Fuzzy Machine Vision Based Inspection

Pejman Mehran

A Thesis
in
The Department
of
Mechanical and Industrial Engineering

Presented in Partial Fulfillment of the Requirements
for the Degree of Doctor of Philosophy at
Concordia University
Montréal, Québec, Canada

June 2010

Library and Archives
Canada

Published Heritage
Branch

395 Wellington Street
Ottawa ON K1A 0N4
Canada

Bibliothèque et
Archives Canada

Direction du
Patrimoine de l'édition

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file Votre référence
ISBN: 978-0-494-67371-3
Our file Notre référence
ISBN: 978-0-494-67371-3

ABSTRACT

Fuzzy Machine Vision Based Inspection

Pejman Mehran, Ph.D.
Concordia University, 2010

Machine vision system has been fostered to solve many realistic problems in various fields. Its role in achieving superior quality and productivity is of paramount importance. But, for such system to be attractive, it needs to be fast, accurate and cost-effective. This dissertation is based on a number of practical machine vision based inspection projects obtained from the automotive industry. It presents a collection of developed efficient fuzzy machine vision approaches endorsed with experimental results. It also covers the conceptual design, development and testing of various fuzzy machine vision based inspection approaches for different industrial applications. To assist in developing and evaluating the performance of the proposed approaches, several parts are tested under varying lighting conditions.

This research deals with two important aspects of machine vision based inspection. In the first part, it concentrates on the topics of component detection and component orientation identification. The components used in this part are metal clips mounted on a dash panel frame that is installed in the door of trucks. Therefore, we propose a fuzzy machine vision based clip detection model and a fuzzy machine vision based clip orientation identification model to inspect the proper placement of clips on dash panels. Both models are efficient and fast in terms of accuracy and processing time. In the second part of the research, we are dealing with machined part defects such as broken edge, porosity and tool marks. These defects occur on the surface of die cast aluminum automotive pump housings. As a result, an automated fuzzy machine vision based broken edge detection method, an efficient fuzzy machine vision based porosity detection technique and a neuro-fuzzy part classification model based on tool marks are developed. Computational results show that the proposed approaches are effective in yielding satisfactory results to the tested image databases.

There are four main contributions to this work. The first contribution is the development of the concept of composite matrices in conjunction with XOR feature extractor using fuzzy subtractive clustering for clip detection. The second contribution is about a proposed model based on grouping and counting pixels in pre-selective areas which tracks pixel colors in separated RGB channels to determine whether the orientation of the clip is acceptable or not. The construction of three novel edge based features embedded in fuzzy C-means clustering for broken edge detection marks the third contribution. At last, the fourth contribution presents the core of porosity candidates concept and its correlation with twelve developed matrices. This, in turn, results in the development of five different features used in our fuzzy machine vision based porosity detection approach.

ACKNOWLEDGEMENTS

At the outset, I praise God for giving me courage to successfully accomplish my Ph.D. study.

I am extremely grateful to people who have helped me and inspired me throughout my education at Concordia University. First and foremost, I would like to thank my advisor, Prof. Kudret Demirli. I would like to thank him for supporting my Ph.D. study in the past five years. My research progress benefited greatly from him.

I am also thankful to Prof. Brian Surgenor from Queenś University and Prof. Grey Bone from McMaster University for providing us with two datasets of industrial clip and water pump images. Their generous contribution enabled us to supply our detection methods with these applications.

My gratitude also goes to my examination committee, Professors Mingyuan Chen, Brigitte Jaumard and Ali Akgunduz, for their time and valuable comments.

I am also grateful to all faculties, technical and administrative staff members of Concordia University, especially Mrs. Leslie Hosein, who has been kind enough to give me valuable information and support.

I acknowledge Concordia University for the funding that I received from NSERC Research and Auto 21 Grants. The five years at Concordia has become a comprehensive training in research, in writing and in thinking that may benefit me for a whole life.

