center for NDE of Iowa State University to detect flaws in X-ray radiographs. Ulmer [Ulmer, 1993] has developed an automated inspection system which is similar to the difference approach. Instead of using real images as references, the approach utilizes piece-wise polynomial surface fitting algorithm to simulate the background surfaces of the test image and uses the simulated image as the reference. The surface simulation is done by a range image segmentation approach [Besl and Jain, 1988] that breaks an image into several sub-images based on different surface types. Each sub-image is simulated piece by piece using polynomial surface fitting and seed growing techniques. After all the sub-image surfaces are completed, they are reassembled into a big image which was a simulation (fit) of the background surface of the original image. The simulated background image is then subtracted from the original image in the hope that flaws and noise are the only remains in the residual image after the subtraction. The problem with this approach is that the piece-wise surface fitting is a computationally expensive process and is therefore very time consuming. In addition, seed sizes must be larger than flaw sizes, otherwise all the flaws larger than seeds will be fitted as the background. Another difference based approach

uses CAD model to generate image background [Siwek, 1994], and uses the CAD image as the reference image. In this case, a sophisticated CAD model is needed in advance to generate surface simulations, and all the geometric and material parameters must be present to precisely simulate the actually filming condition in order to get an accurate simulation. Both approaches mentioned above subtract a simulated surface image from the original image and a residual image is obtained from the subtraction. Their difference lies in the way simulation is performed, the first approach is data driven while the latter one is model based. Under perfect conditions, both approaches can remove complicated image background and detect flaws very easily in the residual image. The subtracting can improve visual appearance of the image and help a human to identify flaws in the residual image. However, if the imaging condition is less than ideal, the simulated image may not exactly match the background of the original image. In this case, the difference between the background simulation and the true background causes artifacts in the residual image and therefore makes it harder for a computer to find the flaw.

Another software package [Lyseggen, 1994] recently developed for automatic flaw detection combines image feature extraction and classification into one frame. It is developed under Windows 3.1. The first order and second order statistics are computed within a local window to characterize local texture features in the image. Various common classification algorithms are used to classify the features. The software is slow in computing features but is proven effective for homogeneous texture classifications. However, it is less effective for a lot of real images because the windowed approach only computes statistics inside a small window and therefore loses the information beyond the window.

## **Research Objective**

In order to make automated image inspection task more efficient and more reliable, it is desirable to have a data driven, fast and dependable computer vision system architecture that can be applied on many flaw detection applications, but can also be customized easily for a specific problem without changing the structure of the system. The research objective of this dissertation is to develop such a system. This system should be a general purpose one that can handle most of the problems encountered in NDE flaw detection applications, but specific enough to solve any particular application as effective as a customized system. The target system is an artificial intelligent vision system that emulates some functions of human visual and brain functions, but is able to fully utilize the strength of a computer in terms of consistency and cost effectiveness. The computer must be able to understand both local and contextual contents in an image, and classify the image based on what it understands in a fast and effective way. The completed system is also aimed to fill the gap between lab research and field practice. Which means that in addition to the theoretical ground work involving image understanding and decision making techniques, it should be a user friendly system so that an industry user without significant knowledge of computer vision, coding and pattern recognition will be able use it after adequate training. Therefore users in NDE and other related fields will benefit from the result of this research work .

## **Approach**

An image segmentation approach is used for automatic flaw detection. The image segmentation process divides an image into small segments and each segment represents

different objects or structures contained in the image. A computer is used to emulate the understanding and thinking abilities of human beings that can easily distinguish various contents in an image. Image segmentation by a computer is usually achieved with two stages, image feature extraction and decision making.

The image feature extraction task is performed by an image feature extraction process which converts two-dimensional gray tones of a digital image into a series of two dimensional data array containing sensible numbers that are features of the image and can be easily recognized by the computer. The decision making process analyzes the features extracted and decides which part of the image belongs to flaw or background. Since the purpose of an automated flaw detection system is to determine the location and the size of each flaw in the presence of the image background, the features extracted from the image are preferably local features characterizing the properties of a local portion in the image. An image is examined pixel-by-pixel by a knowledge-based decision making device according to the analytical results of the local features extracted. The knowledge base that is used to control the performance of the decision making device is derived from training prototypes for a particular application. The output of the system is a labeled image in which various local portions of the image are marked by various colors (or gray tones) chosen by the user to indicate different classes such as flaw and non-flaw in the image.

A graphical user interface (GUI) is built to establish easy access to the system and the interactive human-machine interfacing. With the graphical user interface, a user is able to work interactively with the computer through mouse drawing, pointing and click-

---

ing, and display images, features, and classification results on a computer screen to actually view the pictures.

### **Overview of Chapters**

This dissertation contains eight chapters. Chapter 1 is an introduction that gives a general description of the problem. Chapter 2 is an overview of automated image inspection approaches and applications. It covers general materials about localization, feature extraction, and classification. Chapter 3 discusses orthogonal basis and signal decomposition, Haar transform, multi-resolution pyramid decomposition algorithm, discrete compactly supported wavelet transform and its normalization. Chapter 4 describes the wavelet transform as it is applied to two dimensional images, the approach to extract local harmonic information from the wavelet decomposition of an image, and a multiresolution representation of the image features. Chapter 5 constructs the classifiers used for pixel classification, gives basic concepts of fuzzy logic, discusses fuzzification, fuzzy inference, defuzzification processes and the membership functions used in rule-based classifiers, describes the creation and the optimization of fuzzy rule bases from training data file, and gives a correctness measure for each classification. Chapter 6 describes the integration of a general purpose computer vision and image processing system, the design of its graphical user interface, the interactive training process and image inspection process. Chapter 7 gives the test results and the correctness measure of image classifications, and the bench mark performance (speed) measure of each operation. X-ray images containing defects in a complex casting and weld object are examined by the automated image inspection system. Various flaws in the images are successfully ex-

---

tracted from the image structural background. Finally, Chapter 8 summarizes the work, gives conclusions, and discusses future directions for this research.

## CHAPTER 2. COMPUTERIZED IMAGE INSPECTION

The automated flaw detection problem can be considered as a typical image pixel classification or image segmentation problem. To detect flaws automatically, the computer must be able to extract features (patterns) from an image, and then classify the image based on feature differences between the flaws and the object structure background in the image. Unlike many image classification applications, such as target, face, or finger-print identifications, in which global features of the image are extracted and the entire image is classified into different categories, a flaw detection algorithm classifies local areas in an image. It is basically a pixel classification application [Fu and Mui, 1981] that divides an image into several parts. The objective of flaw detection is to find which part of the image belongs to a flaw, or a certain type of flaw, and which part of the image belongs to the regular image background. Therefore, features representing flaws and background in an image have to be extracted locally in order to classify a local pixel in the image. Segmentation is achieved by examining the local features associated with each pixel to be classified and assigning each pixel to a certain class. This is done by constructing a feature vector from features extracted at every pixel location and classifying the vector in the feature space based on how well one class (such as a flaw) is separated from the other (such as the background). The classifiers, according to their func-

---

tions, fall into unsupervised and supervised categories. An unsupervised classifier, such as an iterative clustering algorithm, is an objective classifier that divides feature vectors in the feature space solely upon their distribution and/or the measure of the closeness of one feature vector to the other. A group of feature vectors close to each other is found iteratively and assigned as one class, while vectors away from that group may be assigned as other classes. No human input is given during the iteration, and the performance is usually measured by a predefined criterion function. A supervised classifier is more subjective than an unsupervised classifier. Knowledge information, usually generated from training prototypes, is given to the classifier before or during the classification process. A training process is necessary to properly establish a supervised classifier from the training data and/or human input before it can be used for the classification. Because flaw detection is a highly subjective process, supervised classifiers are used here for pixel classification. A supervised classifier is not functioning until it is properly trained by the training prototypes and controlled by the knowledge base established through the training process. Figure 2.1 shows a simplified block diagram of an automated image inspection system, in which arrows in vertical directions represent the inspecting operation of flaw detection, and arrows in horizontal directions represent the training operation. The training process is an interactive process and the inspecting process is automatic.

### **Image Local Feature Extraction**

A digital image consists of a two-dimensional array of gray tones, or pixels. The variation of pixel intensity contributes to the appearance variation that can be observed by human eyes. Image local features are quantities that are used to characterize the

---

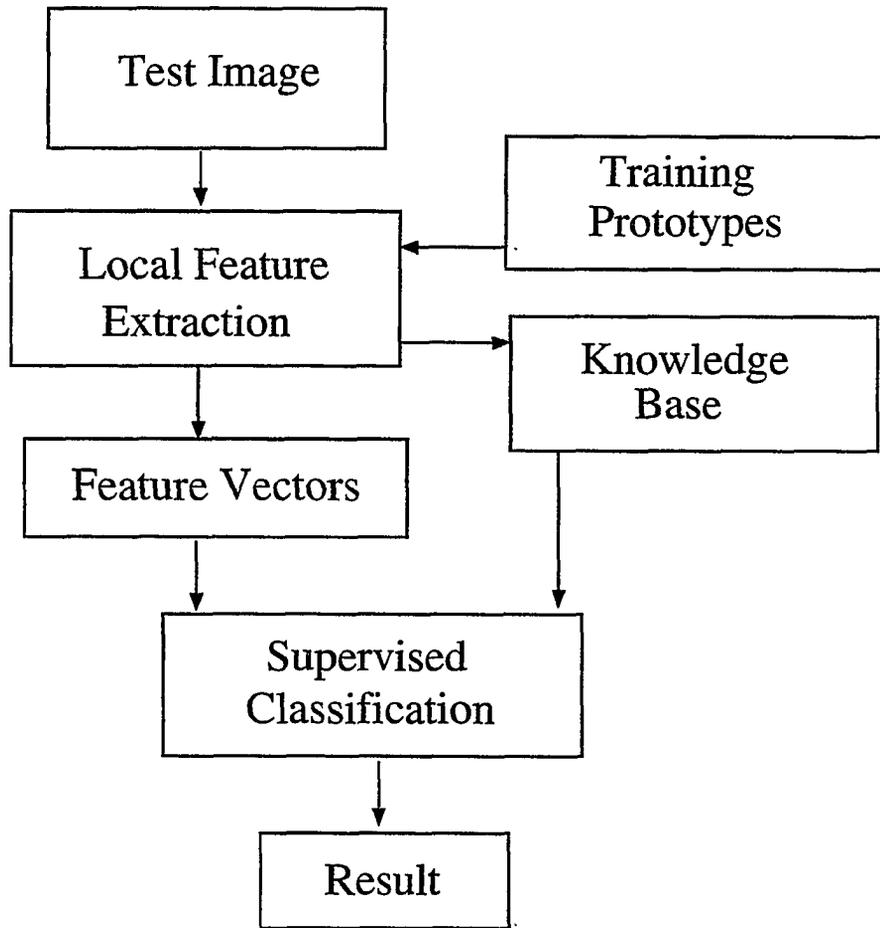


Figure 2.1: The block diagram of an automated image inspection system

appearance of an image in a certain area. For example, the pixel intensity value is a simple feature characterizing the image brightness at this pixel location. More than one feature can be used to represent a local area in the image. Features extracted from the same area in an image are combined into a vector called the feature vector and each feature becomes one element in the feature vector. It is sometimes preferable that the feature elements are orthogonal to each other so that a Euclidean space is ensured in the feature domain. The process that decompose an image into a series of images each containing one feature element is called image feature extraction.

Most of the industrial X-ray images have complicated backgrounds showing the structure of the object tested. The inspection of flaws in these kinds of images means to identify the irregular appearance among a complicated regular structure background. A successful inspection system relies on how well the system can distinguish flaw features from the background features. Extracting features in a local area usually involves a windowed process. As shown graphically in Figure 2.2, a predefined small window covering a certain area of interest is scanned on top of the image to be examined. When the center of the small window moves to any location of interest in the image, features related to that location are computed within the small window. After the completion of feature calculation at that location, the window is then moved to another location and features about the new location will be determined. This process continues for every pixel location to be classified. It is obvious that the feature extraction process directly determines the quality of image classification, because the results of the classification largely depend upon how well the feature vectors are constructed. Some commonly used feature extraction approaches are the method of moments [Hu, 1962; Alt, 1962; Smith and

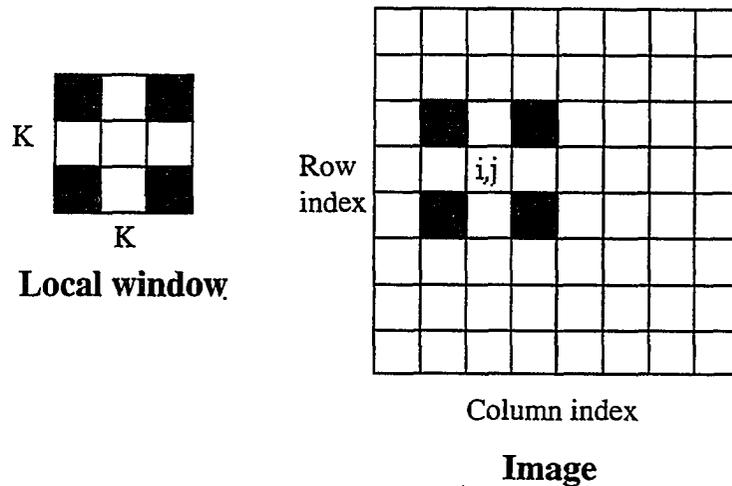


Figure 2.2: Extracting image features inside a local window

Wright, 1971; Dudani *et al.*, 1977; Reddi, 1981; Boerner and Strecker, 1988], statistical measures [Haralick *et al.*, 1973; Haralick, 1979; Pratt, 1991], texture models [Julesz, 1962; Cross and Jain, 1983], and fractals [Pentland, 1984].

The method of moments was originally used by Hu in early 1960's in an attempt to construct invariant geometric descriptors under transforms such as scaling, translation and rotations. However, the construction of higher order moments by Hu's method is tedious and its ability to represent an image by a limited number of moments is relatively poor. Other moment methods, often based on polynomial transforms, were used later to better represent an image. Typical moments are Zernike moments [Teague, 1980], Legendre moments [Teague, 1980, Reddi, 1981], rotational moments [Boyce and Hossack, 1983] and complex moments [Abu-Mostafa and Psaltis, 1984]. Among the mo-

ment methods, The Zernike moments, which are constructed from Zernike polynomials [Zernike, 1934, cited in Born and Wolf, 1980, p. 425 ] are among the best in image representation and noise robustness [Teh and Chin, 1988]. In target identification applications in which global features are needed, moment methods are generally good because of their ability to provide invariant information about the target. However, in image segmentation applications, moments are calculated locally within a small window, their ability to represent image local area properties is limited by the size of window chosen. The areas larger than the chosen window may not be properly represented by the moments.

Image characterization from its local statistical measure is another commonly used approach for feature extraction. First order and higher order statistics can be calculated within a local window to characterize image local features. This approach can effectively represent homogeneous image textures, but is sensitive to noise. It does not provide contextual information in an image and is limited by the size of the window used as well.

Image features can also be extracted based on texture models. Different texture models have been proposed to describe and discriminate the stochastic texture fields. [Julesz, 1962; Pratt *et al.*, 1978; Haralick, 1979; Cross and Jain, 1983]. These texture models, as well as fractals approach, can be successfully used in many image and texture classification applications. However, they are model based approaches and are generally limited in the certain types of images or textures they apply.

In contrast to the limitation of this or other artificial feature models, a human or biological vision system can perceive and classify image easily. The studies in

neurophysiology have revealed that the human vision system is performing a certain kind of local spatial frequency analysis through a bank of tuned bandpass filters [Campbell and Robson, 1968; Kulikowski *et al.*, 1982; Pollen and Ronner, 1982]. This fact motivates the spatial-frequency study in computer vision applications. Multichannel spatial frequency analyses of image texture have been investigated through the approach of windowed Fourier analysis and Gabor analysis [Gabor, 1946].

The biggest difficulty with the spatial frequency analysis based approaches, is the localization problem because they are directly bounded by the Heisenberg uncertainty principle [Eisberg and Resnick, 1974]. The uncertainty principle is a general law of physics originated from quantum mechanics. There are two parts of this principle, also called indeterminacy principle. The first part of the uncertainty principle states that a measurement can not simultaneously determine the exactly momentum  $p$  of a particle, and the exact value of its corresponding position  $x$ . Instead, the precision of the measurement is inherently limited by measurement itself and such that

$$\Delta p \Delta x \geq \frac{h}{4\pi} \quad (2.1)$$

where momentum  $p$  is within an uncertainty of  $\Delta p$ , the position  $x$  is within the uncertainty of  $\Delta x$ , and the  $h$  is the Planck's constant. The second part of the uncertainty principle deals with time and energy. It states that the time interval needs to be large enough to exactly measure the energy of a particle, and the uncertainty of the energy  $\Delta E$  and the the uncertainty of time  $\Delta t$  satisfy

$$\Delta E \Delta t \geq \frac{h}{4\pi} \quad (2.2)$$

Since  $\Delta E = h\Delta f$ , in time and frequency domain, the uncertainty relationship of Equation (2.2) becomes

$$\Delta f \Delta t \geq \frac{1}{4\pi} \quad (2.3)$$

which means that the measurement resolution in both the time and frequency domains can not be arbitrarily small at the same time. In another word, time frequency or spatial spatial-frequency analyses can not be simultaneously localized.

One of the popular approaches in Fourier based local spatial-frequency analyses is Gabor filter [Gabor, 1947] because it reaches the lower bound of Heisenberg uncertainty principle [Bovik *et al.*, 1990; Jain and Farrokhnia, 1991; Dunn and Higgins, 1994]. The structure of a Gabor filter is a complex sinusoid modulated by a 2-dimensional Gaussian function. It has a general form of

$$h(x, y) = g(x', y') \exp [j2\pi(Ux + Vy)] \quad (2.4)$$

where the  $x, y$  are 2-D spatial coordinates and the Gaussian modulator is

$$g(x', y') = \frac{1}{2\pi\sigma_x\sigma_y} \exp \left\{ -\frac{1}{2} \left[ \left( \frac{x'}{\sigma_x} \right)^2 + \left( \frac{y'}{\sigma_y} \right)^2 \right] \right\} \quad (2.5)$$

and its variables

$$\begin{aligned} x' &= x \cos \theta + y \sin \theta \\ y' &= -x \sin \theta + y \cos \theta \end{aligned} \quad (2.6)$$

are the coordinates rotated at an angle  $\theta$  from the  $x$  axis in the  $(x, y)$  coordinates, and  $U, V$  are the center spatial frequencies for the complex sinusoid. The spatial frequency response for the Gabor function of Equation (2.4) is given by its Fourier transform of

$$H(u, v) = \int_{-\infty}^{\infty} h(x, y) \exp [-j(ux + vy)] dx dy \quad (2.7)$$

which leads to

$$H(u, v) = \exp \left\{ -2\pi^2 \left[ \sigma_x^2 (u' - U')^2 + \sigma_y^2 (v' - V')^2 \right] \right\} \quad (2.8)$$

where

$$\begin{aligned} u' - U' &= (u - U) \cos \theta + (v - V) \sin \theta \\ v' - V' &= -(u - U) \sin \theta + (v - V) \cos \theta \end{aligned} \quad (2.9)$$

are the shifted and rotated special frequencies. The filtering process is performed by the convolution between the image function  $i(x, y)$  and the Gabor filter  $h(x, y)$  in the spatial domain as

$$f(x, y) = i(x, y) * h(x, y) \quad (2.10)$$

where  $f(x, y)$  is the filtering result. In the spatial frequency domain, it is equivalent to passing the image through a Gaussian band pass filter of Equation (2.8). By choosing various translations, rotations and scaling parameters  $\sigma_x$  and  $\sigma_y$ , one can design a set of Gabor filters that is band-pass and extract features from different spatial-frequency channels. However, the design of the filter banks from Gabor functions largely depends on individual designers. Gaussian functions are generally not local functions because it starts from minus infinite and ends at plus infinite, and the calculation over the large supporting base of a Gaussian function is computationally expensive. One way to reduce the computation load is to subsample the signal for lower frequency channels. In this case,

the filter bandwidth and the subsampling rate have to be carefully selected in order to avoid aliasing effect [Bovik *et al.*, 1990]. These and other factors make the design and the computation of the Gabor filter banks a tedious task.

The wavelet analysis is an alternative time frequency or spatial spatial-frequency analysis. The wavelet transform measures the local harmonics of a signal using sequentially scaled small waves or wavelets. It is very similar to the template matching approach with the wavelet being the template. The wavelet analysis is a constant Q analysis which decomposes a signal into multiple frequency channels with variant bandwidths. The wavelet transform obeys the uncertainty principle as well as Fourier transform. For a low frequency signal, it takes longer time to collect enough signal variations to characterize it, the transform therefore uses a longer wavelet which decompose the signal into a narrower band (or channel). For high frequency signal, the transform uses a shorter wavelet that decomposes the signal into a wider band (or channel). Within every frequency channel, the wavelets always have the same number of cycles to characterize the signal. The compactly supported wavelets are the small waves that starts and stops, therefore they are localized in the spatial (or time) domain, but their transforms are also spatially localized within each subchannel the signal is decomposed into [Daubechies, 1990]. The wavelet transform performs multichannel frequency analysis similar to what the Gabor analysis does, but it designs the frequency channels based on the constant Q concept, and performs subsampling for low frequency signal in a natural and systematic way. Therefore, it appears natural to use the wavelet transform to extract image local features.

---

## Localized Image Classification

The classification of image pixels based on the local features extracted from the image can be categorized as unsupervised and supervised classifications. The unsupervised classification (e.g., clustering) divides the feature space into several classes based on feature distributions and how close one feature is to the other in the feature space. Therefore, if two feature vectors are very close in the feature space, they are very likely to be classified as the same class. The supervised classifier classifies features based on the feedback of guiding information or previous trials. It can be considered as a "learning" process. For example, in an adaptive supervised classifier, if one previous trial is off the desired results, an error measure is usually made and the system will be adjusted to minimize the error. The system therefore learns recursively until it can reach the correct output. A typical example of supervised classifier is the backpropagating artificial neural network.

Both unsupervised and supervised classifier can give satisfactory results if the features are well separated in the feature space. A supervised classifier will be trained based on the sample training data. A good supervised classifier can distinguish features even if they are close in the feature space. In this dissertation, supervised classifiers will be used to perform pixel-wise image classification, and the classifiers are constructed based on fuzzy logic concepts. Unlike the conventional artificial neural network, which is a blind "black box" initiated from random numbers and converges slowly (if it converges), a fuzzy system is built on the fuzzy rules directly derived from the training data, or the linguistic rules from human experts. It uses fewer steps to achieve the training process because of its direct relation to the training data. An adaptive fuzzy system is

proven to be a universal approximator [Wang and Mendel, 1992] that can be used as a nonlinear mapping function similar to a neural system. The optimization of a fuzzy classifier summarizes the training data and builds the rule base in an efficient yet effective way. Fuzzy systems only sum the output fuzzy sets while neural networks sum the throughputs, therefore, a fuzzy system also avoids the crosstalk of neural system in the training process [Kosko, 1992]. In our image classification applications, a soft logic such as fuzzy logic theory is a proper choice since most of the X-ray radiographs are vague and initial human decisions are usually ambiguous.

### CHAPTER 3. WAVELET DECOMPOSITION

The wavelet transform is a method that breaks up a function or data set into different harmonic components, and analyzes them piece-by-piece at various resolutions [Daubechies, 1992]. Unlike the Fourier transform with pure harmonic bases that last forever, and the short time Fourier transform that has sudden breaks between segments of the signal, the wavelet transform is based on small waves that starts and stops, in different scales with one basic form. The advantage of the wavelet transform over the Fourier based analysis is its time-frequency localization properties. It is well known in Fourier analysis that a delta function in the time domain has constant frequency components over all frequencies, and a constant time signal only has a DC component. The reason is that the measurement resolution for time and frequency in Fourier based analysis is lower bounded by the Heisenberg uncertainty principle of Equation (2.3), which implies that accurate measure for both the time and the frequency cannot be achieved at the same time.

One of the main reasons behind the fast growing popularity of the wavelet transform in signal analysis is for its time-frequency or spatial spatial-frequency localization. Compactly supported wavelets sacrifice their symmetry and continuity in exchange for spatial localization. The wavelets (or small waves) are designed based on a constant  $Q$

---

concept [Rioul and Vetterli, 1991], which means that ratio between the bandwidth of each frequency channel and the central frequency of the channel is a constant. This is analogous to the construction of a tone burst with the high frequency base lasting for a shorter time and the low frequency base lasting for a longer time, but always keeping enough time frame to provide details of the signal. The operations of the wavelet transform are similar to those of matched filters. A wavelet transform of a signal can be viewed as a series of matched filters or templates operating on the signal, with the shape of the filters constructed by a small wave and the scale of filters changing for each process. If the local signal waveform matches the shape and the scale of the filter, or matches the harmonics of the small wave, the output is maximized, otherwise the output is near zero. The step size of each filter operation is proportional to the size of the filter. Therefore, small filters give fine resolutions and large filters give coarse resolutions, and the combination of the results for the entire series of filtering provide final output of wavelet transform.

### **Fundamentals of the Wavelet Transform**

There are various types of wavelet transforms. In this chapter, we discuss wavelet transform in one dimension. For the wavelet transform of a continuous signal, the translation and scale parameters in the transform can be continuous which leads to a continuous wavelet transform. If the translations and scales in the transform are chosen to be discrete, the transform becomes a wavelet series expansion of the continuous signal [Daubechies, 1988; Mallat, 1989]. For a discrete signal, its wavelet transform is called the discrete wavelet transform. These are analogous to Fourier based transforms such as

---

the continuous Fourier transform, the Fourier series expansion, and the discrete Fourier transform [Rioul and Vetterli, 1991].

### Continuous wavelet transform

The general form of the continuous wavelet transform for a time signal  $f(t)$  is given as [Daubechies, 1992]

$$F(\tau, a) = \frac{1}{\sqrt{|a|}} \int_{-\infty}^{\infty} f(t) W^* \left( \frac{t-\tau}{a} \right) dt \quad (3.1)$$

where  $W(t)$  is the wavelet,  $a$  is the scale,  $\tau$  is the translation and the "\*" indicates the complex conjugate. Equation (3.1) would be a standard convolution between the signal and the "small wave" if the scale is a constant value instead of a variable.

If the wavelets have zero average over the time, the inverse transform for the wavelet transform of Equation (3.1) can be calculated by the integral over the translation and scale plane, which is given by

$$f(t) = c \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \frac{1}{|a|^{5/2}} F(a, \tau) W \left( \frac{t-\tau}{a} \right) da d\tau \quad (3.2)$$

where  $c$  is a constant that depends only on the wavelet function  $W(t)$ .

### Wavelet series expansion

The wavelet series expansion decomposes a continuous signal into discrete wavelet bases. If we use a real wavelet function  $W(t)$  and choose the scale  $a = 2^{-m}$  and translation  $\tau = n/2^m$ , which are discrete values depending on  $m$  and  $n$ , the wavelet transform of Equation (3.1) leads to a series of discrete parameters

$$F_{mn} = 2^{m/2} \int_{-\infty}^{\infty} f(t) W(2^m t - n) dt \quad (3.3)$$

in which the transform parameters  $F_{mn}$  are discrete values depending on discrete integers  $m$  and  $n$ .

Like all the other orthogonal expansion series, if the wavelet function in Equation (3.3) satisfies its orthonormal condition

$$\int_{-\infty}^{\infty} 2^{m/2} 2^{m'/2} W(2^m t - n) W(2^{m'} t - n') dt = \delta_{mm'} \delta_{nn'} \quad (3.4)$$

the function  $f(t)$  in Equation (3.3) can be expanded in terms of the wavelet bases and the the transform parameters as

$$f(t) = \sum_m \sum_n 2^{m/2} F_{mn} W(2^m t - n) \quad (3.5)$$

where  $F_{mn}$  represents the transform parameters in and  $W(t)$  represents the wavelets.