I am indebted to my all former supervisors and professors especially Dr. Mahmood Hooshmand at Sharif University of Technology, and Professor Vinod Kumar at Carleton University for all the inspiration and guidance which I received from them.

I am also thankful to my previous and current friends in Intelligent Fuzzy Systems Lab, in alphabetical order, Ahmet Kolus, Alebachew Dessalegn Yimer, Hani Osman, Hasan Mehrjerdi, Mina Khoshnejad, Rami Asad and Suji Vijaya kumar.

Final thanks go to my mother, Lila, my father, Hesam and my sister, Mojgan who have always had faith in me and this dissertation would not have been completed without their support and encouragement.

Pejman Mehran
June 2010, Montréal, Canada

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ACRONYMS

2D	-	Two-Dimensional
3D	-	Three-Dimensional
ADC	-	Analog to Digital Conversion
ALPR	-	Automatic License Plate Recognition
ANFIS	-	Adaptive Neuro Fuzzy Inference Systems
ANN	-	Artificial Neural Network
CBIR	-	Content Based Image Retrieval
CV	-	Computer Vision
DSP	-	Digital Signal Processing
ECG	-	Electro-Cardiography
FCM	-	Fuzzy C-Means
FL	-	Fuzzy Logic
FOV	-	Field of View
FS	-	Fuzzy Set
FSMF	-	Fuzzy Set Membership Function
HSL	-	Hue, Saturation and Luminance
IC	-	Integrated Circuit
IP	-	Image Processing
MCU	-	Micro Controller Unit
MMA	-	Multi-modal Analysis
MVBI	-	Machine Vision Based Inspection
PLC	-	Programmable Logic Controller
RAM	-	Read Only Memory
RGB	-	Red, Green and Blue

RMS	-	Reconfigurable Manufacturing System
ROI	-	Region of Interest
SCA	-	Self Clustering Algorithm
SOM	-	Self Organizing Map

Chapter 1

Introduction

1.1. Background and Motivation

Machine vision based inspection, a major part in automation, plays an essential role in many modern manufacturing industries. The success of a company, as a leader or as a follower, depends on its flexibility in launching high quality and competitive products to the changing market. As a result, many manufacturing operations necessitate the inspection of different parts to ensure high quality of the products. Conventionally, these inspections are performed by sampling a batch of products during manufacturing and using statistics and probability to determine acceptance. In fact, the inspections are accomplished by an operator using a flashlight, a comparator with templates or a naked eye. The worker identifies defects in products based on his/her visual judgment. As such, potential inspection defects can be overlooked during the process due to human fatigue. To avoid this problem, it is crucial to provide good environment and work conditions. Fine illumination, comfortable chairs and ergonomic workstations affect the workers' performance in the factory. In addition, it is usually necessary to frequently change operators to avoid fatigue. Some manufacturing processes require more robust and accurate inspection systems such as automotive industry. Corporations like Toyota, Honda and Nissan are only a few cases of such industry requiring a robust inspection system. A good alternative could be the automation of the process based on machine vision. Automated inspection avoids problems associated with operator's fatigue and increases productivity and efficiency while reducing the long term cost. Furthermore, automated inspection allows for 100% inspection as opposed to the sampling approach typically adopted in conventional inspection methods. However, the range and scope of automation are very wide and widespread. Therefore, in this work, we pay special attention to fuzzy machine vision based inspection as one important tool for automation.

The automotive part industry signifies a fast-paced economy sector that experiences high growth and flexibility. This research has been inspired and sponsored by Auto 21. The mission of the Auto 21, an organized network of the Canadian automotive industry, is to assist Canadian auto maker companies in solving their research and development problems. These problems may be related to either manufacturing processes or to the development of new products and technologies. The upcoming dissertation is an endeavor which may result in the use of new technologies in industry and eventually growth of economy. There are many possible manufacturing applications for the proposed machine vision based approaches in this dissertation. In the next section, we define machine vision based inspection.