The wavelets  $W$  can be constructed from certain scaling function  $\Phi$  that satisfies a dilation equation [Daubechies, 1992]

$$\Phi(t) = \sum_{k=0}^{K-1} c_k \Phi(2t - k), \quad (3.6)$$

provided that  $\Phi$  satisfies the orthonormal condition

$$\int_{-\infty}^{\infty} \Phi(t - k) \Phi(t - k') dt = \delta_{kk'}. \quad (3.7)$$

The constant coefficients  $\{c_k: k = 0, 1, 2, \dots, K-1\}$  ( $K$  is an even integer) in Equation (3.6) are the "magic" numbers for the wavelet to be constructed. The wavelet function can be derived explicitly from the scaling function  $\Phi$  through another dilation equation

$$W(t) = \sum_{k=0}^{K-1} (-1)^k c_{K-1-k} \Phi(2t - k) \quad (3.8)$$

where the coefficients  $\{c_k\}$  are the same as those in Equation (3.6) but are placed in an inverse order.

With the introduction of the scaling functions, the wavelet expansion of Equation (3.5) can also be written in a split form which contains both scaling functions and the wavelets [Crandall, 1994]. This is given by the following equation:

$$f(t) = \sum_k s_k \Phi(t - k) + \sum_{m \geq 0} \sum_n d_{mn} W(2^m t - n) \quad (3.9)$$

where

$$s_k = \int_{-\infty}^{\infty} f(t) \Phi(t - k) dt \quad (3.10)$$

represents the signal component smoothed by a scaling function and

$$d_{mn} = 2^{m/2} F_{mn}; \quad m \geq 0 \quad (3.11)$$

represents the signal decomposed by a wavelet. Equation (3.9) shows that the smoothed and decomposed components in discrete-basis wavelet expansion fully represent the original signal.

## Discrete wavelet transform

The discrete wavelet transform is the discrete-basis wavelet transform for discrete signals. Pyramidal coding or multiresolution signal analysis was developed in early eighties [Burt and Adelson, 1983] which built the foundation of the discrete wavelet transform.

The discrete wavelet transform involves scaling and decomposition with sequentially changed scales. Large scales represent global views of a signal and the smaller scales represent local details. Usually, the length of the signal is made to be an integral power of 2 for fast computation. The bridges between the discrete wavelet transform and the continuous transforms lie in the dilation equation of Equations (3.6) and (3.8). For example, given the discrete points  $W(1)$  and  $W(2)$  of a wavelet, half integer points such as  $W(1/2)$ ,  $W(3/2)$ ,  $W(5/2)$ ,  $W(7/2)$  can be computed from the dilation equation. With the half integer points given, quarter integer values can be computed in a similar way. Eventually, a continuous wavelet can be computed numerically.

In digital image processing applications, only the discrete wavelet transform is used to process the image. The pyramid structure for the discrete transform leads to a fast algorithm to compute the transform. In the following section, the discrete wavelet transform will be discussed in detail through special cases such as Haar transform [Haar, 1910] and the compactly supported wavelet transform [Daubechies, 1988].

### Haar Transform

The Haar wavelets are the simplest wavelets. They are square waves that only have up and down sides at 1 and -1. The Haar transform of a discrete signal compute the

---

average and the difference of the neighboring signal values for various segments of the signal. The scale for the wavelet and the smoothing function sequentially changes by a factor of two at each level of operation.

### Decomposition to orthogonal basis

It is well known that any signal can be decomposed into a set of orthogonal basis vectors, while the signal is uniquely represented by its decomposed components at each base. The Cartesian coordinate system is the most commonly used orthogonal basis in which unit vectors  $(1, 0, 0, \dots)$ ,  $(0, 1, 0, \dots)$ , etc., are used as the basis vectors. In a two dimensional Cartesian space, any vector  $(x, y)$  in the space is represented in terms of two basis vectors,  $(1, 0)$  and  $(0, 1)$ , as shown in Figure 3.1.

The orthogonal basis for signal analysis is, however, not unique. As shown in Figure 3.2, if the coordinates of Figure 3.1 rotate 45 degrees clockwise, a new basis is

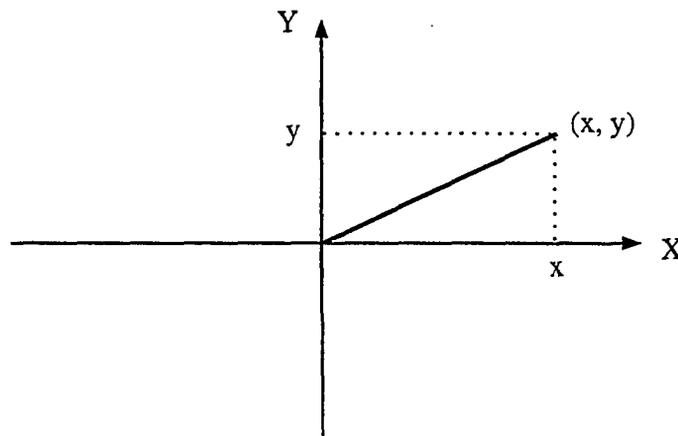


Figure 3.1: A vector is represented by its orthogonal basis

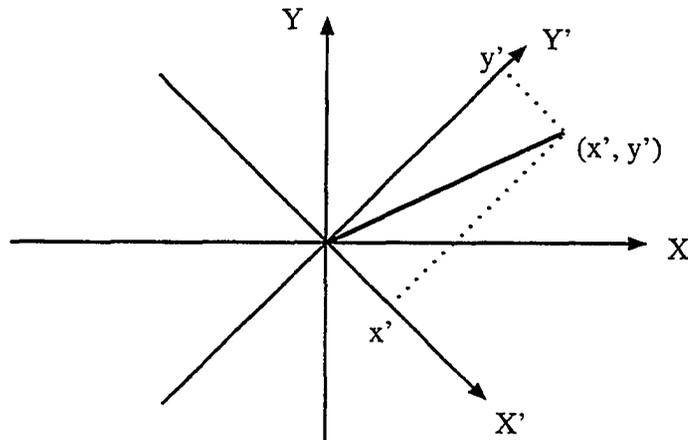


Figure 3.2: A vector is represented by another orthogonal basis

formed and the unit vectors in the new basis are parallel to vectors  $(1, 1)$  and  $(1, -1)$ , respectively. For the new coordinate values  $(x', y')$  in the rotated basis,  $y'$  is related to the average of the original components  $x$  and  $y$ , and  $x'$  is related to the difference between  $x$  and  $y$  (with a factor of square root of 2). The transform from the original basis to the new average and difference basis is called Haar transform. It is named after German mathematician Alfred Haar [Haar, 1910] who originally constructed the orthogonal basis. In a higher dimensional space, an orthogonal Haar basis is completed by a series of dilations and translations about vectors  $(1, 1)$  and  $(1, -1)$ . A regular Haar transform matrix of size 4 is given as [Strang, 1994]

$$\left[ \tilde{\mathbf{H}} \right]_{4 \times 4} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & 0 & 0 \\ 0 & 0 & 1 & -1 \end{bmatrix} \quad (3.12)$$

in which the first 2 vectors,  $(1, 1, 1, 1)$  and  $(1, 1, -1, -1)$  are dilated from the base vectors  $(1, 1)$  and  $(1, -1)$ . The third and fourth vectors are constructed by padding zeros after  $(1, -1)$ , and the fourth vector is constructed by shifting  $(1, -1)$  to the beginning and the end of the vectors. The four vectors in the Haar matrix are apparently perpendicular to each other and therefore, provide an orthogonal basis for signal decomposition. The first line in Equation (3.12) is a scaling vector, while the remaining three vectors are Haar wavelets. Higher order Haar matrices can be constructed in a similar manner. It is apparent that Haar wavelets are rectangular waves. The Haar transform for a vector of size  $N$  can be calculated by the multiplication between an  $N$  by  $N$  Haar matrix and the vector. Similar to the 4 by 4 Haar matrix of Equation (3.12), an  $N$  by  $N$  matrix can as well be formed from scaled and shifted vectors of  $(1, 1)$  and  $(1, -1)$ .

### The pyramid algorithm

The multiplication between an  $N$  by  $N$  Haar matrix and a size  $N$  vector results in the Haar transform of the vector. Such a multiplication takes  $N^2$  multiplication and accumulation steps. A fast algorithm, which is analogous to the FFT in Fourier analysis, computes the transform in a multiresolution manner and therefore forms a pyramid structure. A 4 by 4 transform matrix for the pyramid algorithm is written as [Strang, 1994]

$$\begin{bmatrix} \mathbf{H} \end{bmatrix}_{4 \times 4} = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & -1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & -1 \end{bmatrix} \quad (3.13)$$



As discussed previously, the multiplication between an  $N$  by  $N$  Haar transform matrix and a size  $N$  vector requires  $N^2$  multiplications and  $N^2$  accumulations. The pyramid algorithm introduces a fast approach to calculate the Haar transform by reducing the total number of operations, provided that  $N$  is an integral power of 2.

For  $N = 2^m$ , the pyramid algorithm takes totally  $m$  steps. Therefore the total number of smoothing operations (denoted as  $NOS$ ), which is equal to the total number of differentiations (denoted as  $NOD$ ), is given by

$$\begin{aligned}
 NOS &= \frac{N}{2} + \frac{N}{2^2} + \cdots + \frac{N}{2^{m-1}} + 1 \\
 &= 2^{m-1} + 2^{m-2} + \cdots + 2 + 1 \\
 &= 2^m - 1 \\
 &= N - 1.
 \end{aligned} \tag{3.14}$$

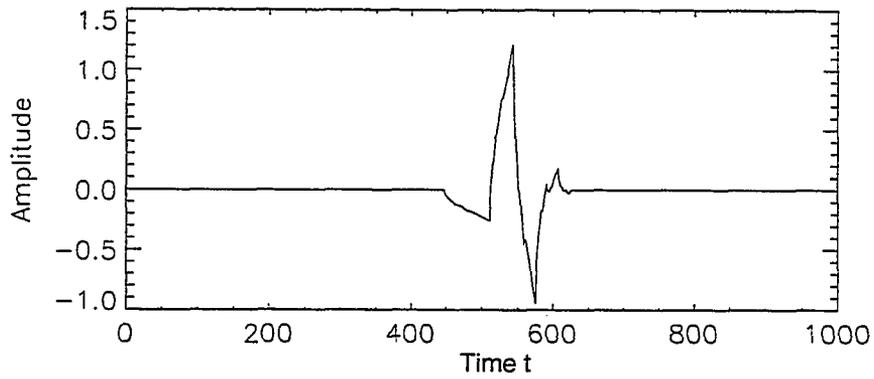
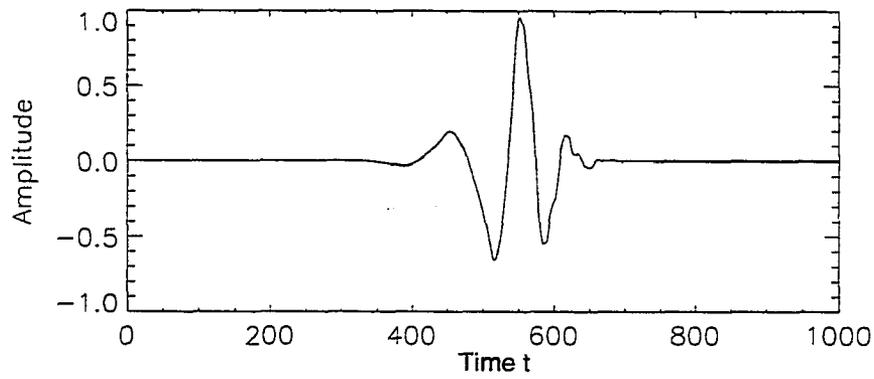
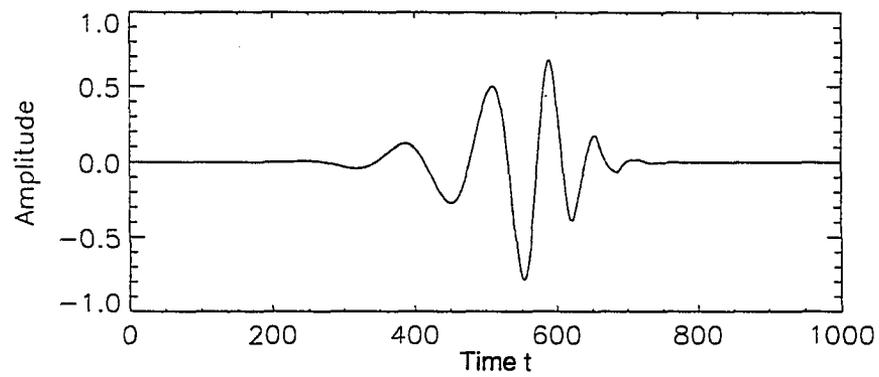
The result of the above calculation indicates that there are totally  $N-1$  smoothing and  $N-1$  differentiating operations for the pyramid Haar operation. Since each smoothing or differentiating operation involves two multiplications and one accumulation (addition or subtraction), for the pyramid Haar operation of a size  $N$  vector, only  $4(N-1)$  multiplications and  $2(N-1)$  accumulations are needed. Comparing to  $N^2$  multiplications and  $N^2$  accumulations for the normal matrix operation, it is a reduction in an order of  $N$ .

### Compactly Supported Wavelet Transform

Haar transform has finite spatial (time) support, but it is made of squares and therefore is not good in characterizing frequencies. To effectively analyze image local harmonics requires a transform that is localized in both spatial and spatial frequency domains. This seemed unlikely because of the Heisenberg uncertainty principle stated the

law of physics for all time-frequency based analyses. In late 1980's, Daubechies discovered a class of orthogonal wavelet bases that have time and frequency localization property [Daubechies, 1988], the transform against the orthogonal basis is referred as compactly supported wavelet transform. Daubechies wavelets are toned pulses that start and stop and are orthogonal to all their dilations and translates. Unlike sine and cosine bases in Fourier analysis that last for ever, these wavelet bases are localized in time. However, it is also localized in the transformed domain in a constant Q fashion. Within each sub-band of wavelet transform, the transform is localized for each small wave [Daubechies, 1992; Rioul and Vetterli, 1991]. Wavelet transform is actually more of a template matching approach than a direct time-frequency analyzer. By using small waves (wavelets) representing local harmonics as the templates, it provides an alternative way to represent local time-frequency relationship. This is probably the most attractive property for the wavelet transform to be used in signal processing applications because most of the real world signals do not fit the stationary assumption and the wavelet transform provides a tool to break a signal into pieces without losing contextual information. There are many compactly supported wavelets. A particular set of wavelets is specified by a particular set of "magic" numbers, which are usually called wavelet filter coefficients [Strang, 1994]. The most compactly supported wavelets discovered by Daubechies has four filter coefficients, therefore they are sometimes referred as DAUB4 wavelets [Press *et al.*, 1992]. The DAUB4 wavelets are the least continuous wavelets. Some other compactly supported wavelets Daubechies discovered have 6, 8, 10, 12, 14, 16, 18, and 20 filter coefficients. With the increased number of filter coefficients, the wavelets become more and more smooth, but less and less localized. Figure 3.4 (a), (b), and (c) plot three

---

(a) DAUB4 wavelet  $W(t)$ (b) DAUB8 wavelet  $W(t)$ (c) DAUB16 wavelet  $W(t)$ Figure 3.4: The compactly supported Daubechies wavelets  $W(t)$

Daubechies wavelet functions  $W(t)$  [Press *et al.*, 1992] with 4, 8, and 16 filter coefficients, respectively. Only the most compactly supported wavelet transform (DAUB4) will be used in the image feature extraction process.

### Transform matrix and parameters

Similar to that of the Haar transform, the discrete wavelet transform of a one-dimensional vector  $\mathbf{X}$  can be computed by the pyramid algorithm. For the most compact wavelets DAUB4, there are only four wavelet filter parameters in the transform matrix. These four "magic" numbers ( $c_i$ 's in Equation (3.15)) are arranged into a semi-diagonal transform matrix similar to that of a Haar pyramid transform displayed in Equation (3.13) for a multiresolution operation. Equation (3.15) gives an example of a pyramid transform matrix with a size of 8 [Press *et al.*, 1992]

$$[\mathbf{C}]_{8 \times 8} = \begin{bmatrix} c_0 & c_1 & c_2 & c_3 & 0 & 0 & 0 & 0 \\ c_3 & -c_2 & c_1 & -c_0 & 0 & 0 & 0 & 0 \\ 0 & 0 & c_0 & c_1 & c_2 & c_3 & 0 & 0 \\ 0 & 0 & c_3 & -c_2 & c_1 & -c_0 & 0 & 0 \\ 0 & 0 & 0 & 0 & c_0 & c_1 & c_2 & c_3 \\ 0 & 0 & 0 & 0 & c_3 & -c_2 & c_1 & -c_0 \\ c_2 & c_3 & 0 & 0 & 0 & 0 & c_0 & c_1 \\ c_1 & -c_0 & 0 & 0 & 0 & 0 & c_3 & -c_2 \end{bmatrix}. \quad (3.15)$$

There are actually two filters in the transform matrix which are shifted by two columns with each successive pair of rows. These two filters are sometimes called quadrature mirror filters [Press *et al.*, 1992]. The first filter (the first row in Equation

(3.15)) "smoothes" the signal and provides an local average. The second filter (the second row of the matrix) "decomposes" the signal and provides local details. The two mirror filters are orthogonal to each other since their dot product equals to zero.

The transform matrices of other sizes are constructed in a similar manner by the filter pairs. When constructing a transform matrix of size  $N$  ( $N$  is an integral power of 2), the mirror filter pair (The first 4 columns of row 1 and row 2 in Equation (3.15)) starts at the beginning of the first two rows of the matrix (zeros are padded after the first 4 elements in each row), then the pair shifts to the right by two columns (the elements vacated after the shift are filled with zero) for the next two rows (row 3 and row 4) in the transform matrix. The two-column shift continues for the next every two rows (row 5 and row 6, row 7 and row 8, etc.) and the transform matrix is formed into a semi-diagonal form. Like the 8 by 8 matrix in Equation (3.15), the last row of the matrix only has non zero elements at the beginning and the end of the row. This is because the last two elements of the mirror filters wrap around to the beginning of the row. The parameters of the mirror filters are chosen such that constant and steadily increasing signals have zero output after a decomposition. For example, in the DAUB4 wavelet transform, parameters  $c_0, c_1, c_2, c_3$  must be chosen so that the dot product between the decomposition filter and the vectors  $(1, 1, 1, 1), (0, 1, 2, 3)$  are zero [Strang, 1994], which leads to the following equations:

$$c_3 - c_2 + c_1 - c_0 = 0 \quad (3.16)$$

$$0c_3 - 1c_2 + 2c_1 - 3c_0 = 0. \quad (3.17)$$

Two additional equations are needed in order to determine exclusively the four parameter in the transform matrix. One of the equations is obtained from the orthogonal condition between rows 2 and 4 in the transform matrix of Equation (3.15), which leads to

$$c_1 c_3 + c_0 c_2 = 0. \quad (3.18)$$

The other is a normalization condition which sets the magnitudes of the four parameters, and makes sure the matrix for the inverse transform is the inverse matrix of that of the forward transform, given as

$$c_0^2 + c_1^2 + c_2^2 + c_3^2 = 1. \quad (3.19)$$

The four equations (3.16), (3.17), (3.18) and (3.19) determine the four parameters in the DAUB4 wavelet transform matrix exclusively. The values of the parameters are given in the following:

$$\begin{aligned} c_0 &= \frac{1+\sqrt{3}}{4\sqrt{2}} = 0.482962913145, \\ c_1 &= \frac{3+\sqrt{3}}{4\sqrt{2}} = 0.836516303739, \\ c_2 &= \frac{3-\sqrt{3}}{4\sqrt{2}} = 0.224143868042, \\ c_3 &= \frac{1-\sqrt{3}}{4\sqrt{2}} = -0.129409522551. \end{aligned} \quad (3.20)$$

Similar to the Haar decomposition, the wavelet transform decomposes a signal into a scaling function and a series of dilated wavelets at different locations. These functions form the orthogonal basis of the transform. Given the filter parameters, the orthogonal basis functions for the Daubechies wavelet decomposition are given by the dilation equations in which the filter parameters are used as the coefficients. The dilation equation also provide an iterative way to numerically calculate continuous wavelets and scaling functions from the discrete results. The dilation equations for the scaling function  $\Phi(t)$  and the wavelet function  $W(t)$  are [Daubechies, 1992]

$$\Phi(t) = c_0\Phi(2t) + c_1\Phi(2t-1) + c_2\Phi(2t-2) + c_3\Phi(2t-3) \quad (3.21)$$

$$W(t) = c_3\Phi(2t) - c_2\Phi(2t-1) + c_1\Phi(2t-2) - c_0\Phi(2t-3). \quad (3.22)$$

Figures 3.5 and 3.6 display a scaling function and a wavelet function, respectively, for the DAUB4 wavelet transform. A wavelet transform actually decomposes a signal into the orthogonal basis of a scaling function and a set of wavelet functions with various scales and shifts.

To calculate the parameters of wavelets with various size, similar approaches are used. Daubechies has calculated the parameters for compactly supported wavelet transforms ranging from filter sizes of 4 to 20 [Daubechies, 1992], which are listed below in Tables 3.1 through 3.4.

### **Wavelet transform in a pyramid structure**

Similar to the pyramid Haar transform discussed previously, the Daubechies wavelet transform can also be computed through a pyramid structure. As sketched

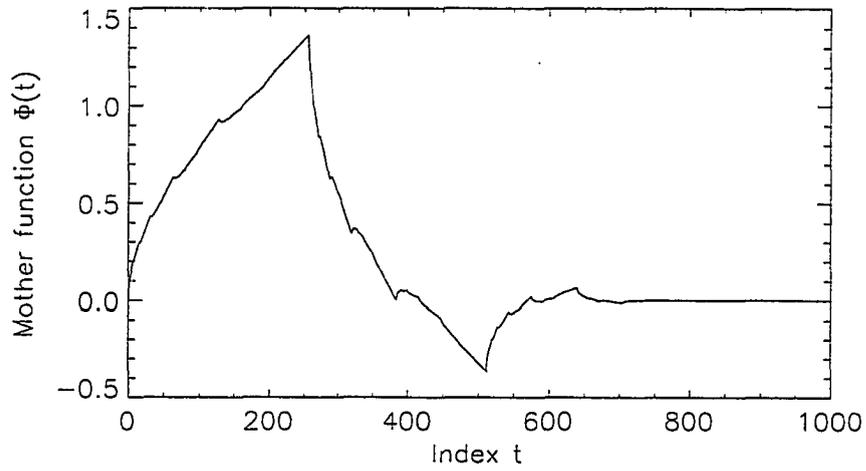


Figure 3.5: The scaling function,  $\Phi(t)$ , of DAUB4 wavelet transform

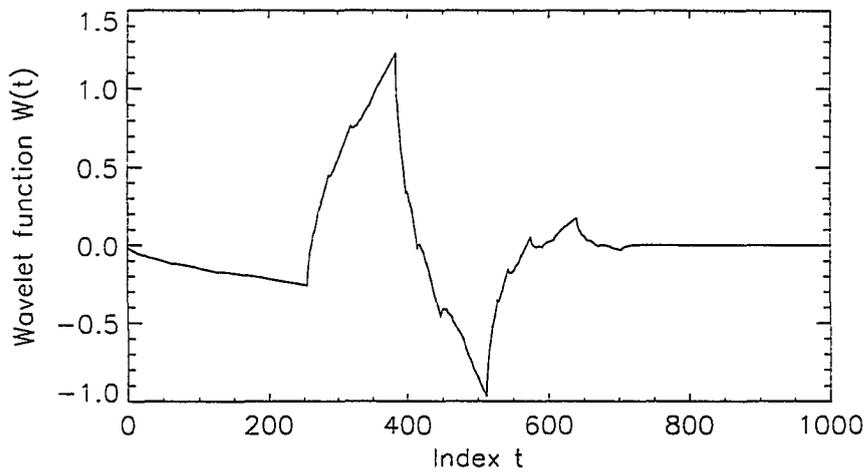


Figure 3.6: The wavelet function,  $W(t)$ , of DAUB4 wavelet transform

Table 3.1: The filter parameters for Daubechies wavelet transform (part 1)

n	Filter parameter $c_n$
Size = 4	
0	0.482962913145
1	0.836516303739
2	0.224143868042
3	-0.129409522551
Size = 6	
0	0.332670552950
1	0.806891509311
-2	0.459877502118
3	-0.135011020010
4	-0.085441273882
5	0.035226291886
Size = 8	
0	0.230377813309
1	0.714846570553
2	0.630880767940
3	-0.027983769417
4	-0.187034811719
5	0.030841381836
6	0.032883011667
7	-0.010597401785
Size = 10	
0	0.160102397974
1	0.603829269797
2	0.724308528438
3	0.138428145901
4	-0.242294887066
5	-0.032244869585
6	0.077571493840
7	-0.006241490213
8	-0.012580751999
9	0.003335725286

Table 3.2: The filter parameters for Daubechies wavelet transform (part 2)

n	Filter parameter $c_n$
Size = 12	
0	0.111540743350
1	0.494623890398
2	0.751133908021
3	0.315250351709
4	-0.226264693965
5	-0.129766867567
6	0.097501605587
7	0.027522865530
8	-0.031582039318
9	0.000553842201
10	0.004777257511
11	-0.001077301085
Size = 14	
0	0.077852054085
1	0.396539319482
2	0.729132090846
3	0.469782287405
4	-0.143906003929
5	-0.224036184994
6	0.071309219267
7	0.080612609151
8	-0.038029936935
9	-0.016574541631
10	0.012550998556
11	0.000429577973
12	-0.001801640704
13	0.000353713800

Table 3.3: The filter parameters for Daubechies wavelet transform (part 3)

n	Filter parameter $c_n$
Size = 16	
0	0.054415842243
1	0.312871590914
2	0.675630736297
3	0.585354683654
4	-0.015829105256
5	-0.284015542962
6	0.000472484574
7	0.128747426621
8	-0.017369301002
9	-0.044088253931
10	0.013981027917
11	0.008746094047
12	-0.004870352994
13	-0.000391740373
14	0.0006754494065
15	-0.000117476784
Size = 18	
0	0.038077947364
1	0.243834674613
2	0.604823123690
3	0.657288078051
4	0.133197385825
5	-0.293273783279
6	-0.096840783223
7	0.148549749338
8	0.030725681479
9	-0.067632829061
10	0.000250947115
11	0.022361662124
12	-0.004723204758
13	-0.004281503683
14	0.001847646883
15	0.000230385764
16	-0.000251963189
17	0.000039347320

Table 3.4: The filter parameters for Daubechies wavelet transform (part 4)

n	Filter parameter $c_n$
Size = 20	
0	0.026670057901
1	0.188176800078
2	0.527201188932
3	0.688459039454
4	0.281172343661
5	-0.249846424327
6	-0.195946274377
7	0.127369340336
8	0.093057364604
9	-0.071394147166
10	-0.029457536822
11	0.033212674059
12	0.003606553567
13	-0.010733175483
14	0.001395351747
15	0.001992405295
16	-0.000685856695
17	-0.000116466855
18	0.000093588670
19	-0.000013264203

schematically in Figure 3.7, a DAUB4 wavelet transform for a size 8 vector  $\mathbf{X} = (x_0 \ x_1 \ x_2 \ x_3 \ x_4 \ x_5 \ x_6 \ x_7)$  is obtained in a step by step manner. In the first step, the multiplication between the vector and the 8 by 8 transform matrix of Equation (3.15) leads to another vector which contains the smoothed and decomposed results of every four neighboring elements in  $\mathbf{X}$  obtained by shifting the quadrature mirror filters two element at a time. In the next step, a similar operation is applied to the values already smoothed in the first step, which leads to a second level of smoothing and decomposition. The second level of smoothed and decomposed components, combined with the first level decompositions are reordered and constructed into a vector of same size as the original to give the final result of the wavelet transform. The wavelet transform provides a natural way to decompose a signal into orthogonal basis in a multiresolution fashion. The pyramid algorithm for DAUB4 wavelet transform for a size  $N$  vector also has  $N-1$  smoothing and  $N-1$  differentiating operations. Since there are four parameters in each mirror filter, the

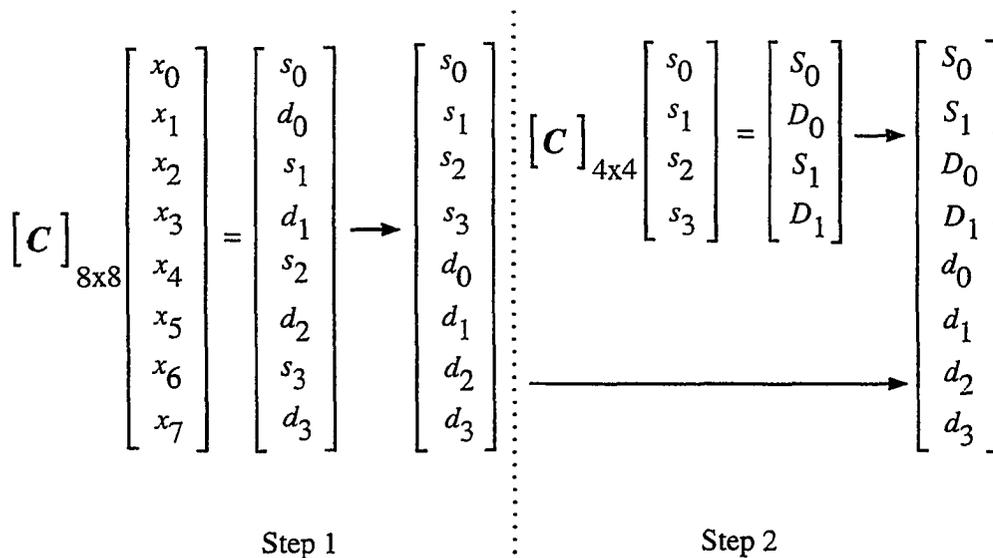


Figure 3.7: The pyramid computation of DAUB4 wavelet transform

algorithm results in  $8(N-1)$  multiplications and  $6(N-1)$  accumulations. Considering the  $N^2$  multiplications and  $N^2$  accumulations required in the regular matrix transform, the pyramid algorithm gives significant reduction in computation load, especially when  $N$  is large.