1.2. Machine Vision Based Inspection Definition

The automatic acquisition and analysis of images to obtain desired information for inspecting a specific activity is called machine vision based inspection. Machine vision based inspection has been receiving more emphasis in industrial applications over the recent decades. It has been taking a key role for making fast and appropriate decisions in production and service sectors. Although, machine vision based inspection has been advancing rapidly across a broad front, the number of systems actually installed in industry is far smaller than the number that could be technically justified [Batchelor and Waltz 1990]. To some degree it is analogous to the transformation of visual sensation into visual perception in biological vision. So it is rational to expect that the biological vision knowledge has a significant impact on machine vision and computer vision literature and methodology. In short, the goal of machine vision is primarily to enable engineering systems to understand and interpret the environment by using visual sensation whereas the computer vision objective is mainly focused on image processing problems.

Inspection means the act of examining and evaluating the production or handling operation of an applicant for certification to determine compliance with the act and the regulations on this part. Machine vision based inspection is the ability of a computer to see and evaluate the production or to handle the operation of an applicant for certification in order to determine compliance with the examiner vision expectations through vision accessories. In other words, machine vision based inspection means interpretation of an image through the use of optical non-contact sensing mechanisms for the purpose of obtaining information to inspect expected items. Machine vision systems often utilize one or more video cameras, analog to digital conversion (*ADC*), often a PC or embedded processor such as a digital signal processing (*DSP*), and finally a program to process images and make relevant decision. The output signal is sent to a robot, a controller or an actuators to reject defective parts.

1.3. Human Vision Performance

The human eye adapts well to different scenes by operating within the frame of his/her training. In fact, the person has been trained during lifetime to distinguish different visual scenes and adapt to new ones by continuously adjusting the visual perception. The human brain builds up representation of objects in his/her visual memory, which guides an efficient selection mechanism. Professor Semir Zeki, of the Institute of Cognitive Neuroscience at University College London believes that the most important factor of human vision system's ability is to maintain color constancy despite diverse light shifts. For instance, when we look at our skin in luminescent room light, and then in the light coming from the window, we don't see a drastic shift in color even though physical measurements (or the output of film or a video camera) would detect a strong reddish tint from the room light and a strong bluish tint from the skylight Solomon [2002]. This is distinctively significant to inspection tasks, where large parts should be investigated. In addition, human has the ability to move and rotate the part, thus altering the illumination direction. The combined multi-direction information enhances the robustness of the image recognition [Abramovich 2004]. Moreover, human vision system merges multiple representations of the inspected object, where both the object of interest and its background change their appearances in different angles and orientations. Zitova and Flusser [2003] makes a distinction for four major machine vision applications in terms of image registration as follows. Our point is that the human vision system has the ability to handle these tasks at the same time and usually better. To provide a fair comparison between human vision performance and current machine vision performance, each visual task is described with an example.

1. Different view points or multi-view analysis: images of the same scene are captured from different viewpoints. In this case, human vision system can analyze different point of views from a particular object. For example, we are able to detect our

friend's face from different views (Full face, right profile and left profile). However, it is not an easy task for current machine visions.

2. Different times or multi-temporal analysis: images of the same scene are captured at different times and possibly under different conditions. The aim is to find and evaluate changes in the scene which appeared between the consecutive image acquisitions. Due to the human visual memory the human vision system has a great performance on this part. For example, we can correctly identify our friend's face during the day under different illuminations. Machine vision systems are not as good as human in this task.

3. Different sensors or multi-modal analysis: Images of the same scene are captured by different sensors. The aim is to integrate the information obtained from different source streams to gain more detailed representation. For example, human vision is stereoscopic, consisting of two sensors which may allow for 3-D view which results in sensing the depth of information. For instance, the cooperation of eyes in human vision system as two separate sensors is glorious. Such outstanding coordination between human eyes is far much better than current multi-camera vision systems.

4. Scene to model registration: Images of a scene are fitted to a predefined model for the sake of image registration. The model can be a computer representation of the scene. For example, a human can categorize different pieces based on a predefined model to some degree but its vision performance in terms of measuring the distance or counting the objects in an image is often poor, problematic and slow. Apparently, it seems that machine vision systems can handle this task much more efficiently.

However, in all the above cases due to monotony and fatigue, human vision performance reduces over time while the machine vision keeps its initial performance and is probably much less costly in the long term.

1.4. Fuzzy Machine Vision Based Inspection

Machine vision based inspection has been drawing more attention in industrial control applications during recent decades. It has been taking a key role for making fast and appropriate decisions in production and service sectors. Machine vision is arguably a mature technology [Pastorius 2001] but there are still many difficulties limiting its widespread use in industry. The operator either sets a wide range of tolerances, or provides multiple master images and finds a way for the system to know which one to use [Killing and Mechefske 2006]. If the changes in light and background are negligible then the system performance is acceptable otherwise in most cases accuracy will be lost. For instance, most inspection algorithms are based on fixed thresholds or single master images. The foundation of a successful machine vision based inspection depends on image processing which in turn relies on feature selections, decision making rules and object classifications [Ozols and Borisov 2001]. Machine vision usually requires digital input/output devices and computer networks to control other manufacturing equipment such as robotic arms. Proper placement of cameras and light sources is perhaps a crucial step in obtaining high quality images which can greatly simplify the vision algorithms and improve their reliability [Cowan *et al.* 1992]. While hardware equipments are very important for image acquisition and storage, the soft computing algorithms remain as the key players in this field in the foreseeable future. A good example of soft computing tool is fuzzy set (*FS*) and fuzzy logic (*FL*). *FS* and *FL* theories are considered as versatile tools for soft computing or computational intelligence, which deal with the design of flexible information processing system. The representation and processing of images depend on the selected fuzzy techniques and the nature of the problem to be solved. An extended analysis of approaches to a problem of fuzzy image processing in a vague environment is addressed by [Zadeh 1977, Bellman *et al.* 1966]. There are different reasons why researchers use fuzzy in their applications. Perhaps, the most important of them is stated by Professor Zadeh, the founder of fuzzy set and fuzzy logic. He says "fuzzy models due to their summarization and information compression through the use of granulation capabilities are functional tools not only in quantifying the evidence from partially developed patterns but also combining evidence from different patterns to identify underlying causes [Zadeh 2008]."

In many machine vision based inspection applications, expert knowledge has been used to overcome the difficulties of recognition. Fuzzy set and fuzzy logic theories offer powerful tools to represent and process human knowledge in form of fuzzy if-then rules. On the other hand, many difficulties in machine vision based inspection arise because the data, applications and uncertainty of environment such as lighting conditions. It is important to note that the uncertainty is not always due to the randomness but to the ambiguity and vagueness. In short, we summarize the contribution of fuzzy in machine vision based inspection as follows:

1. Fuzzy techniques are powerful tools for image representation.

2. Fuzzy techniques can manage the vagueness and ambiguity of noisy images in preprocessing stage.

3. Fuzzy techniques are capable of processing of the images efficiently.

We differentiate three other kinds of imperfection in the machine vision applications. They are grayness ambiguity, geometrical fuzziness and complex or ill-defined knowledge. These problems are fuzzy in the nature. Grayness ambiguity problem deals with the brightness or darkness of a pixel, the geometrical fuzziness problem copes with the boundary between two image segments, and complex or ill-defined knowledge problem is about the validity of the inference system. All of these and similar problems are examples for situations that a fuzzy approach can be more appropriate way to manage the imperfection.