Figure 3.8 shows a simulated signal consisting of DAUB4 wavelets of different scales as a function of time. The wavelet transform of this signal is calculated by the pyramid algorithm. In the first step of the decomposition, the signal is multiplied by a 1024 by 1024 ( $N = 1024$ ) transform matrix (Extended form of Equation (3.15)) and is decomposed into 512 smoothed and 512 differentiated values following the procedure shown in Figure 3.7. The process then moves to the second level, a 512 by 512 transform matrix is applied on the first level smoothed values and decomposes them into 256 smoothed and 256 differentiated values. This process continues at higher level of decompositions until the smoothed signal is no longer decomposable (when there are less than four smoothed values left). The result of the wavelet transform is a size 1024 vector that

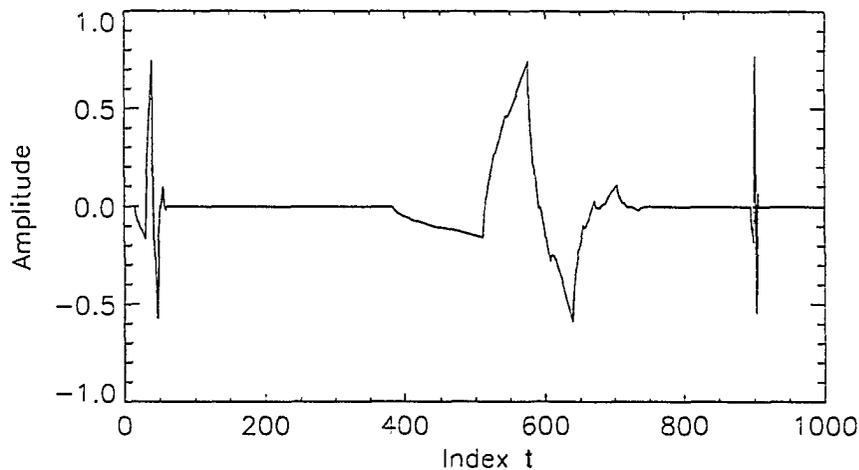


Figure 3.8: A one dimensional signal with differently scaled wavelets

is constructed from the signal components decomposed from all levels of decompositions. Figure 3.9 displays the first 600 data points (the rest of the data are all zero) of the wavelet transform, in which the horizontal axis consists sequentially changed scales and only delta functions are observed at the appropriate scale level. These delta functions appear in the transform because the simulated signal in Figure 3.8 only contains wavelets. Since the wavelet transform is similar to matched filtering, when the signal component matches a certain wavelet, it gives a delta function response in the transform domain. Note that the original signal contains wavelets of equal magnitude, but in its transform of Figure 3.9, the lower level decomposed signal has a smaller magnitude. This is due to the fact that the regular wavelet transform is normalized by its energy instead of its amplitude [Telfer and Szu, 1992]. Therefore, a wavelet transform is biased in representing frequencies. Specifically, if a signal contains two frequency components with the same amplitude strength, the wavelet transform magnitude for the low frequency component is higher than that of high frequency components.

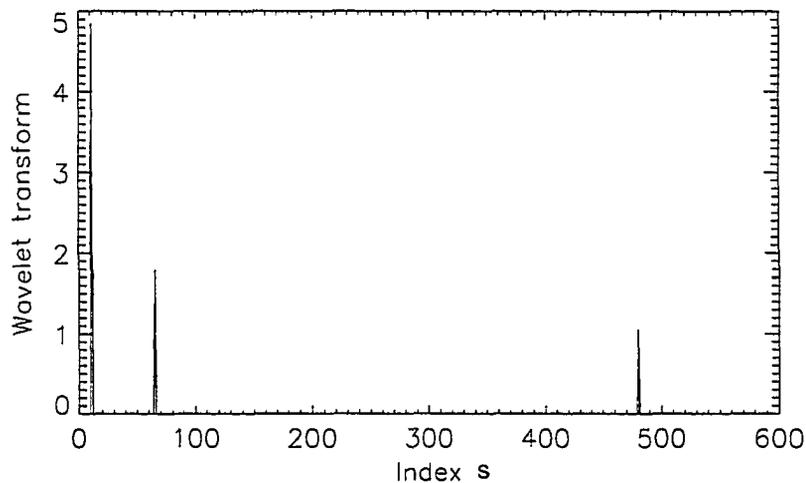


Figure 3.9: The regular wavelet transform of the signal in Figure 3.8

### Amplitude normalization

The constant energy normalized wavelet transform is biased against frequency. An unbiased normalization for all frequencies is needed if an application involves frequency comparison over different subband channels. For the discrete transform, a set of correction factors can be found numerically by comparing the magnitude of wavelet transforms for the wavelets with successively changed scales. Table 3.5 shows the normalization factors computed for various Daubechies wavelets at every step of decomposition. For the DAUB4 wavelets, only the first column in Table 3.5 is used. At each step of decomposition, the amplitude normalization is done by multiplying the differentiating filter in the transform matrix (the second filter in the quadrature mirror filter pair) by the corresponding correction factor. The amplitude normalized wavelet transform becomes unbiased for all frequency channels.

Table 3.5: The normalization factors for Daubechies wavelet decomposition

Step	Wavelet filter size								
	4	6	8	10	12	14	16	18	20
1	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
2	1.063	1.150	1.387	1.281	1.126	1.244	1.434	1.318	1.326
3	1.338	1.434	1.700	1.734	1.606	1.596	1.747	1.873	1.909
4	1.793	1.910	2.385	2.427	2.273	2.207	2.467	2.620	2.606
5	2.471	2.624	3.342	3.415	3.204	3.115	3.452	3.689	3.683
6	3.450	3.659	4.719	4.819	4.533	4.402	4.876	5.215	5.201
7	4.849	5.133	6.663	6.817	6.408	6.224	6.893	7.373	7.353
8	6.836	7.233	9.423	9.638	9.062	8.803	9.748	10.425	10.398
9	9.652	10.213	13.327	13.587	12.687	12.321	13.780	14.789	14.160
10	13.640	14.823	17.447	17.260	16.369	16.671	17.060	16.240	15.794

Amplitude correction should also be made on the smoothing filters in order to preserve the same signal level after each smoothing operation. For the continuous transform, it is suggested that the square root of the scale should be multiplied for amplitude normalization [Telfer and Szu, 1992]. For the discrete transform the scale is  $1/2$ . Therefore, if a wavelet transform has totally  $N$  level of smoothing and decomposition operations, a factor of  $2^{N/2}$  should be divided into the smoothed portion of the signal to ensure the smoothed signal has the same amplitude as the original signal.

For the signal displayed in Figure 3.8, its amplitude normalized wavelet transform is displayed in Figure 3.10. The transformed values for the three differently scaled (in different frequency channel) components are equal in normalized transform. This means that normalized wavelet transform provides an unbiased representation of all frequency components. The process to extract image features from its normalized wavelet decompositions is discussed in the following Chapter 4.

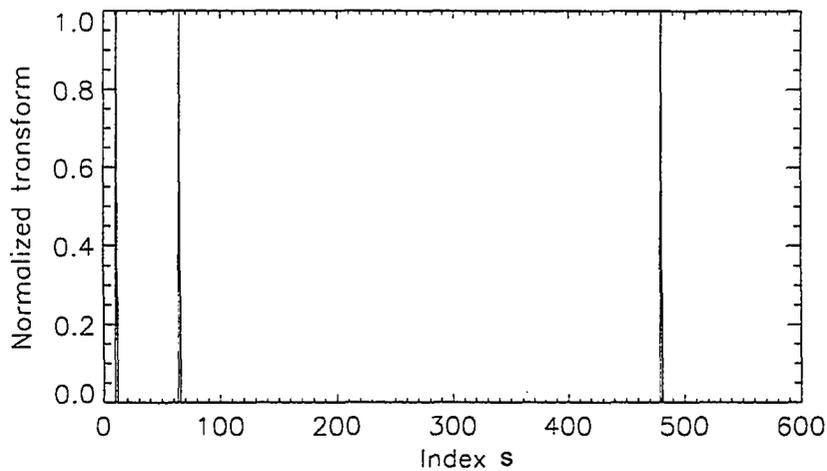


Figure 3.10: The amplitude normalized wavelet transform of the signal in Figure 3.8

## **CHAPTER 4. WAVELET BASED LOCAL FEATURE EXTRACTION**

The local features in an image are extracted from the compactly supported wavelet decompositions of the image. Since an image is a two-dimensional array, the one dimensional wavelet transform discussed in Chapter 3 has to be extended to two dimension to process a 2-D image.

### **Two Dimensional Wavelet Transform**

Like the Fourier transform, the wavelet transform can also be applied to a two dimensional signal. A standard approach for decomposing a two dimensional array (or image) into a set of 2-D orthogonal bases was developed by Mallat [Mallat, 1989] and has been widely known as the two-dimensional wavelet transform. The Mallat 2-D wavelet decomposition is similar to the Daubechies decomposition in one dimension. In contrast to 1-D pyramid decomposition, which breaks a signal into two parts in each step, each step in Mallat's algorithm "smoothes" and "decomposes" a 2-D array into 4 parts: One is smoothed in both horizontal and vertical directions, another is decomposed in both directions, and the other two parts are smoothed at one direction and decomposed at the other direction, while the next level of operation applies only to the part smoothed at both directions.

The Mallat decomposition is a multiresolution decomposition for a two dimensional image. In the spatial frequency domain, Mallat's decomposition provides DC components (by smoothing), a full range of spatial frequency components at horizontal or vertical directions (by decomposition in the horizontal or vertical direction), and a full range of spatial frequency components at a 45 degree angle from the horizontal (or the vertical) direction (by decomposition in diagonal directions). In image feature extraction, the local spatial frequency components at all angles are equally important. Therefore, a two dimensional decomposition should be able to provide a full range of the information in spatial frequency domain for spatial frequency analysis. The two-dimensional extension of the 1-D Daubechies wavelet transform decomposes a 2-D signal into a set of subchannels that cover the entire spatial frequency range, and therefore is more suitable for our application. The two-dimensional Daubechies wavelet transform of an image is executed by applying the one-dimensional transform on every row of the image, and then the one-dimensional transform on every column of the row transformed image [Press *et al.*, 1992].

To extract wavelet features from an image, a two-dimensional DAUB4 discrete wavelet transform is performed on the image for the best spatial localization. The transform is performed to a certain level of decomposition in the pyramid structure. The total level of decomposition is determined on a trial-by-trial basis that varies for different images and flaws. The reason is that there are various object structures displayed in different kind of images. Features for smaller objects may be extracted by the initial a few steps of decompositions, but features for bigger objects need more steps to unveil the entire structure. The total number of decomposition is determined by the operator in the

machine training stage based on the nature of the image, the appearance of the features, and the size of the flaws to be detected. Because the pyramid algorithm requires both dimensions of the image to be an integral power of 2, an image smaller than that will be expanded to its nearest limit and zeros are padded in the extra space.

Figure 4.1 displays a 256 by 256 test image containing a round plate and several circular holes with various depths. The first level and second level wavelet decompositions of this image are shown in Figures 4.2 and 4.3, respectively, in which image intensities in transformed images are displayed in a logarithm scale. Within each spatial frequency channel, the decomposed signal is displayed through scaled maps which display smaller or distorted images. The decomposed components for all subband-channels are displayed within a unified image frame.

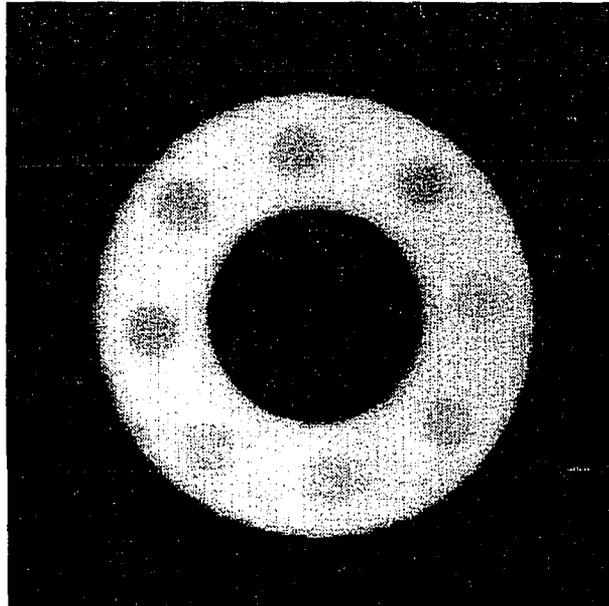


Figure 4.1: A test image

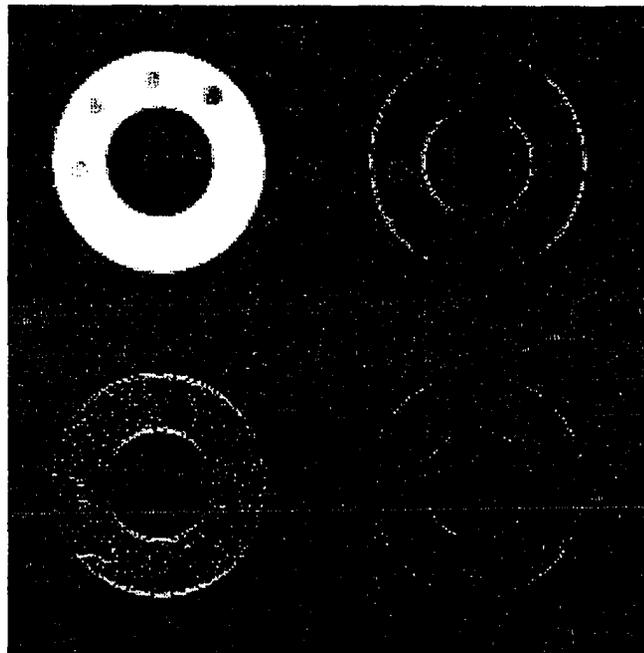


Figure 4.2: The first level wavelet decomposition of Figure 4.1

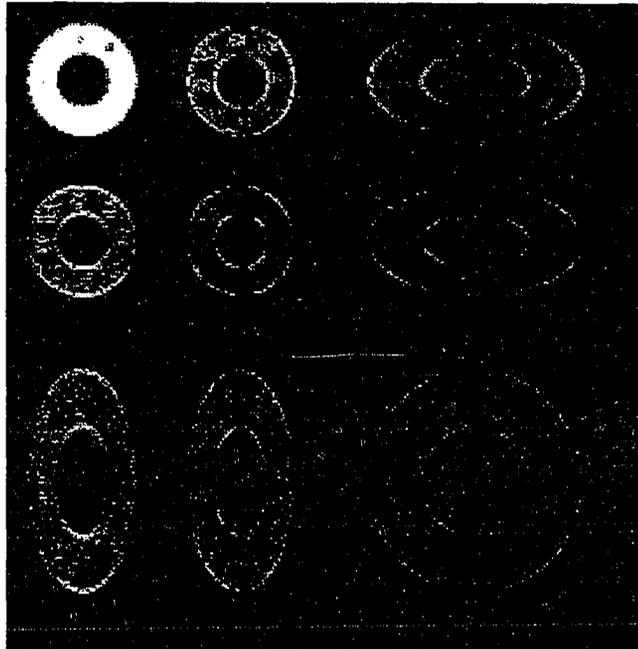


Figure 4.3: The second level wavelet decomposition of Figure 4.1

The first level of the wavelet decomposition of Figure 4.2 decomposes the original image of Figure 4.1 into four smaller versions, each one displays the decomposed signal within one spatial frequency subband channel. The upper left quadrant of Figure 4.2 represents a DC (smoothed) channel in both dimensions. The upper right quadrant represents the smoothing (into a DC channel) in the vertical direction and the differentiating (into a high frequency channel) in the horizontal direction. The lower left quadrant represents the smoothing in the horizontal direction and the differentiating in the vertical direction. Finally, the lower right quadrant represents the differentiating in the diagonal directions. Because every single pixel in a decomposed image corresponds to four pixels in the original image, the resolution for the decomposed images is reduced by a factor of 2 in each dimension.

Similarly, the second level of decomposition of Figure 4.3 decomposes the original image into 9 smaller versions, each one displays the decomposition at a spatial subband-channel narrower than those in the first level of decomposition (except for the lower right quadrant which is copied from the previous level of decomposition). Again, the top left corner represent the decomposition into a (narrower) DC channel, and the rest of the 8 smaller images represent the decompositions into various spatial-frequency channels. The resolutions for the second level of decomposed image are reduced further, which result in better frequency resolutions (narrower subband-channels) based on the uncertainty principle. The distortions in some of the decomposed images in Figure 4.3 are caused by the unequal scaling in the horizontal and the vertical directions. It is clear that the wavelet transform gives a multiresolution representation of an image through multichannel decompositions.

---

### Local Feature Extraction from Wavelet Decomposition

The decomposed signal components within each subband spatial-frequency channel carries out various information about the image. Because the transform is spatially localized within each decomposed channel, for each pixel location in the original image, we can directly locate one value in each channel associated with this pixel. This value represents the local frequency component within the spatial frequency channel, and may be used as a feature element for the pixel of interest to represent the local harmonic component within the channel. However, it is unrealistic to use all the subband-channel components as the feature values because when the level of decomposition increases, the number of subband channels increases exponentially and therefore, the size of the feature vector would be extremely large had the components of all channels been used. Studies of the wavelet transform also found that the majority of decomposed components are near zero [Press *et al.*, 1992], which indicates the wavelet transform has good energy compactness and a few larger transform parameters carry out most of the information about the signal. Considering these facts, a new feature extraction approach is developed to extract the largest decomposed components for local pixels in the image. This is done by comparing decomposed values for all of the subband channels and finding the dominant component as the feature.

Images must be pretreated in order to avoid the effect of artifacts in the wavelet transform. There are two sources for artifacts in the transform, one is caused by the wrapping around effect by the mirror filters in the transform matrix. The other artifact is caused by the artificial edges between the natural boundary of the image and the zero padding if the natural size of the image is not an integral power of 2. The artifacts in

wavelet transform are inconvenient for the feature selection process. Fortunately, the locations and the sizes of the artifacts can be anticipated in each step of wavelet decomposition. Artifacts occur at image natural boundaries and their width or height is proportional to the scale of wavelets at each decomposition level. A simple fix to get around the artifacts is to expand the image beyond its natural boundary before zero padding. This is done by dilating the image and padding pixels outside image boundary by its symmetrical point inside the image boundary. This process actually creates a symmetrical "safety" belt outside the image boundary and therefore avoids the artifacts inside the image natural boundary in the result of the wavelet transformed. When extracting image features, a 2-D amplitude normalized wavelet decomposition is performed on the original image up to a certain level of decomposition. A comparison among all decomposed components is made for every pixel location in the image. The component with the maximum magnitude is then used as a feature associated with a certain location. The choice of the total steps of decomposition to be performed is based on the size of the flaws to be detected. This is done through individual trials for various applications. During the training process, an operator tests training images and views the feature maps extracted from sequentially increased level of decompositions. The decomposition and feature extraction process continue until features selected characterize enough global information about the flaw. Once the total number of decomposition is determined through the trial, it is fixed during the automated detection process.

The procedure of local feature extraction based on the two-dimensional DAUB4 wavelet transform is shown graphically from Figures 4.4 through 4.6. Figure 4.4 illustrates a two-dimensional image and an arbitrary pixel of interest near which features are

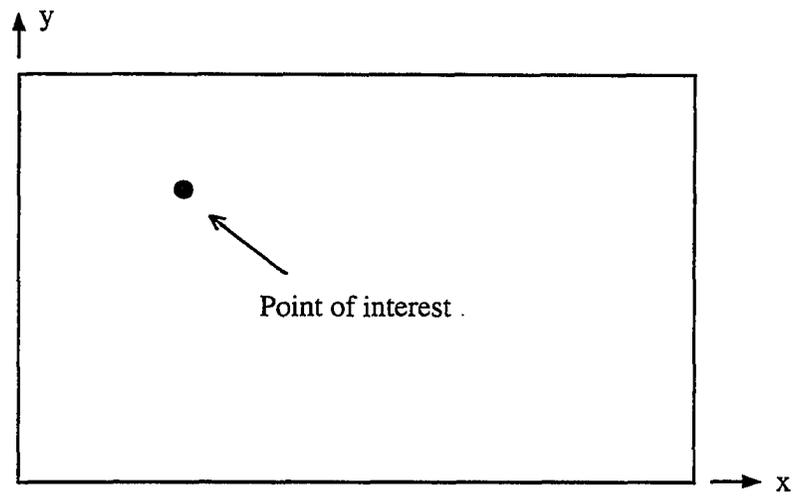


Figure 4.4: The sketch of a 2-dimensional image

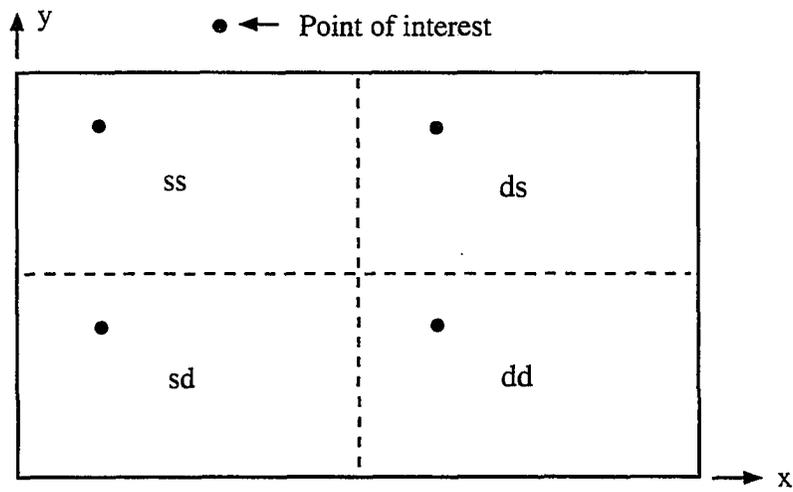


Figure 4.5: The sketch of the first level of decomposition



image, marked as "ss", "sd", "ds", "dd" (in the sketch of Figure 4.5) to indicate the combinations of smoothing "s" and decomposition "d" in horizontal and vertical directions, respectively. Since the transformation is localized, for every pixel of interest in the original image, one can point out four values in the first level transformed image to be associated with the point in the original image. There are one smoothed value in the upper left quadrant "ss", and three decomposed values in the regions associated with "d" representing decompositions in horizontal, vertical and diagonal directions. Similarly, for a second level decomposed image, there are nine smaller versions of the original image, as sketched in Figure 4.6, in which the capital letters indicate second level operations. Note the pattern of second level of decomposition. The upper left quadrant of Figure 4.5 has been decomposed in both directions, the lower right quadrant of Figure 4.5 is not decomposed at all, and the upper right and lower left quadrants of Figure 4.5 are decomposed in one direction only. There are totally nine values in all decomposed regions associated the pixel of interest in the original image. The smoothed and decomposed values, although scaled differently, are all representative of certain pixels in the original image. Therefore, for a level two wavelet transform, one can find 12 values associated with one pixel in the original image (There are totally 13 values in the first and second level transforms, but the first level diagonal decomposition is not changed in the second level of the transform). As the level of decomposition increases, the decomposed values representing the original pixel of interest increase in a near exponential manner. Basically, all the transformed values associated with a certain pixel in the original image can be used as feature values for that pixel in a classification scheme. If this happens, as the level of operation increases, the number of features in the feature vector will increase explosively

which provides too much computation load for the classification process. Therefore, the number of features has to be reduced in order to speed up the classification.

In multiple channel spatial-frequency analysis such as Gabor filtering, power spectrum of each channel is calculated and features are sometimes extracted by choosing the maximum spectral component [Dunn and Higgins, 1994], or by computing moments [Bigun and du Buf, 1994] over all the channels in order to reduce the size of the feature vector. Similar concepts can be used in the wavelet transform as well. In contrast to the time-frequency pair in Fourier analysis, the corresponding variables in wavelet transform are translation and scale, with translation corresponding to time, and scale corresponding to frequency. Therefore, values at various scales corresponding to the components at various "frequency" channels. The feature extraction process is accomplished by comparing all the "frequency" (scale) components for a certain location, and choose the one with maximum magnitude (or the average of the larger values) as the dominant "frequency" component for that location. This process is done at successive decomposition levels and therefore a multiresolution representation is achieved through the feature extraction process. For example, for the local point marked in Figure 4.4, there are four "frequency" components in the first level transformation shown in Figure 4.5, in which the smoothed value in "ss" represents the DC component, and the value in "sd" represents DC in the horizontal direction and spatial frequency in the vertical direction. Similarly, values in other quadrants also represent certain frequency components for that location. At level one decomposition, the value in "ss" can be chosen as one feature to indicate the average tone in that area, which is symbolized as  $f_1 = ss$ . The maximum component of all the other three decomposed, or partially decomposed areas, symbolized

as  $f_2 = \max (sd, ds, dd)$ , can be chosen as the other feature which is equivalent to the dominant frequency component at this level of view. When the decomposition is continued into the second level sketched in Figure 4.6. There are nine values in the transformed image representing a full range of different discrete spatial frequency components for the location of interest. The decomposed values are again compared and the maximum magnitude  $f_3 = \max (SD, DS, DD, Sd, Dd, dS, dD, dd)$  is used as the third feature represents the dominant frequency component at level two decomposition. This process is continued until a desired level of decomposition is reached. A local feature vector  $\mathbf{F} = (f_1 \ f_2 \ f_3 \ \dots)$  is constructed using all features extracted in the local positions and is classified later by a classifier. In addition to the "spectrum" for the dominant local frequency components, the scales of these components representing "which frequency is the dominant frequency" are also important measures of local harmonics. This information is obtained by recording the scales where the maximum decomposed components are, and for every value chosen, its scales in the x and y directions are also recorded. However, experiments show that scale information is vulnerable to noise and is less representative of image geometric structures. Therefore, the scale is eventually dropped in the process of feature selection.

An example of features extracted by the above process is displayed in Figures 4.7, 4.8, 4.9 and 4.10. The original image is an X-ray image of weld displayed in Figure 4.7. The weld portion is the light horizontal band through the image. A dark flaw lies within this band. Figure 4.8 displays a second level smoothed feature (averaged "SS" DC component) for the image, indicating the average pixel intensity near each pixel. Figure 4.9 and Figure 4.10 display the maximum value among the decomposed components in

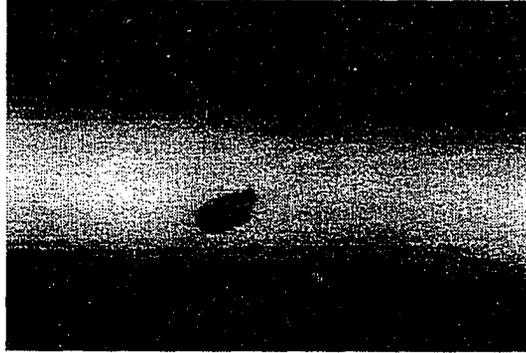


Figure 4.7: A weld X-ray image

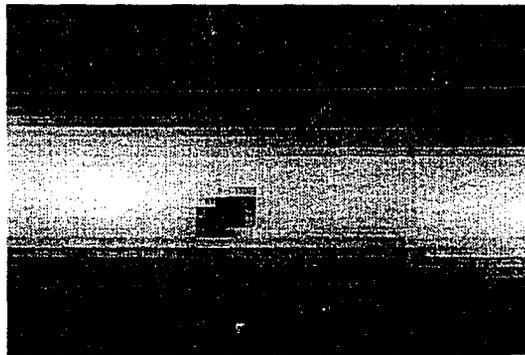


Figure 4.8: The smoothed feature extracted from the second level of decomposition of the image in Figure 4.7

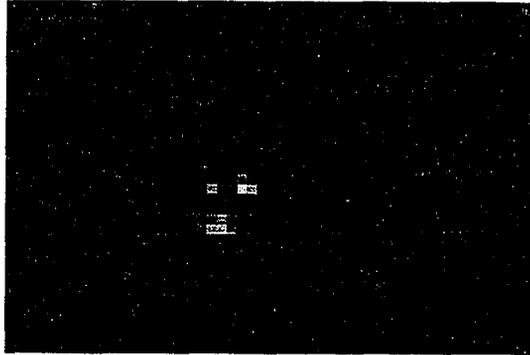


Figure 4.9: The decomposed feature extracted from the second level of decomposition of the image in Figure 4.7



Figure 4.10: The decomposed feature extracted from the third level of decomposition of the image in Figure 4.7

subband channels (e.g.,  $f = \max(SD, DS, DD, Sd, Dd, dS, dD, dd)$  for the second level of decomposition) after the second level and the third level decompositions, respectively. It is clearly shown in Figure 4.9 and Figure 4.10 that features extracted from multichannel wavelet decompositions gives a multiresolution characterization and provide contextual information about that image. This is analogous to human visual perception. Imagine a person taking a look at a picture at a closer distance, then stepping back and taking another look. Each time, he/she may extract new information from the picture. The actual process here is accomplished by a computer through several steps of view.

## CHAPTER 5. FUZZY RULE-BASED CLASSIFICATION

An image segmentation process basically involves two parts, feature extraction and classification. In previous chapters, feature extraction based on the localized wavelet transform has been discussed. Once the local features of an image are extracted, the next step is the classification process that divides the image into different regions according to the features extracted from those regions. Classification can be categorized as supervised or unsupervised classification based on whether or not extra training information has been given to a classifier. Most of the clustering algorithms for pattern recognition, such as k-mean [Coleman and Andrews, 1979], maximum distance [Batchelor, 1974], and fuzzy-C means [Bezdek, 1982] algorithms, are unsupervised because the classifiers divide the image regions only based on how feature vectors are separated in the feature space, and classifications are computed through iterations. When classifying an image with a complicated geometric background, feature vectors are usually not well separated in the feature space. In this case, extra information about each application on hand is needed to achieve more customized classification. Supervised classifiers are used in this dissertation to provide subjective and customized classification. The customization is done through the training. A typical example of a supervised classifier is the backpropagating perceptron [McClelland and Rumelhart, 1986], in which a system func-

tion is built from training data. The main problem associated with the perceptron is the convergence difficulty during the training process. The randomly chosen weights in an artificial neural network cause the training process to be slow and convergence may not be guaranteed. An adaptive fuzzy system is an alternative universal approximator and is usually initialized from training data itself. Therefore, it can greatly speed up the training process. Additionally, fuzzy logic is a soft decision making process which is more robust in complicated situations in which a crisp decision is sometimes difficult to make. In this chapter, supervised classifiers for image classification applications based on fuzzy logic are studied.

### **Fuzzy Logic and Fuzzy Systems**

Fuzzy sets and fuzzy logic theory were first introduced by Lotfi A. Zadeh in the 1960's in an attempt to model the uncertainty of natural languages [Zadeh, 1965]. The name fuzzy was used because it was once considered "unscientific", or "inaccurate" in the mind of conventional logic. The conventional logic is a bivalent logic such as "yes" or "no", "black" or "white", but nothing in between. Fuzzy logic handles the in between or "gray" cases, which are sandwiched between the complete truth and the complete falseness of bivalent logic. It is a "soft" logic and measures everything in a matter of "degree", or by the so called fuzzy membership values. Compared to fuzzy logic, conventional logic is "hard" and handles the extreme cases of fuzzy logic.

Fuzzy membership uses any number between [0, 1] to measure the degree of truth. For example, if  $m_A$  is the membership value of a statement A, then

$m_A = 1$  indicates statement A is completely true,

$m_A = 0$  indicates the statement A is completely false,

$m_A = (0, 1)$  indicates the statement is partially true, the degree of truth is given by  $m_A$ .

A fuzzy membership function is a continuous function that characterizes fuzzy membership. Figure 5.1 shows a simple membership function. It displays the membership of tallness as a function of a person's height. It gives a membership value of 1 to a person above 7', a membership value of 0 to a person below 5' and a value in between for other heights.

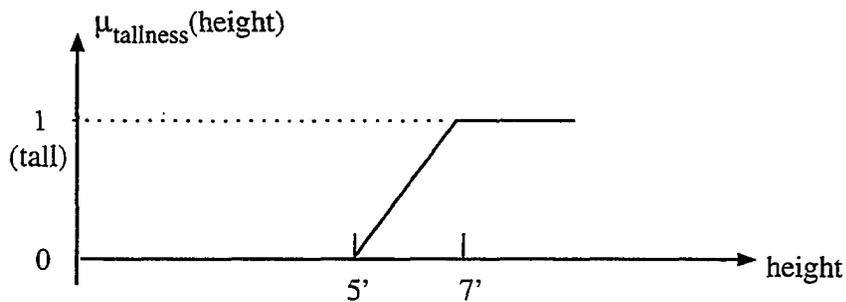


Figure 5.1: The tallness membership function as a function of height

### Fuzzy set

The fuzzy set is defined on top of the conventional set theory. A fuzzy set A in some relevant universe X is defined as [Zadeh, 1965]:

$$A = \{(x, \mu(x)) \mid x \in X, \text{ and } 0 \leq \mu(x) \leq 1\} \quad (5.1)$$

where  $x$  represents the elements of  $X$  that partially belong to the set  $A$ , and  $\mu(x)$  is the membership function that measures the degree of membership for each  $x$  that belongs to  $A$ . The membership function  $\mu(x)$  provides a measure for the degree of belonging, with  $\mu(x) = 0$  representing the elements that are completely not in  $A$ ,  $\mu(x) = 1$  representing the elements that are completely in  $A$ , and  $\mu(x) = (0, 1)$  representing partial inclusion in  $A$ .

The basic logic operations for the fuzzy set defined by Zadeh [1965] are:

$$A \cap B = C = \left\{ (x, \mu_C(x)) \mid x \in X, \text{ and } \mu_C(x) = \min [\mu_A(x), \mu_B(x)] \right\} \quad (5.2)$$

$$A \cup B = C = \left\{ (x, \mu_C(x)) \mid x \in X, \text{ and } \mu_C(x) = \max [\mu_A(x), \mu_B(x)] \right\} \quad (5.3)$$

$$\bar{A} = C = \left\{ (x, \mu_C(x)) \mid x \in X, \text{ and } \mu_C(x) = 1 - \mu_A(x) \right\} \quad (5.4)$$

$$A \subseteq B, \text{ if } \mu_A(x) \leq \mu_B(x) \text{ for all } x \in X \quad (5.5)$$

where  $\mu_A(x)$ ,  $\mu_B(x)$ , and  $\mu_C(x)$  are the membership functions for fuzzy sets  $A$ ,  $B$ , and  $C$ , respectively. When the membership function  $\mu(x)$  is limited to take only 0 and 1, the fuzzy set operations reduce to the corresponding operations for the conventional sets. Therefore, the conventional set operations are special cases of the fuzzy set operations.

An  $\alpha$ -cut of a fuzzy set is the set of elements of a fuzzy set that have a membership value larger than  $\alpha$ , defined as

$$A_\alpha = \{x \mid x \in X, \text{ and } \mu(x) > \alpha\}. \quad (5.6)$$

The collection of all  $\alpha$ -cuts of a fuzzy set defines the set.

### Nonlinear fuzzy system

A typically fuzzy system consists of functional blocks such as fuzzification, knowledge base, fuzzy inference and defuzzification [Lee, 1990], as shown in Figure 5.2. The input of the system is nonfuzzy data, the fuzzification interface establishes input fuzzy sets and the associated membership functions from the crisp input. The fuzzy inference is the decision making logic in the system which generate output fuzzy sets, and the defuzzification interface summarizes the output fuzzy sets and gives nonfuzzy system output. A system knowledge base provides rules obtained from training data and human experts for the processes. It is clear from Figure 5.2 that the fuzzy system is a nonlinear system which maps a crisp input to a crisp output, while fuzzy reasoning and the knowledge base control the performance of the system.

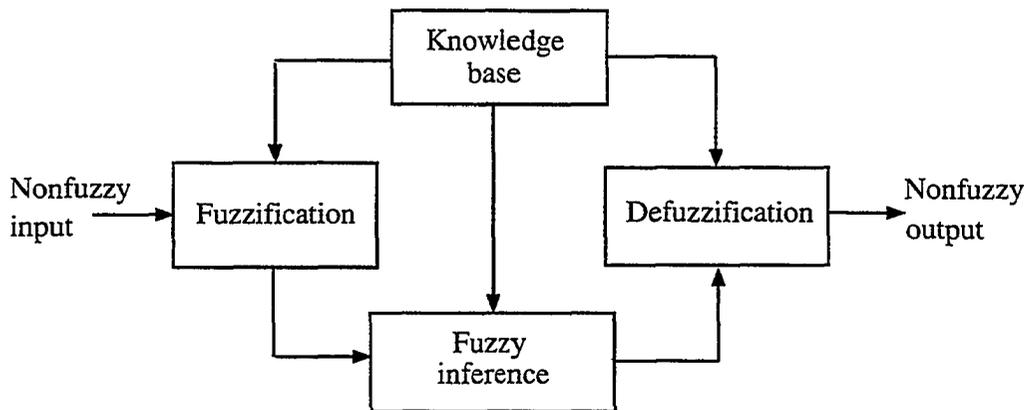


Figure 5.2: The block diagram of a fuzzy system

In image understanding and inspection applications, the input data for a fuzzy classifier are image feature vectors associated with image pixels to be classified. Suppose a pixel in an image has a local feature vector  $\mathbf{X} = (x_1 \ x_2 \ x_3 \ \dots \ x_N)$ , where the elements  $x_i$  ( $i = 1, 2, 3, \dots, N$ ) of the feature vector are the  $N$  feature values extracted in the local area around that pixel. For the feature vector extracted from the wavelet decompositions, the first feature element  $x_1$  is usually the smoothing feature extracted from a DC channel, while the rest of the features are extracted from the differentiating channels at each level of decomposition. Given the input feature vector  $\mathbf{X} = (x_1 \ x_2 \ x_3 \ \dots \ x_N)$ , the fuzzy rules to control the classifier could be "and" rules as the following:

Rule #1: *IF feature  $x_1$  is near  $\bar{x}_{11}$ , and  $x_2$  is near  $\bar{x}_{21}$ , and  $\dots$ , and  $x_N$  is near  $\bar{x}_{N1}$ , THEN output is  $\bar{z}_1$ .*

.

.

.

Rule #j: *IF feature  $x_1$  is near  $\bar{x}_{1j}$ , and  $x_2$  is near  $\bar{x}_{2j}$ , and  $\dots$ , and  $x_N$  is near  $\bar{x}_{Nj}$ , THEN output is  $\bar{z}_j$ .*

.

.

.

where the "bar" symbols are the centroids of fuzzy membership functions to be determined by the training process.

### Adaptive Fuzzy Classification

An adaptive fuzzy system (or neural fuzzy system) and a neural network system are all proven to be universal approximators [Wang and Mendel, 1992; Hornik *et al.*, 1989]. Therefore, both systems can be used to represent any nonlinear functions if given enough training data. The conventional feed forward artificial neural networks are in-

ialized from randomly chosen parameters. A neural system is established by updating the parameters from the training process until the system converges. The training process of a neural network is slow because of its initial blindness and the convergence is sometimes hard to reach. An adaptive fuzzy system, however, is very fast in the training stage and guarantees a convergence due to fact that it is initialized directly from the training data. The early applications of adaptive fuzzy systems in the field of electrical engineering are mostly in the control area, where fuzzy systems are used to built nonlinear system functions [Lee, 1990]. The feature classifier for image classification can also also be considered as a nonlinear mapping function, the input of the function is a feature vector and the output of the function is a number used to mark or label different classes to be classified. Therefore, if properly trained, an adaptive fuzzy system is also a feature classifier that achieves a nonlinear mapping from feature vectors to classified label image.

### **Fuzzy membership functions**

A fuzzy system is established by building fuzzy membership functions for the system. The process for establishing fuzzy membership for the system input is called fuzzification. A membership function can be of many forms. In the adaptive fuzzy system discussed here, a Gaussian form is used. The main reason for using a Gaussian is that its derivative, which is needed later to achieve least mean square training easier to obtain. The Gaussian membership function for the  $i$ th feature in the  $j$ th rule , denoted as  $\mu_{ij}$ , is given as an unit-amplitude Gaussian function by

$$\mu_{ij}(x_i) = \exp \left[ -\frac{(x_i - \bar{x}_{ij})^2}{2\sigma_{ij}^2} \right] \quad (5.7)$$

where  $x_i$  is a variable for feature  $i$  for feature vector  $\mathbf{X}$ , the mean  $\bar{x}_{ij}$  and standard deviation  $\sigma_{ij}$ , which are used to control the position and the shape of the Gaussian function, are subject to the fuzzy rules. The index  $i$  is the index for the feature elements and  $j$  is the index for the rules. For every feature element within each rule, there is a membership function associated with that feature. Figure 5.3 displays graphically membership functions of a size 3 feature vector  $\mathbf{X} = (x_1 \ x_2 \ x_3)$ , in which solid curves indicate the membership function for the current  $j$ th rule, and the dotted curves represent membership functions for other rules.

The output of a fuzzy system is generated through the process of defuzzification. The defuzzification interface sums up the output fuzzy set and gives a crisp system output. In our image inspection application, the output fuzzy set is associated with each fuzzy rule created from the training data. That is, the output fuzzy set measures the degree of truth for every rule in the knowledge rule base. For a certain input feature vector  $\mathbf{X}$ , the output fuzzy membership function for each rule is obtained by multiplying the fuzzy membership functions for all feature elements in the feature vector associated with each rule. The output fuzzy membership function for the  $j$ th rule, denoted as  $\mu_j^o$  where the superscript "o" stands for the "output", is given by

$$\mu_j^o(\mathbf{X}) = \mu_{1j}(x_1) \cdot \mu_{2j}(x_2) \cdot \dots \cdot \mu_{Nj}(x_N) = \prod_{i=1}^N \mu_{ij}(x_i) \quad (5.8)$$

which is a function of the input feature vector  $\mathbf{X} = (x_1 \ x_2 \ \dots \ x_N)$  and its associated membership functions  $\mu_{ij}$ .

The nonlinear system output is then calculated by the standard centroid defuzzification formula which is a normalized sum of all the elements of the output

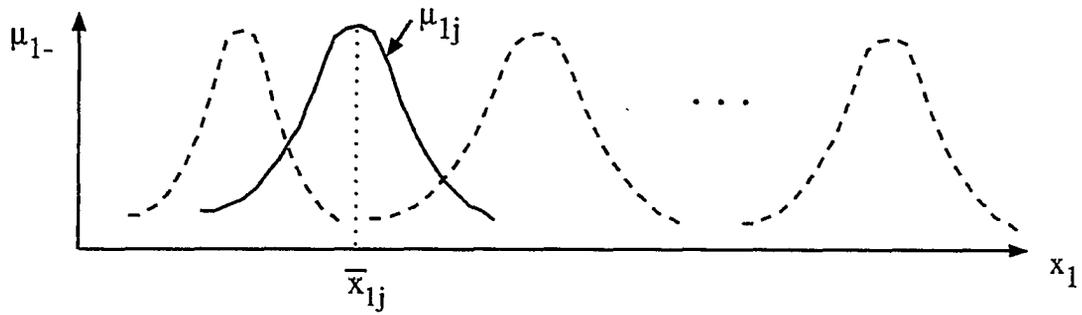
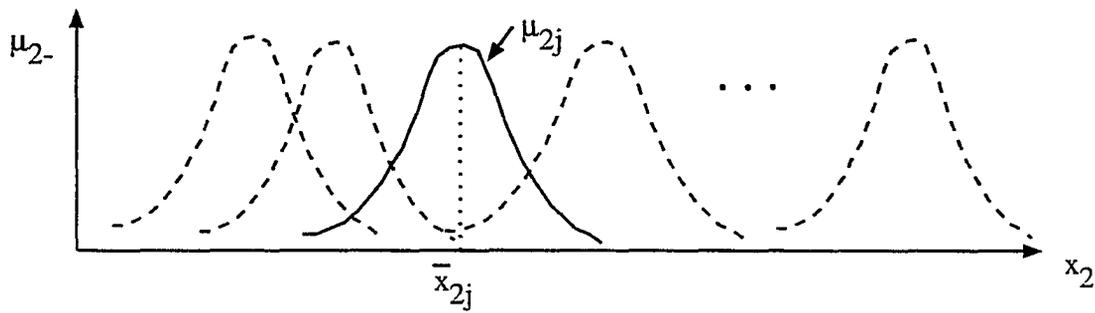
(a) Gaussian membership function for feature  $x_1$ (b) Gaussian membership function for feature  $x_2$ (c) Gaussian membership function for feature  $x_3$ 

Figure 5.3: A sketch of Gaussian membership functions

fuzzy set. Thus,

$$f(\mathbf{X}) = \frac{\sum_{j=1}^M \bar{z}_j \mu_j^o(\mathbf{X})}{\sum_{j=1}^M \mu_j^o(\mathbf{X})} = \frac{\sum_{j=1}^M \bar{z}_j \left[ \prod_{i=1}^N \mu_{ij}(x_i) \right]}{\sum_{j=1}^M \left[ \prod_{i=1}^N \mu_{ij}(x_i) \right]} \quad (5.9)$$

where  $N$  is the size of feature vector,  $M$  is the total number of fuzzy rules applied and  $\bar{z}_j$  is the output centroid of the  $j$ th rule, at which the degree of truth  $\mu_j^o$  reaches the maximum of 1. The output centroid  $\bar{z}_j$  is obtained from system training process covered later in this chapter. Function  $f(\mathbf{X})$  is actually the transform function of this nonlinear system which maps the nonfuzzy input feature vector into the nonfuzzy output value.

### Least square error adaptive training

The training process of an adaptive fuzzy system is to establish a fuzzy rule base from the training data. This is achieved by assigning the parameters such as mean, or standard deviation, for the Gaussian membership functions. These parameters are usually initiated directly from the training data. Table 5.1 shows a typical training data file that has  $M$  data points ( $M$  rows in Table 5.1). Each row in Table 5.1 contains a feature vector extracted for a prototype pixel in the training image and the desired output label for that prototype. The element  $x_{ij}$  represents the  $j$ th training data for the  $i$ th feature, and the value  $d$  represents the corresponding desired output for the  $j$ th data.

The fuzzy membership function of Equation (5.7) is initiated by assigning the functional parameters  $\bar{x}_{ij}$  and  $\sigma_{ij}$  directly from the training data as

Table 5.1: Prototype feature vectors and the desired output values

Prototype feature values $x_{ij}$	Desired output
$x_{11}, x_{21}, x_{31}, \dots, x_{N1}$	$d_1$
$x_{12}, x_{22}, x_{32}, \dots, x_{N2}$	$d_2$
.	.
.	.
.	.
$x_{1M}, x_{2M}, x_{3M}, \dots, x_{NM}$	$d_M$

$$\bar{x}_{ij} = x_{ij}, \bar{z}_j = d_j, \forall i, j;$$

$$\sigma_{ij} = \frac{\max(x_{ij}) - \min(x_{ij})}{const}, j = 1, 2, \dots, M, \text{ for every } i. \quad (5.10)$$

It is clearly shown in Equation (5.10) that the fuzzy membership function for each feature is initiated directly from the training data because the centroids of the Gaussian functions are assigned by the prototype features and the standard deviations of the Gaussian functions are assigned according to the dynamic range of each feature in the training data file. Once the fuzzy rules and membership functions are initiated, the system output can be computed from  $f(\mathbf{X})$  of Equation (5.9) for any training feature vector  $\mathbf{X}$ . The output error  $e_j$  for the  $j$ th training feature vector  $\mathbf{X}_j$  is determined by the difference between the defuzzification result and the desired output for that training prototype,

defined as

$$e_j = \left[ f(\mathbf{X}_j) - d_j \right]. \quad (5.11)$$

The least square error training is achieved by the steepest descent iteration which reshapes the fuzzy membership functions of Equation (5.7) by updating the mean and standard deviation until the error is minimized. The centroids of the output fuzzy set denoted as  $\bar{z}_j$  is also updated. The iterative equations for updating membership functions are

$$\begin{aligned} \bar{x}_{ij}(k+1) &= \bar{x}_{ij}(k) - \alpha \frac{\partial e_j^2}{\partial \bar{x}_{ij}}, \\ \sigma_{ij}(k+1) &= \sigma_{ij}(k) - \alpha \frac{\partial e_j^2}{\partial \sigma_{ij}}, \\ \bar{z}_j(k+1) &= \bar{z}_j(k) - \alpha \frac{\partial e_j^2}{\partial \bar{z}_j}, \end{aligned} \quad (5.12)$$

where  $k$  is the iteration number and  $\alpha$  is a step size that is usually smaller than 1.

### **Eliminating redundant rules**

In the process of creating training data, the prototype features generated prior to the training process may not necessarily be independent of each other. Most of the time, training data generated directly from the training images are redundant. This means that training data might be highly related with many duplicated data points in a training data file. If the training process blindly loads every data point from the training data file, there would be duplicated rules in the established rule base which unnecessarily increase the

size of the rule base, and hence, the computational load. Therefore, it is desirable to optimize the rule base by eliminating redundant rules.

To establish an efficient rule base from redundant training data, each rule is established one-by-one from the training data file. As shown in Figure 5.4, the first rule is loaded directly from the first data point in the training data file. The system has then only one rule. When loading the next training data, the data is used as the input feature vector of the established (the first) rule. If the output membership value obtained by Equation (5.8) is larger than a preset threshold value (usually  $0.5 < \text{threshold} < 1$ , depending on the inference scheme and the application) and its desired output value is equal to that of the first rule, this data point is treated as a duplication of the first rule loaded and can be discarded. Otherwise a new rule based on the second training data point will be established and the number of rules in the system is updated to two. Then a third data point in the training data file is examined based on the existing rules. It is either constructed into a new rule or discarded as redundant data, based on the same threshold. This process continues for the rest of the training data contained in the training data file.

After eliminating duplicated data points and initiating the fuzzy rule base for the system, the adaptive training process discussed previously is applied on the existing rules by calculating the error between the system output and the desired output for the training feature vectors (Equation (5.11)) and updating the functional parameters for each membership function (Equation (5.12)) until the error is minimized. Compared to the training of a conventional backpropagating neural network [McClelland and Rumelhart, 1986], the training process for an adaptive fuzzy system is faster and convergence is easier to achieve because all parameters are initiated directly from the training data.

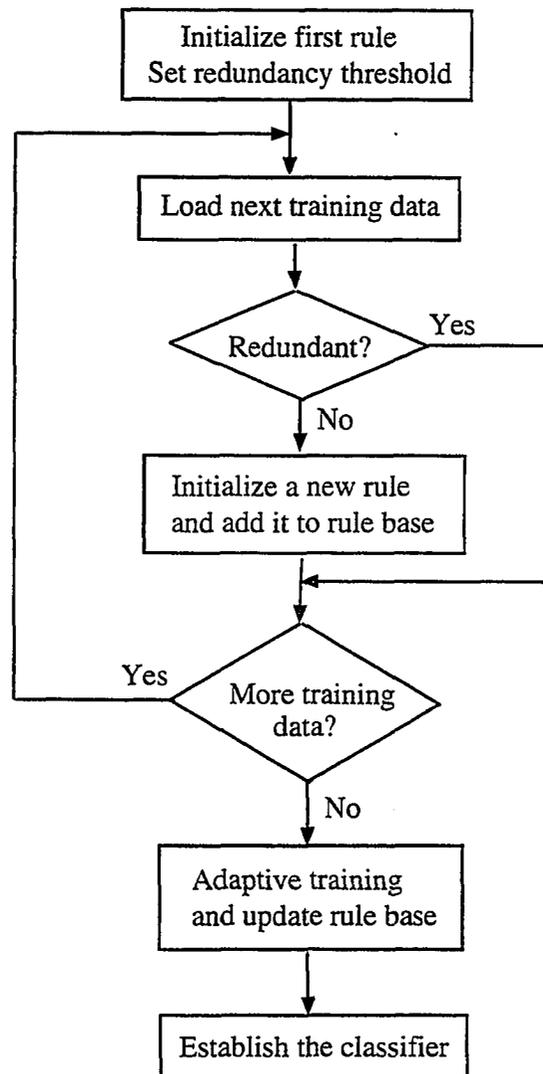


Figure 5.4: Selectively establish the fuzzy rule base from the training data

### **Optimized Fuzzy Classification**

If complete training data are provided, a general impression about a fuzzy system is that the more rules are used, the better does the system perform. This is in fact true if the execution speed is not a concern. However, in most of the image classification applications, especially for real time applications, speed is a major concern for the system. Therefore, a reduction in the number of rules at the minimum cost of reliability is needed to achieve a higher speed.

As discussed in the previous section, an adaptive fuzzy classifier is initiated from the training data file and the total number of rules can be reduced by eliminating similar rules, but the choice of the redundancy threshold is dependent on the application on hand and the fuzzy inference scheme used for a particular classifier. More importantly, the reduction is at the cost of losing information and the question of what is the optimal number of rules for a given problem remains unanswered. In this section, a new fuzzy inference scheme is introduced in order to create an optimal fuzzy rule base for given training data and minimize the number of fuzzy rules in the knowledge base without losing any information contained in the training data.

#### **Asymmetric trapezoidal membership function**

A new fuzzy membership function is introduced here in order to establish an optimal fuzzy rule base. The membership function is generalized in an asymmetric trapezoidal shape. The reason for using a trapezoidal function is that it can be formed into either a triangular form (a special case of a trapezoid) or a flat top trapezoidal form that can provide a certain range of saturation. Figure 5.5 shows a general form of the membership

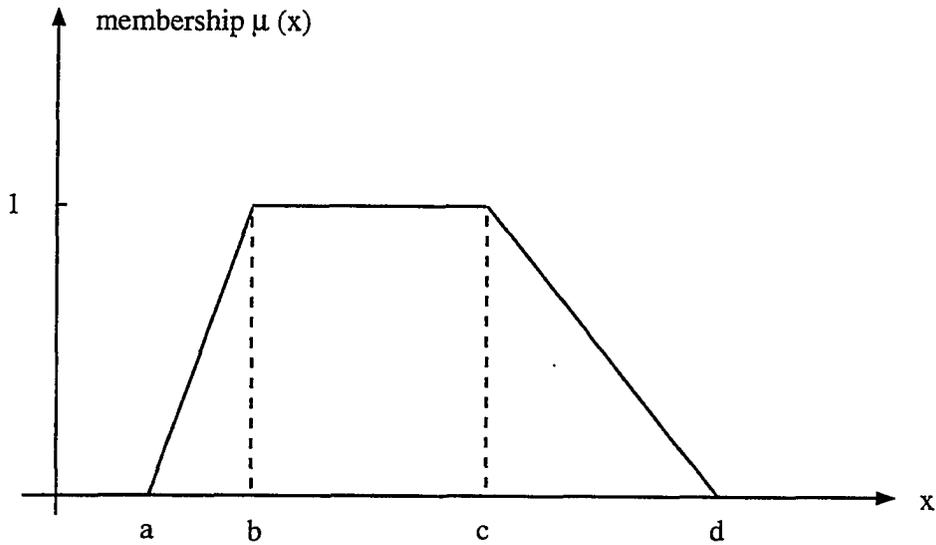


Figure 5.5: The asymmetric trapezoidal fuzzy membership function

function and its mathematical expression is given by

$$\mu(x) = \begin{cases} \frac{x-a}{b-a}, & a < x < b \\ 1, & b \leq x \leq c \\ \frac{d-x}{d-c}, & c < x < d \\ 0, & \text{elsewhere} \end{cases} \quad (5.13)$$

where  $a$ ,  $b$ ,  $c$ , and  $d$  are four parameters that control the location and shape of the membership function. Parameters  $b$  and  $c$  are used to determine the central location of the membership function, while  $a$  and  $d$  are used to determine the edge locations.

Unlike the Gaussian function used for adaptive fuzzy classifier, the trapezoidal membership function is non-zero only when variable  $x$  is located between  $a$  and  $d$  and is zero elsewhere. This eliminates the computation need if  $x$  does not fall into the effective range of a membership function. The flat top of the trapezoid makes the membership one when  $x$  is not only at centroid, but also near the centroid. This property is used later to combine many similar training data points into one rule.

The optimal fuzzy classifier is established such that every feature element within each rule will have a trapezoidal membership function. Analogous to an adaptive fuzzy system, to associate membership function with features in fuzzy rules, subscripts  $i, j$  are used in the trapezoid membership function to indicate the feature and rule, respectively. For example,  $\mu_{ij}$  is the membership function of  $i$ th feature in  $j$ th rule, and  $a_{ij}, b_{ij}, c_{ij}, d_{ij}$ , are the four functional parameters for  $\mu_{ij}$ . These four parameters specify the shape and location of each membership function and will be determined during the training process.

### **Establishment of rule base**

As mentioned earlier, a fuzzy rule base is established directly from the training data. The optimal number of rules used in the knowledge base depends on the images to be classified and how well features separate in the feature space. It is desirable for a system to have the least possible number of rules to achieve fast classification, but still maintain good reliability. In order to reduce the number of rules without losing much information given by the training data, each rule in the knowledge base should be able to represent information provided by multiple data points in a training data file. The idea is to use as few rules as possible, and make the effective range ( $\alpha$ -cut interval for  $\alpha = 0$ ) of

each established rule as large as possible. Multiple "same kind of" training data points belonging to the same class will be combined into the same rule, and therefore the total number of rules is greatly reduced.

The membership function for each feature is constructed by determining its parameters  $a$ ,  $b$ ,  $c$ ,  $d$  shown in Figure 5.5 and Equation (5.7). Before construction of the fuzzy rule base, a default effective range value has to be set to define the minimum effective range for a triangular membership function (equal to  $d - a$ , for  $b = c$  in Figure 5.5). This is to make sure the effective range of the membership function is not too small. The minimum range value is determined by the dynamic range of the feature values for the training data and total number of different feature clusters of existing for the image. Feature vectors are considered within one cluster if they close to each other in the feature space. Note that the name "cluster" used here is different from the name "class" for a classification. The number of clusters is determined by the nature of an image, which may be obtained by an iterative clustering algorithm for unknown number of clusters [Batchelor, 1974]. The number of classes is, however, subject to the choice of the user for various applications. Therefore, the number of classes for a classification application may be different from the number of clusters in an image. For example, a flaw detection application may divide an image into two classes, flaw and nonflaw. In the feature space, however, the flaw features may form one cluster, the object features may form one or more clusters, and image background features may form another cluster. Therefore there are more clusters in image feature space than the total number of classes used here. The default minimum range value, which is the default nonzero range for a triangular membership function ( $b = c$ ), is obtained by dividing the dynamic range of a

feature by the total number of clusters in the image. This will make sure the established fuzzy rule to cover the full range of each feature.

The asymmetric trapezoidal membership functions are established in the following way. In the training data file, if there is a cluster of training data points that belongs to the same class, then these data points are combined into one rule. The membership functions of this rule are constructed such that the central parameters  $b$  and  $c$  are the minimum and the maximum features in that cluster. The edge parameters  $a$  and  $d$  are usually locations where the nearest features of other classes are located. This forces features that are different from the chosen class to have zero membership for that class. If there is no other class feature located on one side of  $b$  or  $c$ , or if the nearest other class feature is too close to  $b$  or  $c$ , edge parameter are determined automatically by the default minimum range value.

The flow chart of the rule-base establishment procedure is displayed in Figure 5.6, in which an optimal rule base is established from the training data. The training data file contains training features and the desired output label for each class. The total number of classes in the training data file can be determined by scanning through the desired output classes. To ease the comparison of training data within one class or between two different classes, the training data out of the data file are rearranged into a structure in which the training data are sorted by class. The first rule for the first class is established from the first data vector read from the sorted training data structure. The read-in feature values are used as the center of each feature membership function for this rule (points  $b$  and  $c$  in Figure 5.4, and  $b = c$  in the beginning). The program then searches in the training data file for the features belonging to other classes that are the nearest to the input

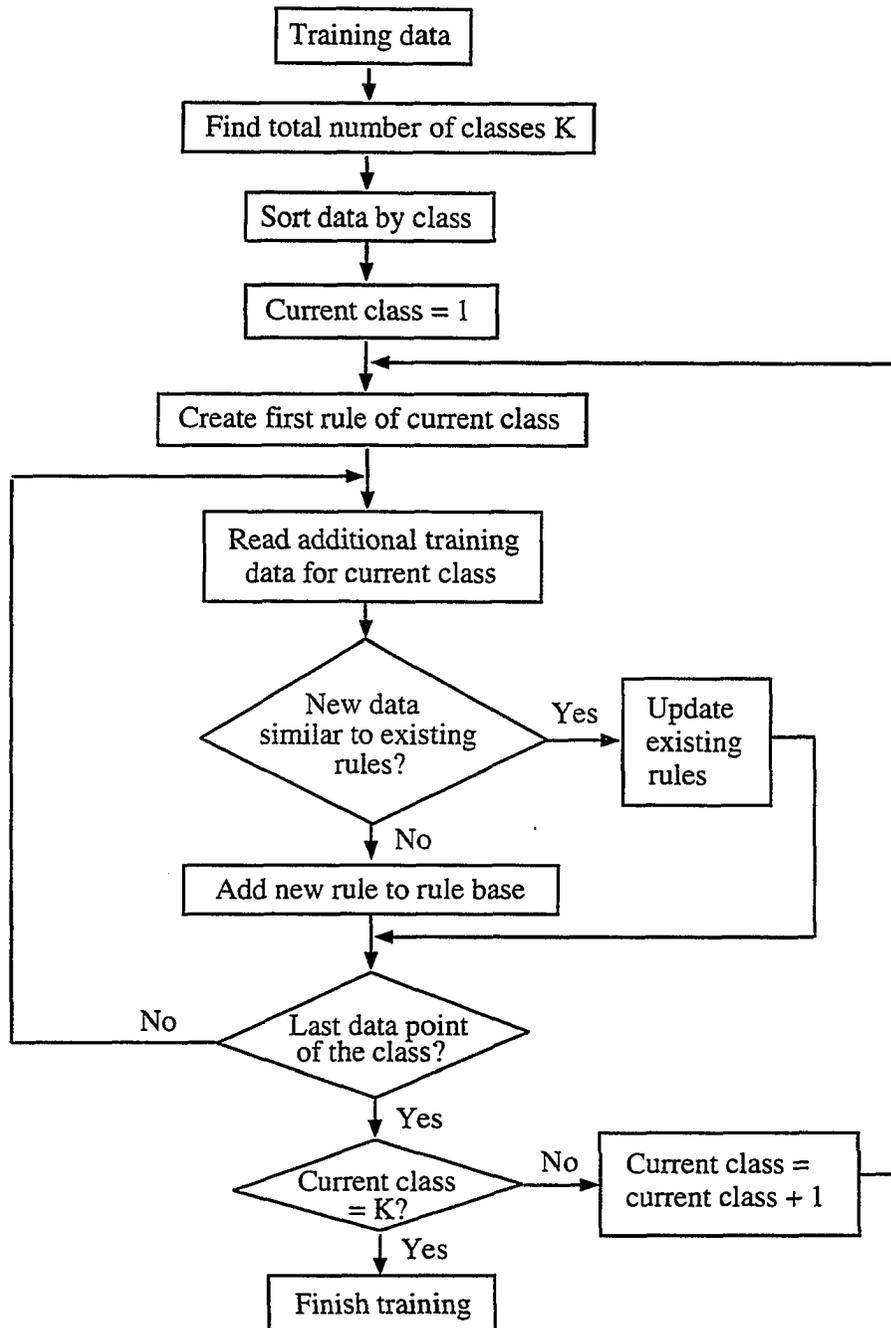


Figure 5.6: The optimized training from the training data

features, and assigns these locations as the edge points of the membership function (points, a or d in Figure 5.5). This initiation procedure enables a test feature vector similar to a training feature vector to have a membership value near one for the current class and a membership value near zero for other classes, provided that different classes are well separated in the feature space.

After the establishment of the first rule, the next training data vector for the same class (if any) is loaded. If all features in the new data vector fall in the effective range of the membership functions that already exist for the class (e.g., range d-a in the membership functions for the first rule), this data vector is compared with the central parameters of the existing membership functions (b and c in Figure 5.4) and the minimum and maximum values in the comparison are used to update the central parameters b and c, respectively, for the existing membership functions. This procedure actually updates the existing rule using the information from a new data point. The total number of rules, however, does not increase. If any feature in this new data vector is beyond the nonzero range of the membership functions of the existing rules, a new rule will be added to the fuzzy rule base for the same class, and the membership functions of the new rule are established following the same procedure for the first rule. The procedure is completed consequently for the other classes in the training data file until all the data have been loaded. This process effectively reduces the total number of rules in the knowledge rule base by combining similar training data vectors into one rule without losing any information, because all of the data points in the training data file are utilized in the combined rules.

In many applications, features for various classes may not be well separated in the feature space. Some feature values for different classes could be very close or equal to each other, and their difference may be represented by other features in the feature vector. Therefore, simply minimizing the membership for neighboring features belonging to other classes may create trivial membership functions (such as a delta function ) if the resulting non-zero effective range is too small. To avoid the above trivial cases, one should overlap multiple rules for different classes at the same location. In fuzzy logic, the sum of the membership values for all the classes is not bounded by any constant. This is a significant difference between fuzzy and probability theories. In probability theory, all the probabilities sum up to one and appearance of one event prevents the appearance of its opposite. Fuzzy theory allows overlap among membership functions for different classes. Therefore, for membership functions with small non-zero interval, we can extend their non-zero interval to the default minimum value previously set from the dynamic range of each feature, and hence, partially overlap membership functions belonging to different classes.

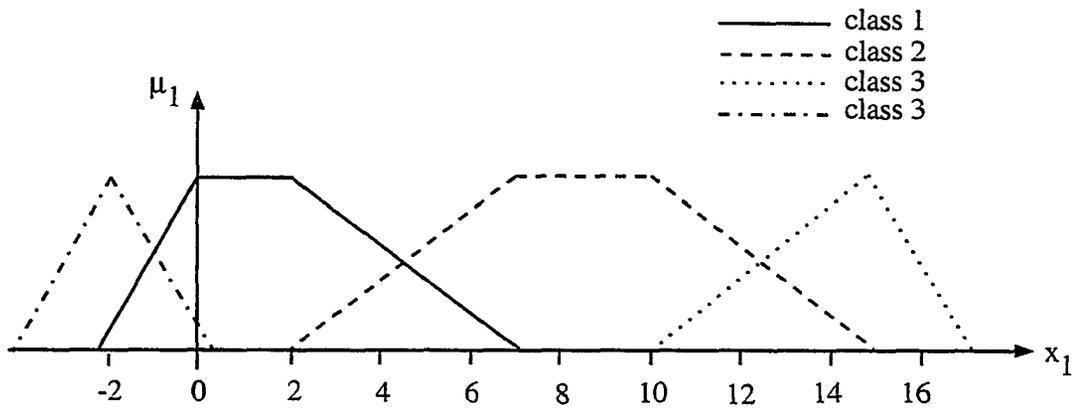
An example of establishment for an optimal rule base from a set of simulated training data will now be given. The training feature values and the desired output are listed in the Table 5.2. There are three features in each feature vector, marked as  $x_1$ ,  $x_2$ , and  $x_3$ , respectively.

From Table 5.2, it is easy to see that there are three classes (column four), three features (column one through three), and nine data points in each feature column. The nine feature values for each feature column are constructed into nine feature vectors, with each vector containing three feature elements. The feature vectors for class one are

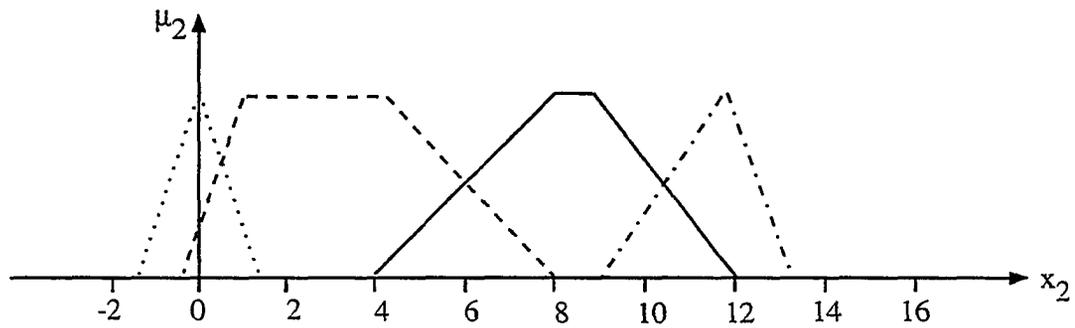
Table 5.2: Simulated training data and the desired output classes

Feature 1 ( $x_1$ )	Feature 2 ( $x_2$ )	Feature 3 ( $x_3$ )	Desired class
1	8	3	1
8	2	6	2
2	8	4	1
0	9	3	1
9	1	7	2
10	3	8	2
15	0	10	3
-2	12	11	3
7	4	7	2

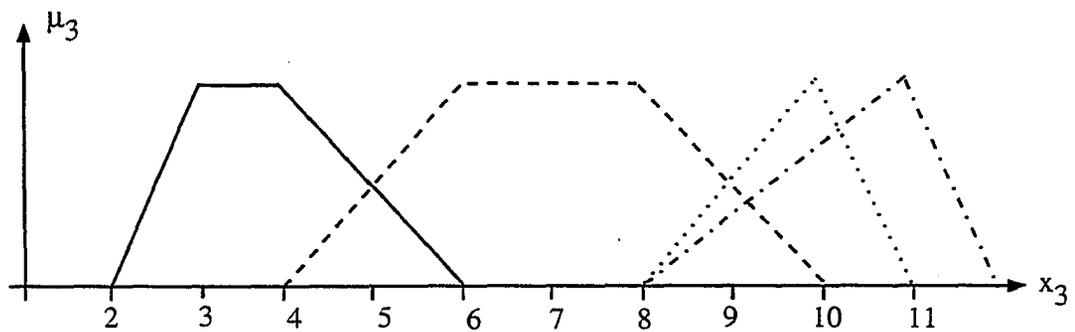
close to each other, as well as the feature vectors for class two. Therefore, the three feature vectors ((1 8 3), (2 8 4) and (0 9 3)) for class one become one cluster, and the four feature vectors ((8, 2, 6), (9, 1, 7), (10, 3, 8) and (7, 4, 7)) for class two become another cluster in a 3-D feature space. The rest two feature vectors for class 3, are not close to each other because their values for feature 1 (15 and -2) and feature 2 (0 and 2) are separated by feature values belonging to other classes. Therefore, the two feature vectors for class 3 form two clusters (actually two points in this case) in the feature space and there are totally four clusters for the data. The default minimum range values for each feature is computed by dividing the dynamic range of the feature (17 for feature 1, 12 for feature 2, and 8 for feature 3) by the total number of clusters (4). Half of the default range for features  $x_1$ ,  $x_2$ , and  $x_3$  are calculated accordingly as 2.1, 1.5 and 1.0, respectively. Since the seven data points for class 1 and class 2 (three feature vectors for class 1 and four feature vectors for class 2) form one cluster for each class, these seven data points are constructed into two rules as plotted in Figure 5.7. Note the feature values for class 3 (in



(a) Membership function of feature one



(b) Membership function of feature two



(c) Membership function of feature three

Figure 5.7: Establish the optimal fuzzy rule base from the training data

Table 5.2), the two values for feature 3 are close to each other, but the values for feature 1 and feature 2 are separated by features belonging to other classes. Therefore, two separate rules are generated for class 3 as shown in Figure 5.7. The central (b and c) parameters for the membership function are determined by the minimum and the maximum values for the features in the cluster for feature 1, feature 2 and feature 3, respectively. The edge (a and d) parameters are determined by the maximum value between the distance from the nearest other class feature to b or c, and half of the minimum default range value calculated previously.

### Output fuzzy membership

The output membership of the  $j$ th rule, given feature membership  $\mu_{1j}$ ,  $\mu_{2j}$ ,  $\mu_{3j}$ , etc. is determined by the distance measure  $D_j$  in a fuzzy hypercube constructed by the membership values of all features. An example is given in Figure 5.8 in which three features are involved. The output membership is determined by the Euclidean distance between the maximum point (where all feature memberships equal to one) and the data point for the  $j$ th rule, which is given by

$$D_j = \sqrt{(1 - \mu_{1j})^2 + (1 - \mu_{2j})^2 + (1 - \mu_{3j})^2}. \quad (5.14)$$

The general form of Equation (5.14) is given by

$$D_j = \sqrt{\sum_{i=1}^N (1 - \mu_{ij})^2}. \quad (5.15)$$

where  $i$  and  $j$  are indices for feature and rule, respectively.

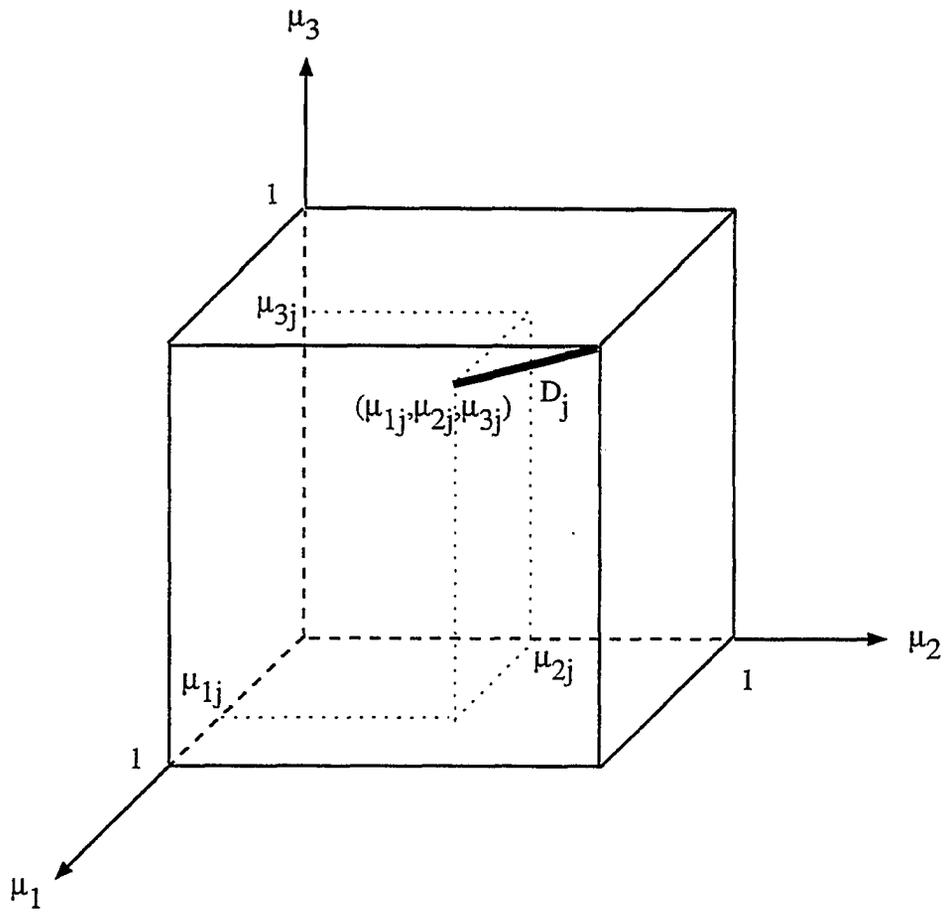


Figure 5.8: The feature distance in a 3-D fuzzy hyper cube

The output membership of the each rule is decided by the Euclidean distance of Equation (5.15). This is given by the triangular form

$$\mu_j^o = 1 - \frac{D_j}{\sqrt{N}} = 1 - \sqrt{\frac{\sum_{i=1}^N (1 - \mu_{ij})^2}{N}}. \quad (5.16)$$

The output membership value  $\mu_j^o$  in Equation (5.16) indicates the degree of truth for the  $j$ th rule.

### System output and defuzzification

Once the fuzzy rule base of a system is established for every input feature vector in the system, it gives an output score, which is the output fuzzy membership value for the input feature vector for every rule in the rule base. The centroid defuzzification is one way to average the scores and provide a crisp system output. However, to effectively use the centroid approach as we did for the adaptive fuzzy system described in earlier sections, the effective range of the output membership functions must be very narrow (as they were for multiplication inference scheme) and close to be a delta function. Otherwise, the various label values for the different classes used in Equation (5.9) would give biased weights for different classes, and the system output would therefore be biased as well. Since the output fuzzy membership function obtained from the distance measure in the fuzzy hypercube is no longer a narrow function, an approach other than the centroid must be used in the defuzzification scheme. Here, an approach similar to the nearest neighbor approach in pattern recognition is used. Each test feature vector to be classified has an output score for each rule in the knowledge base. By testing the feature vector against all the rules in the knowledge rule base and comparing the output scores (output

memberships) of all the classes involved in the rule base, we will be able to find a class for which the input feature vector scores the highest. That is, the class for a rule that the input feature vector obtains the highest output membership value. The desired output label value for that class will then be assigned as the system output for that particular input vector. The value of the output membership, which is the highest for all classes, is also obtained to provide a correctness measure for this classification.

The fuzzy rule base established from the training data gives a direct link between the input feature vector and the output of the classifier. After the training process, any input feature vector extracted from a test image will have an output value given by the classifier. An image is classified pixel-by-pixel, and areas that belong to different classes will be eventually marked by different label values.

### **Correctness measure for image classification**

After the completion of image segmentation, an image is divided into different regions with each region belonging to a certain class labeled by the label value of that class. We would like to know how well the classification is performed. For example, we want to know whether or not the pixels classified as class  $k$  truly belong to that class, or in a fuzzy measure, to what degree they belong to class  $k$ . The output fuzzy membership value we obtained for each pixel in the defuzzification process gives the degree of truth for the classification of that pixel. An average over all pixels classified into the same class will give the degree of truth for the classification of that class. This average value, which is calculated over all pixels belonging to the same class  $k$ , measures the degree of truth for the classification of class  $k$ . If there are totally  $K$  classes in the output image, there are  $K$  values associated with the output image, each one gives a correctness meas-

ure for the classification of its related class, respectively. If there are totally  $P$  pixels classified as class  $k$ , the correctness measure for the classification for class  $k$ , denoted here as  $CM_k$ , is defined in Equation (5.17) as

$$CM_k = \frac{1}{P} \sum_{p \in k} \max_j (\mu_j^{o,p}), \quad (5.17)$$

where  $\mu_j^{o,p}$  represents the output membership for the  $j$ th rule on  $p$ th trial (for the  $p$ th pixel in the image). The correctness measure  $CM_k$ , ranging from 0 to 1, actually gives the average membership for the area being classified as class  $K$ . If the  $CM_k$  value is near 1, the classification for class  $K$  has the maximum confidence (in another words, the classification for class  $K$  is completely true). If the the  $CM_k$  value is near zero, the classification of class  $K$  is near complete false (indicating false classification). If the  $CM_k$  value is between 0 and 1, it indicates the classification is partially true, with the  $CM_k$  value representing the degree of truth for the classification of class  $K$ .

## **CHAPTER 6. INTERACTIVE HUMAN-MACHINE INTERFACE**

The combination of feature extraction and supervised classification provides a structure to perform automated image inspection. The system is a general purpose one that can be used on generally any type of images. The training process of the supervised classification actually customizes the system to a particular application and therefore achieves objective classification. Once a system is trained by the training prototypes, images with similar properties to the training data can be inspected automatically by the system without human interruption. To generate training data, to establish the classifier from training, and to execute automated image segmentation requires an interactive user interfacing with the computer. Therefore, it is necessary to build a graphic user interface to fulfill the interactive training, image processing and display. The image inspection system software and its graphic user interface are developed under a Unix and X window environment named HAPXM, which is a new version of the package named HAPPI [Brown, 1992].

### **Graphical User Interface**

The graphical user interface (GUI) of the image classification system for automatic flaw detection is based on X window (X11), and Motif for Unix workstation. X window is a common open window environment supported by almost all Unix based

systems. Motif is a toolkit library built on X11 and is becoming the most supported toolkit for X-window systems. The graphical user interface of the image classification system HAPXM supports X11 Release 4 or above and Motif 1.1 or above. Figure 6.1 shows a typical appearance of HAPXM user interface, which displays the software main window and menu bar, several image windows, the file selection dialog box, and plots for image analysis. The main window of the software (Upper left window in Figure 6.1) contains the menu system and displays instructions for each operation. The menu is a pull-down type where execution items are listed. Image files are loaded to HAPXM through a file selection dialog box (lower left window in Figure 6.1) which enables user to select and load images from disk, and to change directories. Once a file has been chosen by the user, it is loaded into HAPXM memory and is displayed on the computer screen. The software currently supports common image file formats such as PGM, SUN Raster, TIFF, and ASCII text and displays images with 256 gray tones (8-bit dynamic range).

An image processing operation basically involves two steps. The operator first chooses a menu item from the main window menu, and then uses the mouse of the computer to click inside the window of the target image to which an image processing operation is applied. After the completion of the processing, a new image will be displayed in a new window showing the result of the process. There are three image windows shown in Figure 6.1 (center column). One of the images is an X-ray radiograph of a weld (top of column), the other two images are the display of its one feature, and the result of pixel classification. The user input for image processing is usually handled by a parameter input dialog box in which a user can choose the processing parameters.

---

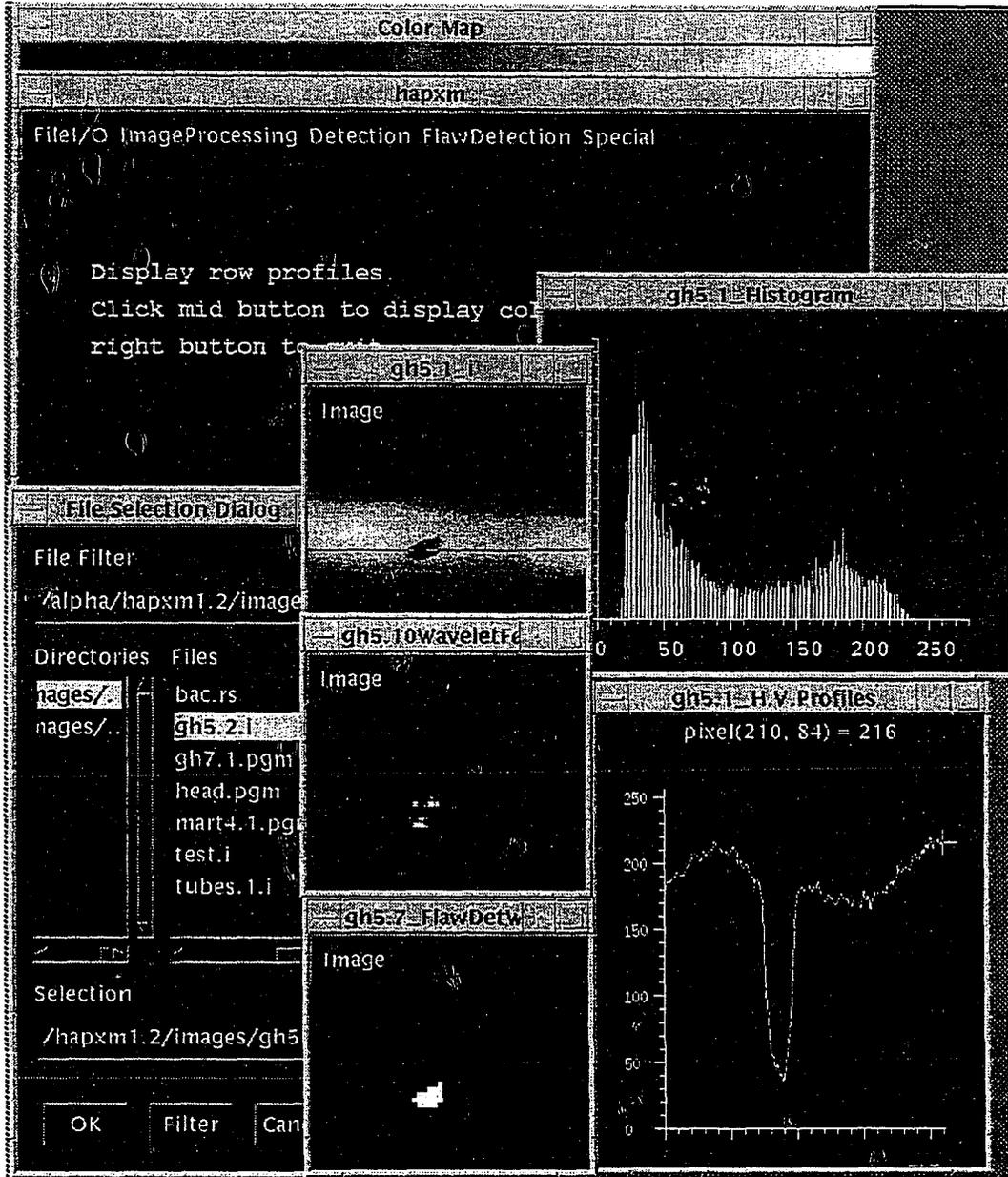


Figure 6.1: HAPXM graphical user interface

In addition to the training, classification and feature extraction functions implemented for automated image inspection, many common image processing routines are also incorporated into the software for general purpose image enhancement, analysis and manipulation. The two plots in Figure 6.1 (right side) are typical examples of image analysis for the weld image. One plot (top right) displays the histogram of the image, the other plot (bottom right) displays the row profile and pixel intensity value at the location chosen by the user. An image can also be processed before the classification operation in cases for which the original images are not feasible for direct feature extraction and classification. The operation results can be save as disk files in a common image file format such as PGM, TIFF, etc., and can also be saved as postscript files for hard copy output.

### **Generation of Training Prototypes**

The supervise fuzzy classifier provides a uniform architecture for classifying different kinds of images. A classifier is not functioning until it's been trained and equipped with the knowledge base for the application. The information used to create the knowledge base is gathered from training data. HAPXM provides easy tools for a user to generate training data from training images.

Training data are generated from training images in which flaws and object structures are known to the user. A user can use a mouse routine to click and draw lines in the training image. For example, after a line is drawn inside the training image, pixels on the lines drawn will be assigned as either the training pixels for the defects, or the training pixels for the structure background. The desired label value that marks different class areas in the output image will also be assigned. For a multiple-class classification applica-

tion, different class label values will be assigned to different prototypes by multiple clickings of the label assignment mouse button. Once a pixel is chosen as a training pixel for a certain class, a feature vector calculated near the local area of that pixel is used as the training data for features of that class. Figure 6.2 shows the diagram for the procedure of assigning training data for different classes from training images.

After the calculation of feature vectors for all the training pixels. These feature vectors are saved into a training data file along with the desired output values for their associated class. The training data file also contains other relevant information such as the file size, feature vector size, type of the features computed and size of the localization filters, etc. that will be used in the training process.

### **Machine Training and Image Inspection**

In Chapter 5, we discussed the classification engines that have all the functions to achieve supervised classification. They are the "brain" of the automated image inspection package and are established for specific problems through the training process. A classifier can not be used until it's been trained and equipped with the knowledge base for a particular application. The classifier itself is a functional device that is independent of applications, while a knowledge base provides detailed information about the image and the object to be classified. One significant advantage of this type of classifier is that it is the knowledge base, not the classifier, modified for different applications. There is no need to rewrite the software for other applications. That means, to apply the automated system to a new image classification application, only a new training process is needed for the classifier.

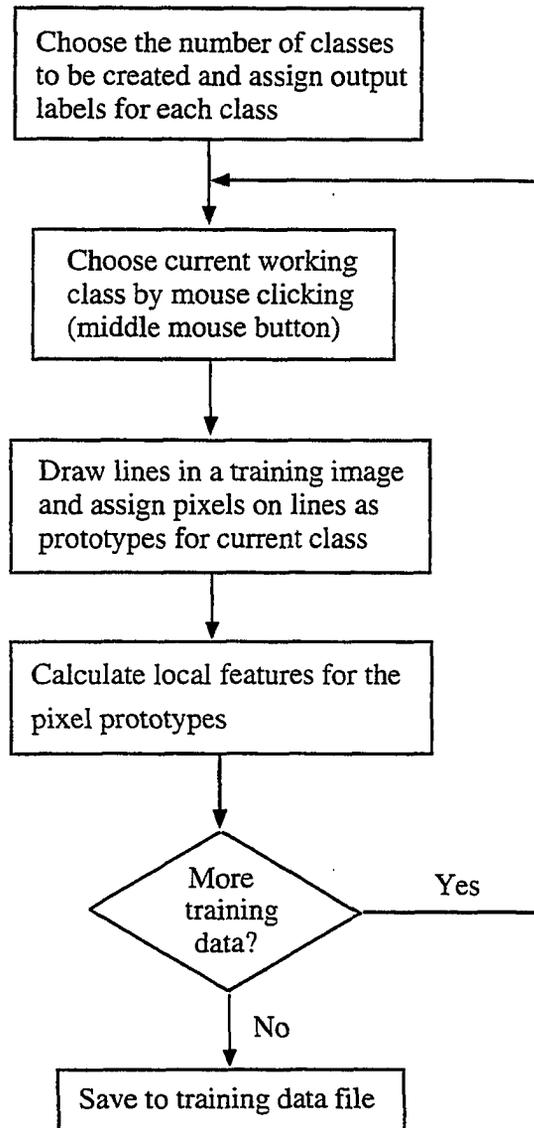


Figure 6.2: Creating the training data from training images

The training process for the computer is conducted by the human operator. Since the fuzzy classifiers constructed in Chapter 5 are initialized directly from the training data file, it takes very little time for the classifier to converge. Among the two classifiers discussed in Chapter 5, establishing the adaptive fuzzy (neural fuzzy) classifier generally takes more time than the one-step optimal classifier. For the adaptive fuzzy classifier, the rule base is constructed iteratively. For a training data file with less than 500 lines of training data, it usually takes a couple of second to establish the fuzzy rule base and complete the training process for the adaptive fuzzy classifier.

The training process actually creates a fuzzy rule base in computer memory to control the classification operation. There are several menu items in the HAPXM main menu for different training schemes. A training data file is either loaded directly into the memory without any optimization, or loaded optimized using the optimization scheme discussed in Chapter 5. A dialog box will pop up for the operator which enables him/her to input the training data file name, select classifiers and the type of membership functions for the rule base. A typical dialog box for the training process is shown in Figure 6.3, in which the adaptive fuzzy classifier and Gaussian membership function were selected.

Once the training parameters have been selected by the operator. The computer will read the training data file, create the fuzzy rule base in its memory according to a user selected inference scheme and membership type, and display a message in the main window indicating system training has been completed. After the training, the image inspection process is activated by clicking inside the image to be inspected after choosing a flaw detection menu item in the main window. The resulting image showing the labels

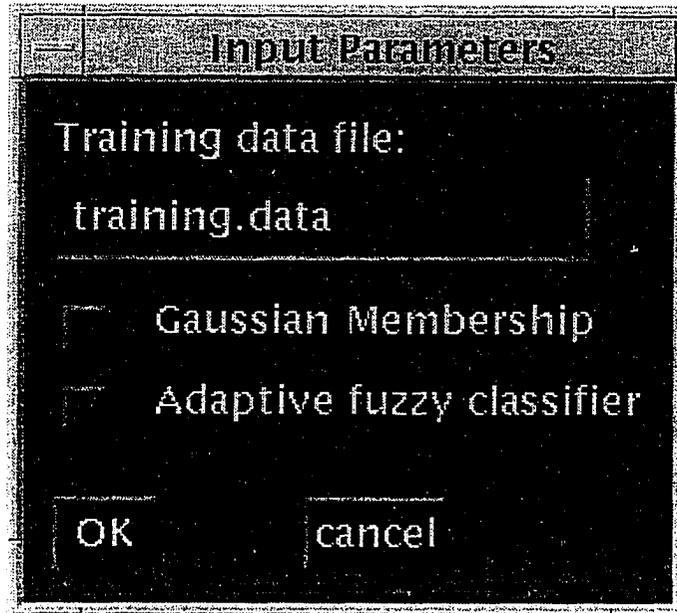


Figure 6.3: The dialog box for selecting training parameters

of different classes will be displayed on the computer screen after the classification. It is also easy to build a processing sequence after the training so that a series of similar test images can be inspected automatically without human interruption. After completion of the inspection of each image, the computer screen displays a bench mark of each operation, and the correctness measure for the classifications (i.e., the average degree of truth for the classification of every class) of every class in the output image.

## CHAPTER 7. EXPERIMENTAL RESULTS

The image classification is done pixel wise. Since features used in feature vectors are extracted within a local area in an image, it is possible to classify one out of several neighboring pixels and still detect flaws when the flaws and local filter sizes are larger than the step size (e.g., a step size = 2 means to skip one neighboring pixel). This could reduce the computation load when the resolution of the output label image is not a big concern. As a matter of fact, the finest resolution one can get from a wavelet decomposed images has a step size of two because at least one decomposition is performed to reduce image resolution by a factor of two in each direction. In our image segmentation applications, a step size of two or four is used in the classification and the output label images are usually constructed by 2 by 2, or 4 by 4 blocks.

To achieve flaw detection, we construct a feature vector for each pixel to be classified following procedures discussed in Chapter 4. Pixels are classified pixel-by-pixel in an image with a fixed step size. A user can choose the total number of features to be used in a feature vectors before the operation. The training process is conducted through the user's input on the screen. The image inspection software runs on UNIX platforms and use X window/Motif graphical user interface (see Chapter 6). An operator uses the computer mouse to draw lines inside the test images that indicate flaw regions and back-

ground regions. The feature vectors of the pixels chosen by the user are computed and stored, together with the corresponding desired output labels in a training data file. As discussed briefly in the previous chapter, when the training process begins, the program reads in the training data from the training data file and updates the fuzzy membership functions for the rule base. This is actually the process of rule base construction according to the numerical training data and optimization criteria. After a training process, a fuzzy rule base, which directly controls the performance of an image classification operation, is loaded into system memory and the classifier is ready to operate. No linguistic rule has been used in the experiments. This is mainly due to the fact that features are extracted directly from images and the application is therefore data driven. But linguistic rules can be added easily because they have the same forms as the rules generated from the numerical data.

### **The Detection of Flaws in Aluminum Castings**

Figure 7.1 displays a size 128 by 128 digital X-ray radiograph for an aluminum casting with complicated geometric structure. The image was generated by an X-ray simulation software package called XRSIM, developed by the X-ray group in Center for Nondestructive Evaluation, Iowa State University [Xu *et al.*, 1994]. The image was created with a simulated generator voltage = 60 KV, current = 1.0 mA, and exposure time = 4.0 sec. under default white noise condition ( $sdv = 1$ ). Figure 7.2 displays a casting image with an elliptical void-type flaw with a size of 0.4 cm, 0.05 cm and 0.3 cm in x (horizontal), y (vertical) and z (parallel to X-ray beam) directions, respectively. Four features displayed in Figure 7.3 were extracted from Figure 7.2 using 2-D DAUB4 trans-

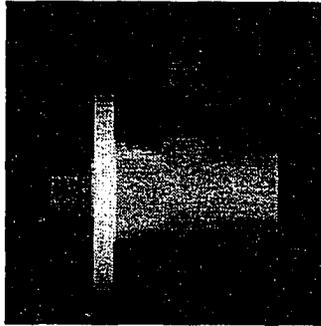


Figure 7.1: The X-ray image of an aluminum casting

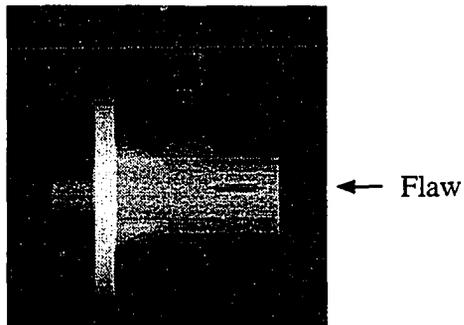
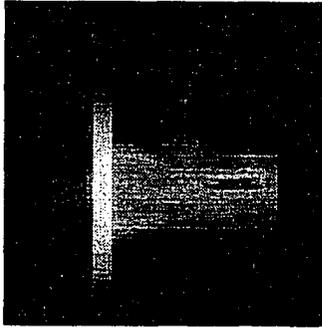
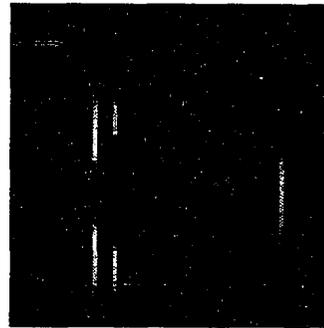


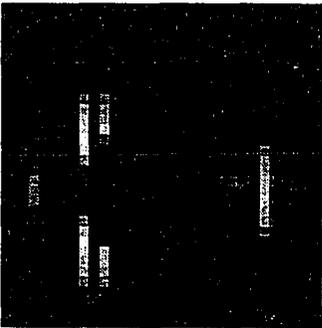
Figure 7.2: A void flaw in the aluminum casting of Figure 7.1



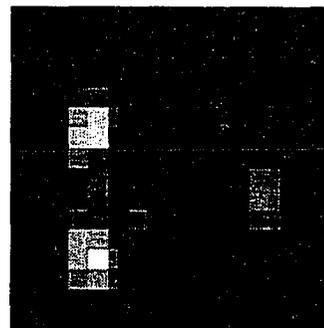
(a) The average gray-tone feature



(b) The first level decomposed feature



(c) The 2nd level decomposed feature



(d) The third level decomposed feature

Figure 7.3: The four features extracted from wavelet decompositions of Figure 7.2

form for 128 by 128 array, and were used to construct the feature vector used for pixel classification. One of the features is a smoothed average gray-tone feature, the other three are decomposed local harmonic features at the first, second, and third level of decompositions. The four features displayed in Figure 7.3 were used as the four feature elements in the feature vector associated with each pixel in the image of Figure 7.2. For example, the feature vector for the left corner pixel in Figure 7.2 was constructed by using the left corner feature values displayed in Figures 7.3 (a), (b), (c), and (d) as its feature elements. The feature vectors for other pixels in Figure 7.2 were also constructed by using the feature values collected in the corresponding locations in features images in Figure 7.3. The feature vector for each pixel to be classifier was then input to the trapezoidal optimal fuzzy classifier discussed in Chapter 6 for pixelwise image classification. The background training data for this classifier were generated from features extracted for every pixel of Figure 7.1 which does not contain any flaw. The training data for the void flaw area were created by clicking the computer mouse inside the flaw area in Figure 7.2 and computing features near the clicked points. The desired output label value for a background pixel was assigned as 50, and the desired output label value for a flaw pixel was assigned as 250. The training data file contained 16484 data points in which more than half were redundant. The optimization process for the training operation scanned through all the data points and combined all the training data into 226 rules within 5 sec. on a DEC 5000/200 workstation. Therefore, the optimization realized a reduction of over 98% on the size of fuzzy rule base and hence improved the speed of classification by the same factor. The feature vectors constructed for the image in Figure 7.2 were sent to the classifier that labeled the pixel to which feature vector was associated.

The result of pixel classification, applied to every other pixel, is displayed in Figure 7.4, in which flaw pixels are marked in white and background pixels are marked in black. The light edge frame of the casting was later superimposed to the classified image as a boundary reference. The classification process of this 128 by 128 image used 20 second of time on a DEC 5000/200. The correctness measure (Equation (5.17)) was 0.959 for the classification of the flaw, and was 0.994 for the classification of the background, which indicated very high confidence level because the classifications for the flaw and background regions were near complete truth. The speed of a classification operation depends largely on the size of the rule base because every rule in the knowledge base has to be checked before a decision can be made. Classification with a smaller rule base increases the speed of the process.

After a classifier has been trained, the rule base is established for the type of application specified by the training data file, and the classifier is ready to examine similar images without further training. Figure 7.5(a) shows an X-ray radiograph of the aluminum casting of Figure 7.1, with a void flaw about half of the size of that in Figure 7.2.

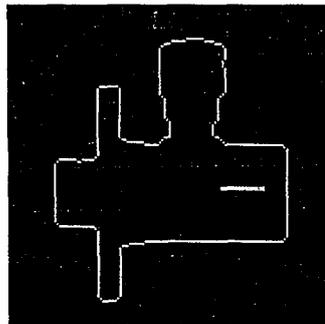


Figure 7.4: The image segmentation result of Figure 7.2

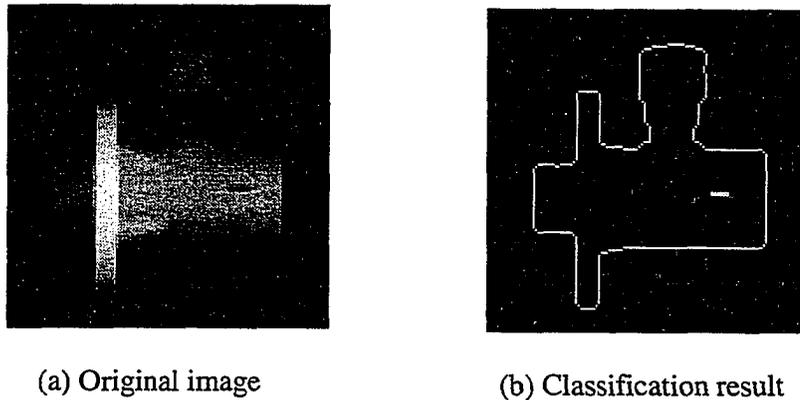


Figure 7.5: The detection of a smaller flaw

Using the classifier trained in last example (Figure 7.2), the flaw in Figure 7.4(a) was successfully detected. The classification result, superimposed on the casting boundary image, is displayed in Figure 7.5(b). The correctness measure for successful flaw classification is 0.814, and the correctness measure for background classification is 0.994. The correctness measure for flaw detection in this case is lower than that of the previous case (Figure 7.2), which indicates less confidence in flaw detection. This is understandable because all the flaw prototypes used in the training process were collected from Figure 7.2. Since the correctness measure of 0.814 is still near 1, the result of flaw detection is most likely true.

Figure 7.6 shows another example of image inspection using the classifier trained by flaw prototypes of Figure 7.2. The classifier is used to detect similar flaws at another location in the casting. In Figure 7.6(a), a smaller flaw is located near a joint between the

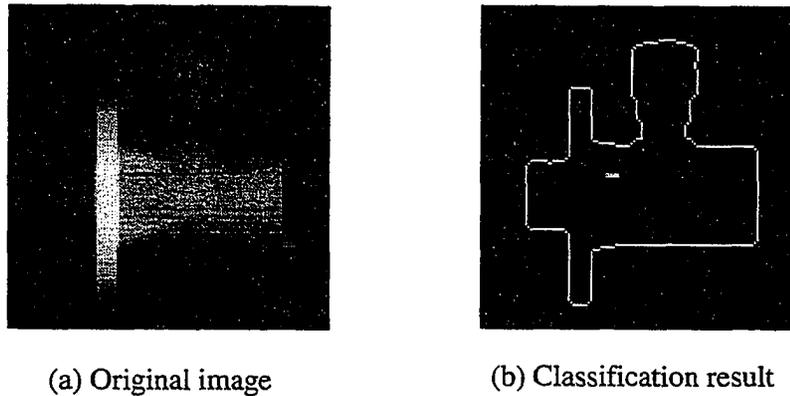


Figure 7.6: The detection of a flaw at another location

cylindrical body of the casting and the "arm" of a circular disk connected to the cylinder. It is a place crack-type flaws are likely to appear in a real part. The size of the flaw in Figure 7.6(a) is 0.2 cm, 0.3 cm, and 0.2 cm in x, y, and z directions. The image in Figure 7.6(a) was classified by the same classifier trained previously by flaw prototypes in Figure 7.2 and background prototypes of Figure 7.1. The classification result of Figure 7.6(a) is shown in Figure 7.6(b). Similar to previous examples the flaw is marked by high intensity pixels in the output image along with the reference boundary of the object. The correctness measure for flaw detection is 0.868, which is still a good indicator for a correct detection.

Regions in an image can be classified into multiple classes, provided that the classifier is properly trained by prototypes of different classes. Figure 7.7(a) shows the same simulated aluminum casting with two different kinds of flaw. There is a void flaw (0.2 x 0.5 x 0.3 cm in size) and a steel inclusion flaw (0.2 x 0.3 x 0.2 cm in size) on the cylin-

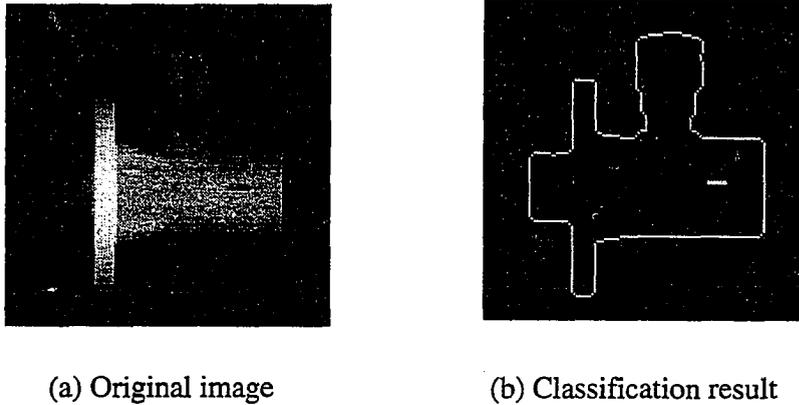


Figure 7.7: The detection of different types of flaws

dricul body of the object. The inclusion flaw shows higher intensity (whiter) in the radiograph because of less X-ray penetration through the dense steel material. By adding a couple of pixel prototypes of the inclusion flaw into the training data file created in the previous example, and repeating the training process for the classifier, the classifier was able to distinguish the void and the inclusion by marking them with different colors (gray tones) in the output image. Figure 7.7(b) show the classification result of Figure 7.7(a), in which the void flaw is marked as light gray, the inclusion flaw is marked by a dark gray color, and the reference boundary of the object in the output image is marked by white. The correctness measures for classifications of the void, the steel inclusion, and the background are 0.814, 0.759, and 0.993, respectively. Each one indicates the confidence level (the degree of truth) for the correct classification.

### The Detection of Flaws in Welds

In addition to the inspections of XRSIM simulated radiographs, a set of real X-ray radiographs of weld were also inspected. Unlike the casting situation which provided a complete training prototype for the image background, in some applications the training from prototypes may not provide complete information. Figures 7.8, 7.9 and 7.10 are radiographs of welds with gas hole (void) flaws. Training data for classification of these images were generated from Figure 7.8, in which background training data were generated from pixels on several lines drawn across the background of the image (without flaw) and the flaw prototypes were generated from several pixels in the flaw region in the image. The training data generated consisted of 319 data points.

Since image backgrounds are not the same for different weld radiographs, and only a small portion of the pixels in Figure 7.8 were used as training data, the training data was incomplete for both background and flaw classifications. It is very common in real world applications that a complete training data set is not always available. A good classifier should be able to correctly "guess" the result based on existing information. By using the optimized fuzzy classifier discussed in the last section of Chapter 5, the fuzzy rule base used to control the classification was established. The 319 data points in the training data file were combined into 77 rules in the fuzzy rule base, which is a reduction of over 75% of training data. The training process only takes 0.0094 second on a DEC 5000/200. The classification results for Figures 7.8, 7.9 and 7.10 are shown in Figures 7.11, 7.12, and 7.13, respectively, in which flaws are marked by white (high intensity) pixels, background are marked by black pixels, and pixels that do not have significant membership for either the flaw or the background classes are marked in gray to indicate

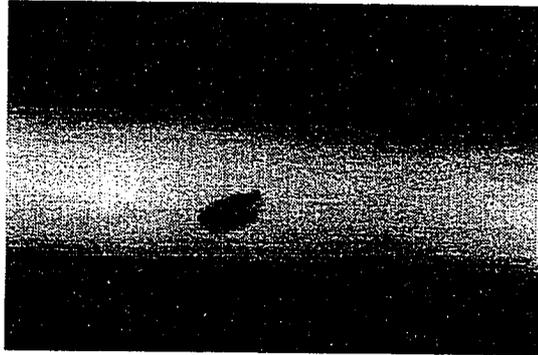


Figure 7.8: A gas hole flaw in weld, example 1

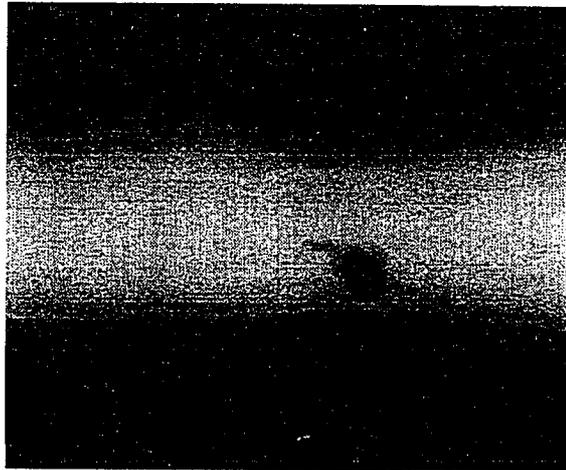


Figure 7.9: A gas hole flaw in weld, example 2

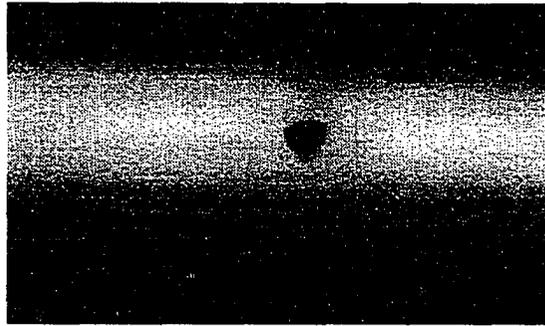


Figure 7.10: A gas hole flaw in weld, example 3

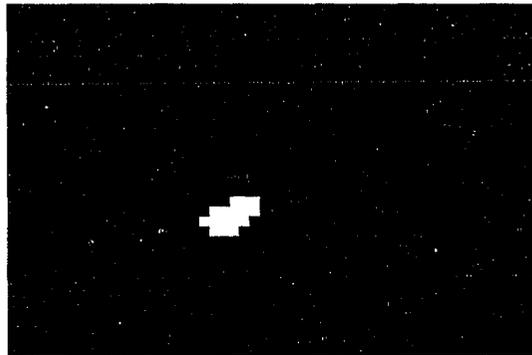


Figure 7.11: The optimized classification of Figure 7.8

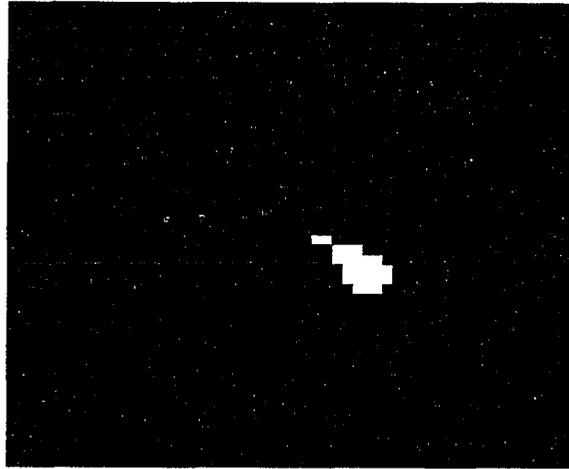


Figure 7.12: The optimized classification of Figure 7.9

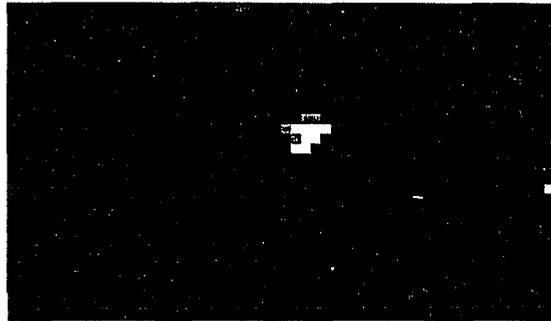


Figure 7.13: The optimized classification of Figure 7.10

used for the classification of Figures 7.8, 7.9 and 7.10 are 2.57 sec., 3.53 sec., and 2.39 sec., respectively. In Figure 7.11, the correctness measures for the classification of flaw and background are 0.741, and 0.943, respectively. In Figure 7.12, the correctness measures are 0.726 and 0.864, and in Figure 7.13, the correctness measures are 0.672 and 0.864 for the classification for flaw and background, respectively. The correctness measures are lower than previous applications (e.g., the aluminum casting application) mainly due to the fact that we have incomplete training data for both the flaw and the background, and the classifier was "guessing" the result for the missing data.

Figures 7.14, 7.15, and 7.16 are the classification results for the same weld images using the adaptive fuzzy classifier. The same training data file was used to train the adaptive system. All the prototyping data points in the training data file were used to establish the fuzzy rule base. Therefore, there are totally 319 rules in the fuzzy rule base after the iterative training process. The training process for the adaptive fuzzy system took about 9.1 seconds, and the classifications for Figures 7.12, 7.13, and 7.14 using the adaptive fuzzy classifier took 36.6, 54.1, and 35.2 seconds, respectively. From the results of the adaptive classification, one can see there are some false classifications for the background pixels, especially in Figure 7.16. This is due to the fact that the training data is not complete for all the images. The training data were generated from Figure 7.8 and were used as prototypes for all three weld images. Since the adaptive fuzzy classifier uses a multiplication scheme to generate the output membership function for each rule, the effective range of the output membership function is very narrow and therefore, less tolerant for incomplete training data. In another words, the ability of "guessing" for the adaptive fuzzy classifier is poorer than that of the optimal fuzzy classifier.

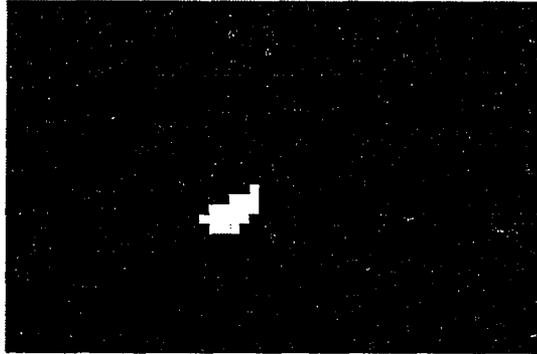


Figure 7.14: The adaptive fuzzy classification of Figure 7.8

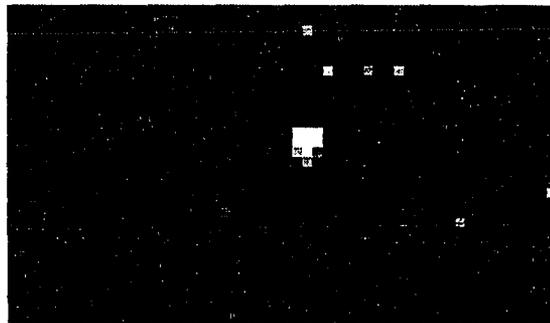


Figure 7.15: The adaptive fuzzy classification of Figure 7.9

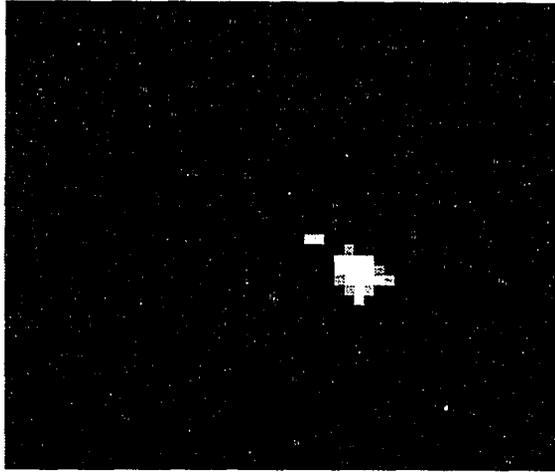


Figure 7.16: The adaptive fuzzy classification of Figure 7.10

uses a multiplication scheme to generate the output membership function for each rule, the effective range of the output membership function is very narrow and therefore, less tolerant for incomplete training data. In another words, the ability of "guessing" for the adaptive fuzzy classifier is poorer than that of the optimal fuzzy classifier.

Because the adaptive fuzzy classifier discussed above directly loads training data into the system, the size of the fuzzy rule base is the same as the size of the training data. Therefore the speed of classification is lower than that of an optimized classifier. In Chapter 5, it was mentioned that one can reduce the rule-base size for an adaptive fuzzy classifier by throwing away redundant training data, and hence reduce the size of fuzzy rule base. However, this process is at the cost of losing information, and the appropriate threshold value for data redundancy is dependent on the application on hand. A user has to experiment with different thresholds for a specific application in order to determine a

good threshold value. If the threshold is set too high there will not be too much data reduction in the process. On the other hand, if the threshold is set too low, a lot of information will be lost during the process. Figure 7.17 shows the adaptive fuzzy classification for Figure 7.8 after discarding 75% of the training data in order to achieve the same rule-base reduction rate as that of the optimal classifier. The execution time was reduced to 11 seconds for the classification of the entire image, but pixels near the boundary of the weld were falsely classified as flaws because the distinct information about the weld boundary pixels was lost during the data reduction process.

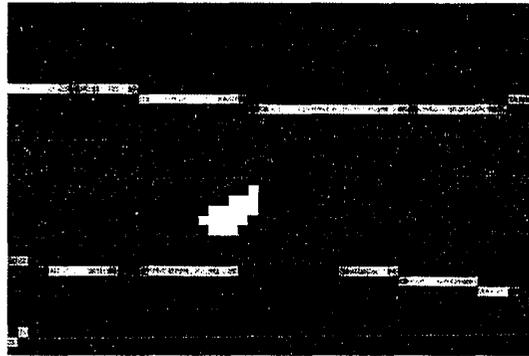


Figure 7.17: The adaptive fuzzy classification using 25% of the training data

In addition to the detection of void-type flaws in weld, the algorithm was also tested on other types of flaws in weld images. Figure 7.18 shows a weld image with an inclusion flaw that has a higher intensity than the weld background, and the detection result by the optimized fuzzy classifier is shown in Figure 7.19 in which the flaw is marked in white and the background is in black. Again, the result is displayed in 4 x 4



Figure 7.18: An inclusion flaw in weld

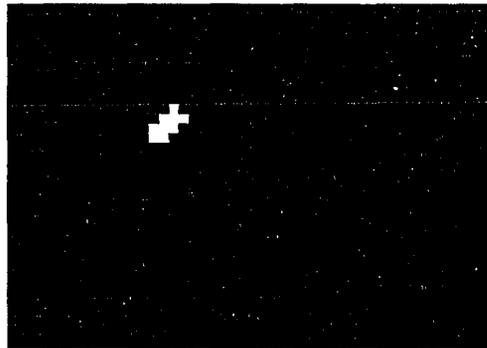


Figure 7.19: The optimal classification of Figure 7.18

blocks. The inclusion flaw is easier to detect because the intensity values of flaw pixels are different from those of the background pixels.

Figure 7.20 shows an example of crack-type flaw in weld, in which the crack is located near the center area of the image. The result of the classification is shown in Figure 7.21 in which the crack detected is marked in white.

Figure 7.22 shows an an under-cut flaw (the dark belt near the bottom of the white band of weld) lying near the edge of the weld. Its classification result is shown in Figure 7.23 where the flaw area is marked in white. Because the flaw area does not have a sharp contrast from the background, some of the background pixels on the edge of the weld were also classified as flaws and therefore caused false alarm.

In addition to the weld images tested previously, the flaw detection algorithm was also applied on a number of images with various flaw types. The detection results are listed in Table 7.1, in which the first column indicates the type of flaw tested, the second column lists the total number of images (each image contains one flaw) tested for each specific type of flaws, the third column shows the total flaws detected, the fourth column shows the flaws that were missed by the detector, and the last column displays the number of false alarm occurred in the flaw detection process. From the experimental data listed in the Table 7.1, there are 6 images containing void-type flaws and 9 images containing inclusion flaws. All the voids and inclusion flaws were successfully detected without false alarm. This is due to the fact that the void or inclusion flaws generally have sharp local transitions for the flaw regions, while the wavelet features are very sensitive to local transitions and therefore provide distinct features for the flaw. Among the 6 crack-type flaws tested, 5 were detected, one was not detected. There were also two false

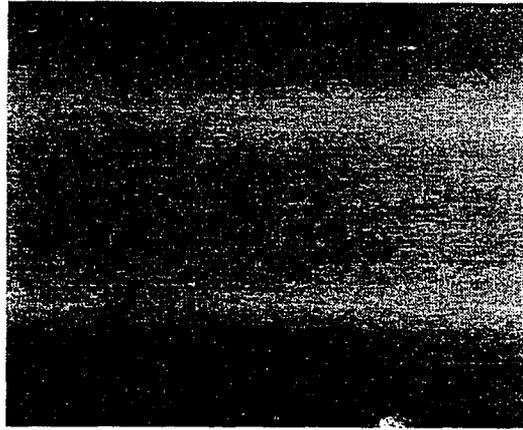


Figure 7.20: A crack flaw in weld

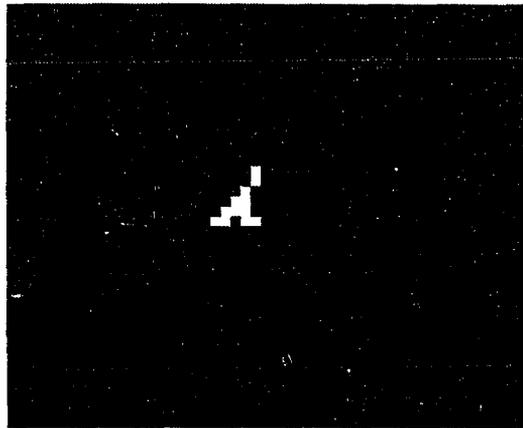


Figure 7.21: The optimal classification of Figure 7.20

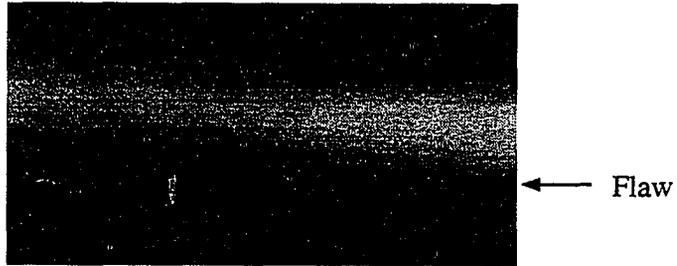


Figure 7.22: An under-cut flaw in weld

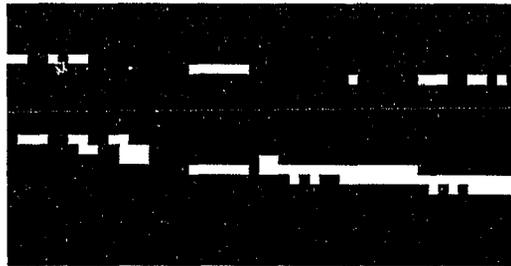


Figure 7.23: The optimal classification of Figure 7.22

Table 7.1: The detection of various flaws in weld

Type of flaw	The number of flaws	detected	missed	False alarm
Gas hole	6	6	0	0
Inclusion	9	9	0	0
Crack	6	5	1	2
Enigma	3	1	2	0
Under cut	6	6	0	5

alarms. The cracks displayed in weld image were narrow dark lines (as shown in Figure 7.20). The feature vector and the classification sometimes fail to distinguish these dark lines (crack flaws) from the noisy background of the image, especially when noise is strong and the crack is small. For more globally distributed and less distinct flaws such as an under cut or an enigma, the wavelet features were less successful. This can be seen in Table 7.1, the software was able to detect 6 under-cut flaws in test images, but 5 out of 6 times false alarms appeared in the results. The wavelet features were not be able to distinguish the under-cut flaw regions (occurred near the edge of the weld) from the edge of the weld. Similar to an under-cut, an enigma flaw is also a narrow belt stretching on top of the weld in the image. It usually occurs near the central region of the weld and is wider than an under cut, but the intensity different between the weld region and the enigma region is very small and can barely be seen by a human if only local details are noticed. The wavelet features have difficulties in recognize this type of flaw. Among the

three images tested, only one flaw was detected and the other two were missing. Based on the test on multiple image with a variety of flaws, we can conclude that wavelet features are good in describing sharp and local variations, sensitive to noise, but less effective in characterizing slow and global variations in the image. The overall performance statistics for flaw detection in weld images is computed from Table 7.1 as: the percentage for detection is 90%, the percentage for missed flaw is 10%, and the percentage for false alarm is 23.3%.

### **The Detection of Flaws in an Air Conditioner Part**

The background structure for weld in the images we previously tested is relatively simple. A flaw detection example for images with more complicated background is discussed here. The image in Figure 7.24 is a portion of the X-ray radiograph for an automobile air conditioner part. The air conditioner part is a cylindrical shaped casting in which there are complicated structural variations. The structural change inside the part causes the X-ray background to vary significantly across the image. By using three wavelet features extracted from the second and the third level of decompositions, and the optimized fuzzy rule-based classifier, we were able to detect the flaws in Figure 7.24 from the complicated image background. The classification result is shown in Figure 7.25 in which the flaw areas are marked in light intensity along with the reference boundary of the part. In the resulting image, the isolated pixel blocks indicate the voids in the part and the connected blocks form into a line across the central part of the image indicating a big crack. The training prototypes for the flaw and background pixels were collected from Figure 7.24. The resulting image is display in 4 by 4 pixel block because

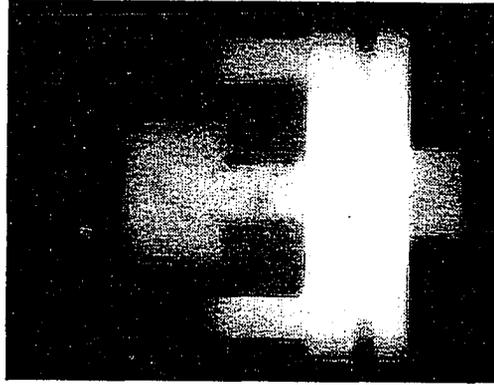


Figure 7.24: The X-ray image of an air conditioner part

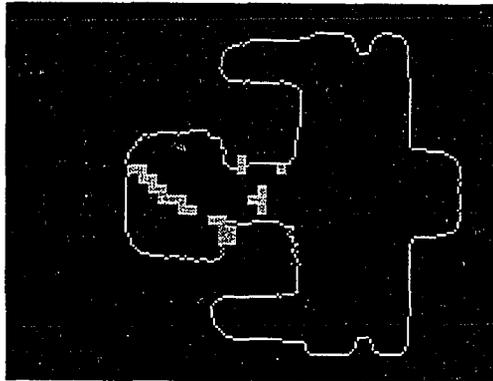


Figure 7.25: The flaw detection result of Figure 7.24

the decomposition reduces spatial resolution.

Because the thickness of the wall in the air conditioner part varies from one place to another, one X-ray exposure may not be able to display all the details in the image. For instance, in Figure 7.24, the left part of the image is over exposed because the wall in that area is thinner than other places. Therefore, the details for that area can not be easily seen in the original radiograph. To improve the performance of the automated inspection, it is sometimes necessary to apply image processing routines before the flaw detection. The purpose of the preprocessing is to enhance the quality of the image and therefore make machine inspection more straight forward. By applying contrast stretching to Figure 7.24, we are able to display details within one image for areas with different material thickness. Figure 7.26 shows the enhanced image after intensity stretching. Its corresponding intensity stretching transfer curve is plotted in Figure 7.27, in which the horizontal axis indicates the image intensity for the original image and the vertical axis indicates the stretched intensity for the enhanced image on an intensity scale from 0 to 255.

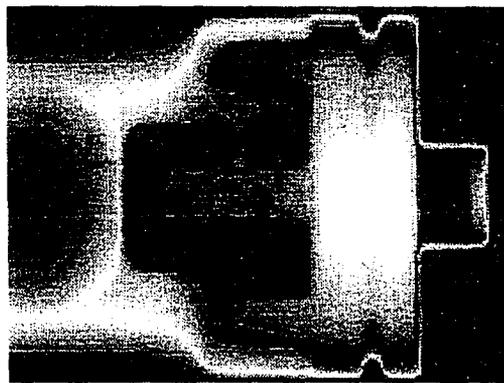


Figure 7.26: The intensity stretching of Figure 7.24

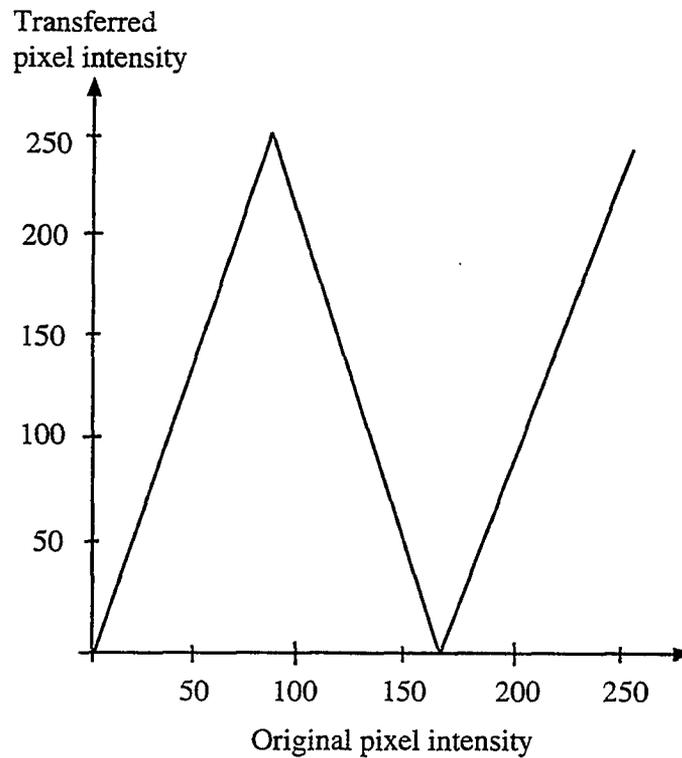


Figure 7.27: The transfer function for intensity stretching

The three-way contrast stretching shown in Figure 7.27 expands the dynamic range of low (0 to 85), medium (86 to 170) and high intensity (171 to 255) segments in the original image into full range (0 to 255), and therefore enhance the contrast for each intensity segment. The display for the medium range is reversed after the stretching in order to keep a smooth transition at the joints of the intensity segments. This contrast stretching strategy is similar to combining three different X-ray exposures into a single image. The classification result for the enhanced image of Figure 7.26 is shown in Figure 7.28, in which flaw regions are marked in white and are superimposed with the reference

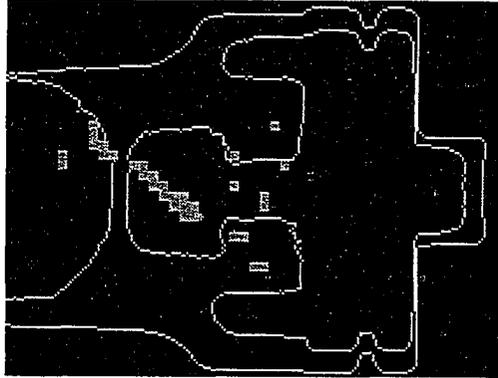


Figure 7.28: The flaw detection result of Figure 7.26

boundary of the air conditioner part. The extension of the crack (into the over exposed dark region on left side of Figure 7.24) that had not been detected previously in Figure 7.25 was detected after the enhancement. Several void flaws in the image were also detected. There is no artifact appears in the output image.

### **Performance Measure for Image Classification**

As mentioned previously, an optimized fuzzy classifier combines multiple training data points into fewer rules without losing training information. This is important for classification applications because algorithm optimization improves the speed and the performance. In this section, a comparison between the optimized classifier and the adaptive fuzzy classifier are made in terms of the execution time and training time. The comparison is based on the classification results for Figure 7.8 on a DEC 5000/200 system. As plotted in Figure 7.29, the execution time for the optimal fuzzy classification,

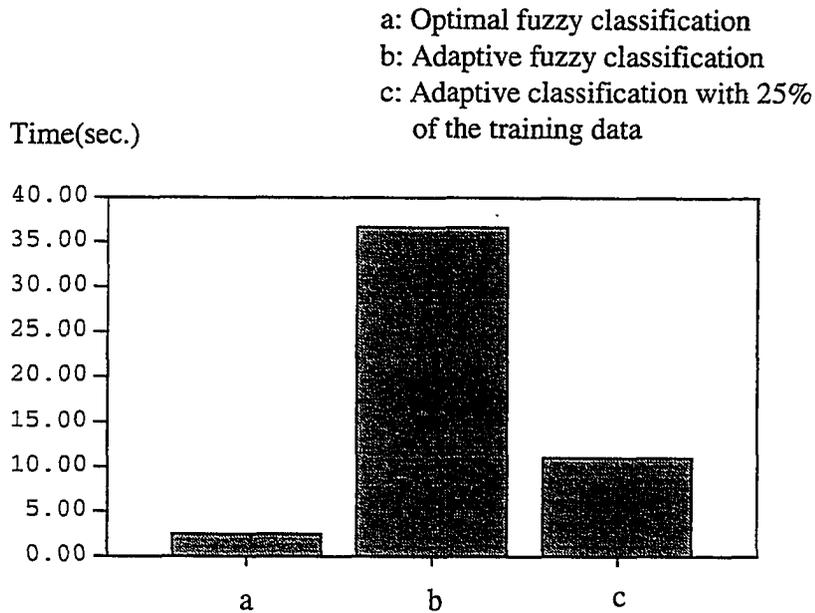


Figure 7.29: The classification time for different fuzzy classifiers

for the optimal fuzzy classification, the adaptive fuzzy classification and the adaptive classification with selected 25% of training data is 2.57 sec., 36.6 sec., and 11.0 sec., respectively.

The training process for a classifier establishes the fuzzy rule base that controls the behavior of the classifier from the training data. As shown in Figure 7.30, for the same training data file, the time used to train the optimal fuzzy classifier, the adaptive fuzzy classifier and the adaptive classifier with selective data is 0.094 sec., 9.1 sec., and 3.9 sec., respectively. Based on the comparison of training and execution speeds, the optimized fuzzy classifier clearly outperforms the adaptive fuzzy classifier. The reason is

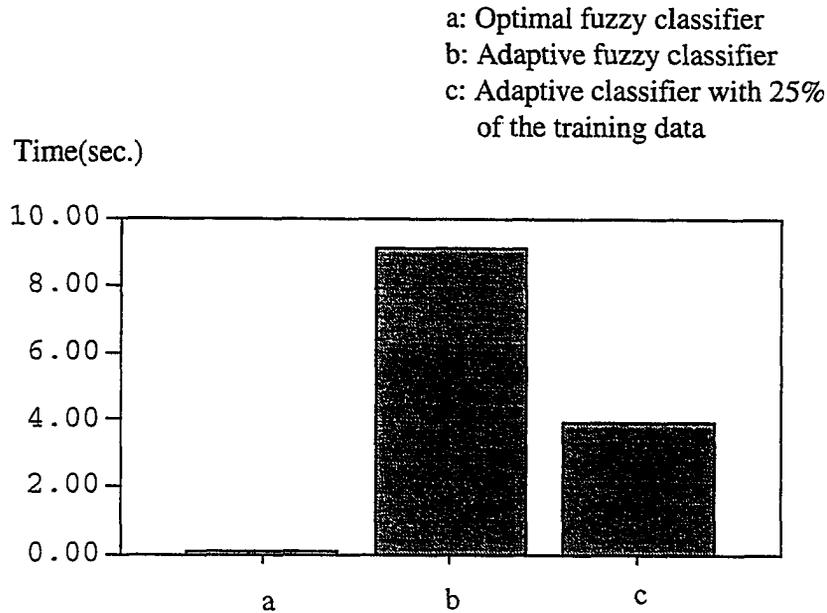


Figure 7.30: The training time for different fuzzy classifiers

that an adaptive fuzzy classifier is a continuous nonlinear mapping function. Even though the output for a fuzzy classifier has quantized values, the adaptive fuzzy system still fits the system transfer function into a continuous curve. The optimal fuzzy classifier, however, takes the advantage of the quantized output for the classifier. If there are fewer output class (e.g., two classes for flaw and non-flaw result in two output values), fewer rules are needed for the classifier and therefore the classifier has faster speed. For a more complicated classification problem which may involve many classes (result in many output values for the classifier), the speed for the adaptive fuzzy classifier will not change because it still fits the system function as a curve. However, the speed for the

optimal fuzzy classifier will be much slower than that of a two-class case because the number of rules are increased. Therefore, it is reasonable to say that an adaptive fuzzy system is more suitable for a curve fitting problems that have many output values, while the optimal fuzzy classifier is more suitable for classification applications that only have fewer output values.

Since the results of wavelet decomposition are similar to those of matched filter with filters being sequentially scaled wavelets, they are usually very sensitive to the corruptive noise that changes local pattern matches, especially under smaller scales. This is an integral property of the localized process. When extracting localized features in a very small region, the introduction of noise will definitely have a big impact on the outcome of the feature calculation. Fortunately, when using supervised classification, the noise factor can be incorporated in the training stage and the classifier can then be used under noisy conditions. Figure 7.31 displays the casting image of Figure 7.2 corrupted by

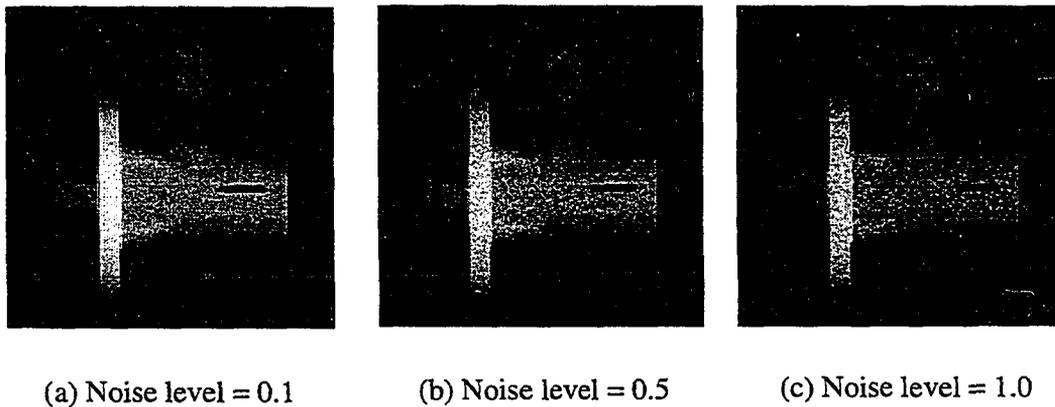


Figure 7.31: The casting image corrupted by Poisson noise

simulated Poisson-type noise. Poisson noise is the noise type that is most representative to the photon noise in X-ray radiographs. Figure 7.31 (a) shows the image corrupted by poisson noise with a variance equal to 0.1 of the signal intensity. In Figure 7.31 (b) the noise variance is half of the image intensity and in Figure 7.31 (c) the noise variance is equal to image signal intensity. It can be seen from the noise corrupted image Figure 7.31 (c) that a lot of the background pixels are less distinguishable from those of the flaw because of the corruption of the noise.

If the noisy images are classified by a classifier without training knowledge of the noise, there will be false classifications in the output image because of the noise sensitivity of the feature vector. Figure 7.32 shows the classification results for the noisy images in Figure 7.31 by the classifier trained previously from Figure 7.2 which does not contain noise information. From Figures 7.32 (a) to 7.32 (c), we can see that more false classifications occur as the noise level increases.

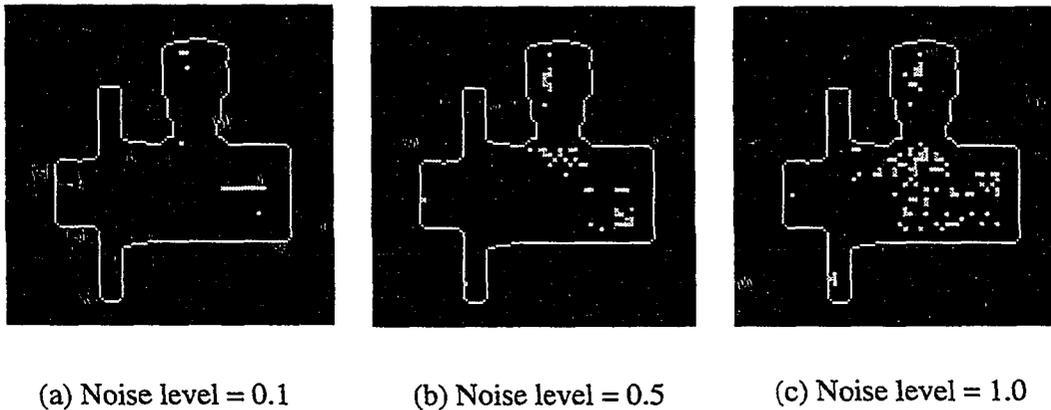


Figure 7.32: Classification without noise training

The inclusion of noise will definitely degrade the quality of the image and therefore, affect the reliability for flaw detection. A supervised classifier can overcome some of the noise problem by incorporating noise information into the training process. This is achieved by using training prototypes selected from noisy images, a classifier is then trained by flaw and background structures with the presence of noise. Figure 7.33 shows the classification results of the noisy images of Figure 7.31, after incorporating noise in the training process. There is no false classification when noise level is 0.1 and 0.5, and one false classification when noise variance is equal to the signal. This example shows the effectiveness of the training process when noise is involved.

Another factor that affects the noise issue is the resolution. The decomposition based wavelet features inherently have lower resolution than the original image. The best

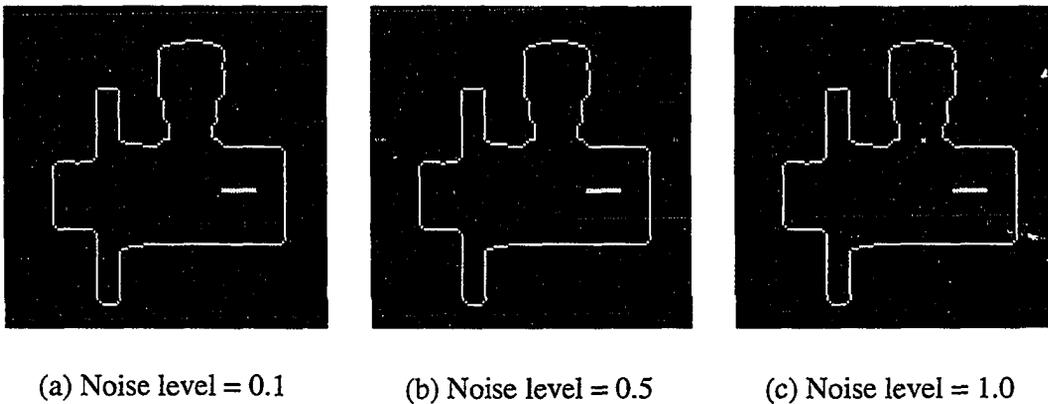


Figure 7.33: Classification with noise training

resolution in the decomposed image is 2 by 2 pixel blocks when only one level of decomposition is applied. In classification applications in which detection, rather than resolution, is the major concern, local features can be extracted from a broader area (larger scale) and hence each feature is less sensitive to the noise signal. In this case, the classification output image will have a coarse resolution as well (e.g., 4 by 4 pixel blocks). The classification speed for classifying each image will increase because only one of several neighboring pixels is classified.

## CHAPTER 8. SUMMARY AND CONCLUSIONS

This dissertation establishes a general architecture for computer vision applications. Although the original work is aimed at solving automated flaw detection problems in nondestructive evaluation applications, the algorithm itself is a general purpose one that can be applied in many other computer vision applications. One of the main contributions of this dissertation is using the wavelet transform to perform spatial spatial-frequency analysis for 2-dimensional images and extract truly localized features based on wavelet analysis. This approach is superior to existing approaches such as windowed Fourier analysis and the Gabor filtering approach because wavelet approach is truly localized and much faster. Another contribution of this work is the creation of a fast and effective supervised classifier based on the optimization of a fuzzy rule base for image classification. The issue of optimization for the fuzzy system is very important but under developed. The current research on fuzzy systems is mostly related to nonlinear mapping functions. It is generally assumed that the fuzzy rule base for the system already exists and the rules are orthogonal. In a flaw detection application, the rule base has to be established from the training data which are most likely redundant. The fuzzy rules have to be generated from the training data in order to make the classifier perform correctly. The rule base generation is very important because it is directly related to the performance of

the classifier. This dissertation discusses the training data issue and develops a new approach to optimize the rule base from the training data and therefore achieves fast training and fast classification. In addition, this completed system has a unified structure, and the training capability makes the system versatile for a lot of different applications.

Local features in an image are extracted from compactly supported wavelet decompositions. The wavelet transform decomposes a signal into multiple subband channels. Within each channel, the wavelet transform behaves like a band-pass filter. By using the wavelet transform in image feature extraction, the computer is able to understand the image through the results of band-pass filtering in the spatial frequency domain, much like what a human vision system does [Campbell and Robson, 1968; Kulikowski *et al.*, 1982; Pollen and Ronner, 1982]. Therefore, the feature extraction process develops the "eye" of the vision system. The use of fuzzy logic in the classification process makes the vision system not only look like a human, but also think like a human. Fuzzy logic models the ambiguity and uncertainty aspects in a human thinking process. The fuzzy logic and fuzzy inference scheme make up the "brain" of the vision system and make the classification task fast and reliable.

Compared to Fourier based harmonic analysis such as short time Fourier analysis and Gabor filters which are not truly localized, the compactly supported wavelet analysis has the advantage of being localized in both spatial and transformed domains. In addition to its localization property, it also gives a multiresolutional representation of the image and therefore provides contextual information about the image. Amplitude normalization is applied on wavelet decomposition in order to provide an unbiased representation toward all frequencies. Features are extracted by comparing all frequency components at a

certain decomposition level. Features extracted from the wavelet decomposition provide a step-by-step, or fine to coarse view of an image and therefore are able to give the details and the global connections of an image by using features extracted at various resolution levels. The pyramid algorithm for the discrete wavelet transform provides a unified frame for local harmonic analysis involving scaled match filtering (by wavelets), subsampling in lower frequencies and fast algorithm implementation. The features extracted from wavelet decompositions provide local and contextual information about an image at the same time and therefore give quick and complete representation of an image.

Supervised classification is used for classifying flaws in NDE images. In images with complicated background variations, the feature changes in flaw areas are usually less significant than feature differences for various background structures. Therefore, unsupervised clustering algorithms, which divide images into different clusters in the feature space, may not be able to successfully separate a flaw from the background because they give nonselective results that largely depend on the nature of the image. Supervised classification is more selective, it tells a classifier exactly what it is looking for through the training process, and the classifier behaves accordingly. Fuzzy rule-based classifiers are used in this vision system because fuzzy logic is a soft logic which is more suitable for complex applications. It is also close to the decision making nature of human beings.

Fuzzy set and fuzzy logic theories have been successfully used in the control area in the past several years and are gaining increased popularity in signal and image processing fields. Like the artificial neural network, the majority of the supervised fuzzy sys-

tems developed are nonlinear mapping functions which map input and output pairs after the adaptive training. The training processes of adaptive fuzzy systems are usually initiated from the training data and therefore can more easily achieve system convergence than a neural network. Once a fuzzy system is trained, it performs the mapping from input to the output according to the fuzzy rule base established during the training. For the nonlinear mapping function, it is generally believed that more rules result in more reliable mapping. This is understandable because more rules provide more "examples" for the mapping function to look at, and hence, more information. However, the more rules that are used in the mapping function, the more execution time is needed for getting an output. This dissertation introduces a new approach to establish the fuzzy rule base which evaluates the training data and establishes a smaller rule base by combining similar training data into a single rule. The rule base generated by the new approach provides the optimal rule base in terms of its size and effectiveness for a given problem. Unlike an artificial neural network which has a fixed number of layers and cells for all applications, the rule base for an optimal fuzzy classifier is optimized from the training data for a particular application and is customized for that application only. If there is another application with different training data, the rule based established after the new training is then optimized for the new application, and its size and rules are completely different from those for the previous application. This kind of optimization process makes the classification engine general enough for most of the applications, but specific enough for each task through customized training. Examples in image segmentation show that wavelet based local feature extraction coupled with optimal fuzzy classification can successfully detect flaws in complicated structural background. The optimal fuzzy classifier has an

---

advantage in speed and reliability over the commonly used adaptive fuzzy (neural fuzzy) classifier for flaw detection applications.

In addition to feature extraction and classification, the training process is equally important in image understanding applications. Successful classifications depend on complete and reliable training data. The graphical user interface of HAPXM provides an interactive tool for a user to generate training data from prototyping images, and to evoke different training schemes for the system. This "point and click" interface is especially important in industrial fields when the operator of the system may not be an expert in computer programming but has expertise in identifying flaws. The graphical user interface of HAPXM is built in an attempt to create a user friendly system for the manufacturing industry which incorporates the state of the art technologies developed in research institutions.

The automatic flaw detection package has been tested on various X-ray images. The experimental results show that the system is very effective if the images to be inspected have similar structural backgrounds and exposures. However, the system is less effective if the backgrounds of the test images vary from time-to-time because it would be difficult to obtain consistent training data for the background. The wavelet features are very sensitive to the local variations in an image, therefore, flaws with sharp transitions or obvious boundaries can be detected very easily. Flaws with slow transitions or without obvious boundaries are more difficult to detect because more decompositions are needed to extract global features and there is a trade off between the number of decompositions and the spatial resolution. To extract more global features from the wavelet transform requires higher levels of decompositions, this inevitably degrades the spatial

---

resolution for the features extracted, and therefore, makes it harder to locate a flaw. The noise corruption is another concern. It is preferable to have the noise in the test images removed before they are examined by the automated flaw detection system. The wavelet feature extractor is very sensitive to local variations. The existence of noise will definitely change the local features extracted by the shortest wavelet. To increase noise robustness, longer wavelets have to be used and hence spatial resolution of the features is degraded. Although the supervised classifier is flexible enough for noise training and therefore relieves part of the problem, it does not solve the problem for the feature extractor. The noise existence actually complicates the training process and usually makes the fuzzy rule base for the classifier larger than necessary, and as a result, slows down the flaw detection process considerably. Therefore, it is necessary to have a noise-free image if we want to obtain the best spatial resolution and the best detection reliability. If the test images happen to have strong noise presence, preprocessing such as noise filtering has to be applied before the flaw detection process in order to have good results.

There are two aspects in the future development of the automated flaw detection system. One is the real-time issue. The system developed is a research prototype and studies are focused on algorithm development and optimization. For a system being used in an industrial field, it must be fast and reliable. A real-time system will be able to inspect a part on the assembly line and inspect multiple image frames quickly. The development of a real-time system will involve hardware and software integrations in the future. Another issue is the sequential processing and recording capabilities. They will enable the inspection of multiple images to be completed in sequence. A user only needs

to specify the sequence for an automated inspection. The software has been designed to give room for future development such as a macro language and a recording process.

**BIBLIOGRAPHY**

- Abu-Mostafa, Y. S. and Psaltis, D., "Recognitive aspects of moment invariants," IEEE Trans. Pattern Anal. Machine Intell., Vol. PAMI-6, No. 6, pp. 698-706, 1984.
- Abu-Mostafa, Y. S. and Psaltis, D., "Image normalization by complex moments," IEEE Trans. Pattern Anal. Machine Intell., Vol. PAMI-7, No. 1, pp. 46-55, 1985.
- Alt, F. L., "Digital pattern recognition by moments," in *Optical Character Recognition*, (G. L. Fischer *et al.*, eds.), Spartan, Washington DC, 1962, pp. 153-179.
- Basart, J., Zhang, Z., and Lyseggen, J., "Automatic detection of flaws in X-ray images," Review of Progress in QNDE, Vol. 13, pp. 747-754, 1994
- Basart, J. P. and Xu, J., "Automatic detection of flaws in welds," Technical Report, Contract #A71445, Center for Non-Destructive Evaluation and Dept. of Electrical Eng. and Computer Eng., Iowa State University, Oct., 1991.
- Batchelor, B. G., *Practical Approach to Pattern Classification*, Plenum Press, , New York, 1974.
- Besl, P and Jain, R. C., "Segmentation through variable-order surface fitting," IEEE Trans. Pattern Anal. Machine Intell., Vol. PAMI-10, No. 2, pp. 167-192, 1988.
- Bezdek, J. C., *Pattern Recognition with Fuzzy Objective Function Algorithms*, Plenum Press, New York, 1982.
- Bhanu, B. and Faugeras, O. D., "Shape matching of two-dimensional objects," IEEE Trans. Pattern Anal. Machine Intell., Vol. PAMI-6, No. 2, pp. 137-156, 1984.
-

- Bigun, J. and du Buf, J. M. H., "N-folded symmetries by complex moments in Gabor space and their application to unsupervised texture segmentation," *IEEE Trans. Pattern Anal. Machine Intell.*, Vol. PAMI-16, No. 1, pp. 88-87, 1994.
- Beale, R. and Jackson, T., *Neural Computing: An Introduction*, Adam Hilger, Bristol, 1990,
- Boerner, H and Strecker, H, "Automated X-ray inspection of Aluminum castings," *IEEE Trans. Pattern Anal. Machine. Intell.*, Vol. 10, No. 1, pp. 79-91, 1988.
- Born, M. and Wolf, E, *Principles of Optics*, Pergamon Press, New York, 1980.
- Bovik, A., Clark, M. and W. Geisler, "Multichannel texture analysis using localized spatial filters," *IEEE Trans. Pattern Anal. Machine Intell.*, Vol. 12, No. 1, pp. 55-73, 1990.
- Boyce, J. F. and Hossack, W. J., "Moment invariants for pattern recognition," *Pattern Recognition Lett.*, Vol. 1, No. 5-6, pp. 451-456, 1983.
- Brown, R. A., "Image processing workstation software development and feature size measurement methods for NDE X-ray images," M.S. Thesis, Iowa State University, Ames, 1992.
- Burt, P. J. and Adelson, E. H., "The Laplacian pyramid as a compact image code," *IEEE Trans. on Communications*, Vol. 31, No. 4, pp. 532-540, 1983.
- Campbell, F. W. and Robson, J. G., "Application of Fourier analysis to the visibility of gratings," *J. Physiol. (London)*, Vol. 197, pp. 551-566, 1968.
- Closier, M. J., "Automatic product analysis using X-rays," *NDT International*, pp. 59-65, April, 1981.
- Coleman, G. B., Andrews, H. C., "Image segmentation by clustering," *Proceedings of IEEE*, Vol. 67, No. 5, pp. 773-785, 1979.
- Crandall, R. E., *Projects in Scientific Computation*, Springer-Verlag, New York, 1994
- Cross, G. R. and Jain, A. K., "Markov random field texture models," *IEEE Trans. Pattern Anal. Machine Intell.*, Vol. 5, No. 1, pp. 25-39, 1983.

- Daubechies, I., "Orthogonal bases of compactly supported wavelets," *Communications on Pure and Applied Mathematics*, Vol. 41, pp. 909-996, 1988.
- Daubechies, I., "The wavelet transform, time-frequency localization and signal analysis," *IEEE Trans. Information Theory*, Vol. 36, No. 5, 1990.
- Daubechies, I., *Ten Lectures on Wavelets*, SIAM, Philadelphia, 1992.
- Davis, L. S., "Shape matching using relaxation techniques," *IEEE Trans. Pattern Anal. Machine Intell.*, Vol. PAMI-1, No. 1, pp. 60-72, 1979.
- Decker, H., "A difference-technique for automatic inspection of casting parts," *Pattern Recognition Lett.*, Vol. 2, pp. 125-129, 1983.
- Dudani, S. A., Breeding, K. J. and McGhee, R. B., "Aircraft identification by moment invariants," *IEEE Trans. Comput.*, Vol. C-26, pp. 39-46, 1977.
- Dunn, D. and Higgins, W. E., "Texture segmentation using 2-D Gabor elementary functions," *IEEE Trans. Pattern Anal. Machine Intell.*, Vol. PAMI-16, No. 2, pp. 130-149, 1994.
- Eisberg, R and Resnick, R., *Quantum Physics of Atoms, Molecules, Solids, Nuclei, and Particles*, John Wiley & Sons, New York, 1974.
- Fu, K. S. and Mui, J. K., "A survey on image segmentation," *Pattern Recognition*, Vol. 13, No. 1, pp. 3-16, 1981.
- Gabor, D., "Theory of communication," *J. IEE*, Vol. 93, pp. 429-457, 1946.
- Gatot, C., "Feature extraction in medical radiographic images," M.S. thesis, Iowa State University, Ames, 1988.
- Haar, A. "Zur theorie der orthogonalen funktionensysteme," *Math. Annal.*, Vol. 69, pp. 331-371, 1910.
- Haralick, R. M., Shanmugan, K., and Dinstein, I., "Texture features for image classification," *IEEE Trans. System, Man, and Cybernetics*, Vol. SMC-3, No. 11, pp. 610-621, 1973.

- Haralick, R. M., "Statistical and structural approach to texture," Proceedings of the IEEE, Vol. 67, No. 5, pp. 786-804, 1979.
- Haralick, R. M., "Digital step edges from zero crossing of second directional derivatives," IEEE Trans. Pattern Anal. Machine Intell., Vol. PAMI-6, No. 1, pp. 58-68, 1984.
- Hornik, K., Stinchcombe, M. and White H., "Multilayer feedforward networks are universal approximators," Neural Networks, Vol. 2, pp. 359-366, 1989.
- Hu, M. K., "Visual pattern recognition by moment invariants," IRE Trans. Inform. Theory, Vol. IT-8, pp. 179-187, 1962.
- Jain, A. K. and Farrokhnia, F., "Unsupervised texture segmentation using Gabor filters," Pattern Recognition, Vol. 23, pp. 1167-1186, 1991.
- Julesz, B., "Visual pattern discrimination," IRE Trans. Information Theory, Vol. IT-8, No. 1, pp. 84-92, 1962.
- Kosko, B., "Fuzziness vs. probability," International Journal of General Systems, Vol. 17, No. 2, pp. 211-240, 1990.
- Kosko, B., *Neural Networks and Fuzzy Systems: A dynamical Systems Approach to Machine Intelligence*, Prentice-Hall, Englewood Cliffs, New Jersey, 1992.
- Kulikowski, J. J., Marcelja, S. and Bishop, P., "Theory of spatial frequency relations in the receptive fields of simple cells in the visual cortex," Bio. Cybern., Vol. 43, pp. 187-198, 1982.
- Lee, C. C., "Fuzzy logic in control systems: Fuzzy logic controller, Part I," IEEE Trans. Syst. Man Cybern., Vol. 20, No. 2, pp. 404-418, 1990.
- Lyseggen, J., "An MS Windows prototype system for automatic general purpose image-based flaw detection," M.S. Thesis, Iowa State University, Ames, 1994.
- Mallat, S., "A theory for multiresolution signal decomposition: the wavelet representation," IEEE Trans. Pattern Anal. machine Intell., Vol PAMI-11, NO. 7, pp. 674-693, 1989.

- McClelland, J. L. and Rumelhart, D. E., *Parallel Distributed Processing, Volume 1*, MIT Bradford Press, Cambridge, 1986.
- Nevatia, R and Babu, K. R., "Linear feature extraction and description," *Computer Graphics and Image Processing*, Vol. 13, pp. 257-269, 1980.
- Pentland, A., "Fractal-based description of natural scenes," *IEEE Trans. Pattern Anal. Machine Intell.*, Vol. 6, No. 6, pp. 661-674, 1984.
- Pollen, D. A. and Ronner, S. F., "Spatial computation performed by simple and complex cells in the visual cortex of the cat," *Vision Research*, Vol. 22, pp. 101-118, 1982.
- Pratt, W. K., Faugeras, O. D., and Gagalowicz, A., "Visual discrimination of stochastic texture fields," *IEEE Trans. Systems, Man, and Cybernetics*, Vol. SMC-8, No. 11, pp. 796-804, 1978.
- Pratt, W. K., *Digital Image Processing*, John Wiley & Sons, New York, 1991.
- Press, W. H, Teukolsky, S. A., Vetterling, W. T, and Flannery, B. P, *Numerical Recipes in C: The Art of Scientific Computing (2nd Edition)*, Cambridge University Press, New York, 1992.
- Reddi, S. S., "Radial and angular moment invariants for image identification," *IEEE Trans. Pattern Anal. machine Intell.*, Vol PAMI-3, pp. 240-242, 1981.
- Rioul, O. and Vetterli, M., "Wavelet and signal processing," *IEEE Signal Processing Magazine*, Vol. 8, No. 4, pp. 14-38, 1991.
- Siwek, E. M., "Application of the X-ray measurement model to image processing of X-ray radiographs," M.S. Thesis, Iowa State University, Ames, 1994.
- Sklansky, J., "On the Hough technique for curve detection," *IEEE Trans. Computers*, Vol. C-27, No. 10, pp. 923-926, 1978.
- Smith, F. W. and Wright, M. H., "Automatic ship photo interpretation by the method of moments," *IEEE Trans. Comput.*, Vol. C-20, pp. 1089-1094, 1971.
- Strang, G., "Wavelets," *American Scientist*, Vol. 82, No. 3, pp. 250-255, 1994.

- Teague, M. R., "Image analysis via the general theory of moments," *J. Opt. Soc. Am.*, Vol. 70, No. 8, pp. 920-930, 1980.
- Teh, C. and Chin, R. T., "On image analysis by the methods of moments," *IEEE Trans. Pattern Anal. Machine Intell.*, Vol. 10., No. 4, pp. 496-513, 1988.
- Telfer, B and Szu, H. H., "New wavelet transform normalization to frequency bias", *Optical Engineering*, Vol. 31, No. 9, pp. 1830-1834, 1992.
- Ulmer, K. W. and Basart, J. P., "General automated flaw detection scheme for NDE X-ray images," in *Review of Progress in Quantitative Nondestructive Evaluation* (D. Thompson and D. Chimenti, eds.), Vol. 12, pp. 327-334, Plenum Press, New York, 1993
- Wang, L and Mendel, J. M., "Generating fuzzy rules by learning from examples," *IEEE Trans. Syst., Man, and Cybern.*, Vol. 22, No. 6, pp. 1414-1427, 1992.
- Wang, L., and Mendel J. M., "Fuzzy basis functions, universal approximation, and orthogonal least-square learning," *IEEE Trans. Neural Networks*, Vol. 3, No. 5, pp. 807-814, 1992.
- Xu, J., Wallingford, R. M., Jensen, T., and Gray, J. N., "Recent developments in the X-ray radiography simulation code: XRSIM," *Review of Progress in Quantitative Nondestructive Evaluation* (D. Thompson and D. Chimenti, eds.), Vol. 13, pp. 557-562, Plenum Press, New York, 1994.
- You, Z., and A. K. Jain, "Performance evaluation of shape matching via chord length distribution," *Computer vision, Graphics, and Image Processing*, Vol. 28, pp. 185-198, 1982.
- Zadeh, L. A., "Fuzzy sets," *Information and Control*, Vol. 8, pp. 338-353, 1965.
- Zhang, Z., and Basart, J., "Feature extraction and classification in automated inspection of NDE images", to appear in *Review of Progress in Quantitative Nondestructive Evaluation* (D. Thompson and D. Chimenti, eds.), Vol. 14, 1995.
- Zhang, Z., and Basart, J., "General purpose automated flaw detection of X-ray images," *Proceedings of the third Annual Midwest Electro-technology Conference, IEEE*, pp. 18-21, April, 1994.