Therefore, fuzzy logic theory has been successfully applied to many other areas, such as pattern recognition, pattern matching, computer vision, etc [Pedrycz and Gomide 1998]. Dealing with the uncertainty problem in image analysis has gained widespread popularity in this field [Krishnapuram and Keller 1992]. Fuzzy machine vision based inspection is a set of different fuzzy approaches to machine vision based inspection. In fact, fuzzy machine vision based inspection is a collective term for different fuzzy approaches that can represent, process and recognize images, with their segments and features as fuzzy sets for the purpose of inspection. Fuzzy logic has also been applied to image enhancement [Cheng and Glazier 2006], contrast enhancement [Chang and Xu 2002], color segmentation [Cheng and Sun 2000], object segmentation [Karmakar and Rahman 2001], threshold optimization [Cheng and Xu 1998, Cheng and Jiang 2000], edge detection [Bezdek and Attikiouzel 1998, Ho 1995, Russo 1999], etc.

1.5. Steps in Machine Vision Based Inspection

There are five general steps in machine vision based inspection as follows:

1. *Image Acquisition*: The first stage in any image processing system is data collection. After images are taken, they are digitized and consequently vectorized. Generally speaking images are converted into numerical form. The best way for representing an image is by using a matrix. For instance in RGB modeling, the dimension of the matrix is three, each dimension representing the colors red, green and blue. This kind of representation is also applied for HSL modeling which stands for hue,

saturation and luminosity. An appropriate type of light and uniform intensity of lighting are essential for proper image acquisition. Therefore, poor illumination leads to a difficult detection process.

2. *Image Registration*: Image registration is the process of aligning two or more images of the same scene. In this process, basic model fitting is performed. A new coordinate system could also be defined. Those parts of the image which are of no interest would be eliminated and the image would be reduced to a smaller size. Equivalently, this action is similar to converting the representing matrix of the image to the corresponding matrix of newly reduced image. This stage is crucial and important in terms of image storage and image retrieval in the database.

3. *Image Pre-processing*: An inborn variance causes the images to differ from one another. Real-world input data always holds some amount of noise and certainly pre-processing is a need to reduce this effect. The term noise could be defined as anything that hinders a pattern recognition system to fulfill its duty, no matter how it inheres to the nature of the data [Cornelius 1998]. Some transformations such as altering contrasts, enhancing images, or filtering operations to highlight the features of interest and diminish background noise take place in this stage.

4. *Image Processing*: This phase plays the main role in image processing. It could be merged with the previous stage. In this stage, the images are broken into features. If the features are carefully chosen, it is expected that the feature set to extract the relevant information from the input image and properly perform the desired task using the reduced representation instead of the full size input. In fact, a feature approach reduces the quantity of information that needs to be processed by subsequent stages.

5. *Decision Making*: This is the final stage. Classification and clustering are done in

this phase. Classification and clustering are two different tasks. In classification, there is some target concept which distinguishes one set from another. In clustering, the idea is merely to produce natural grouping among data. Finally, a decision is made about the trial image. For instance, a defective part image should be recognized in this stage.

1.6. Research Objectives

This thesis attempts to solve machine vision based inspection problems by the aid of fuzzy theory. The main objective of this research is to develop suitable methods for fuzzy machine vision based inspection.

This dissertation includes the two main machine vision based inspection problems which come from two different dataset images. The first problem consists of two sub-problems while the second one is composed of three sub-problems. In the next two paragraphs, we briefly introduce the former and the latter, respectively.

The two inspection sub-problems, faced by a Canadian automotive part manufacturer, are used as two case studies. The first sub-problem involves a vision system which is used to confirm the placement of metal fastening clips on a structural member that supports a truck dash panel. The second sub-problem is more challenging since it is related to the determination of proper orientation of clip if any exists. Therefore, in this part our research objectives are to design intelligent machine vision based inspection methods for

1- Missing clip detection

2- Clip orientation identification

The second problem, consisting of three sub-problems, concerns the manufacturing defects that might occur during the production of water pumps using the die casting method. Such water pumps are made out of aluminum alloys and are used in car engines. The three above mentioned sub-problems are: broken edge detection in a noisy environment, porosity detection versus tool marks effects and part classification based on different tool mark patterns. Therefore, our research objectives are to address these sub-problems pertaining to the design of intelligent machine vision based inspection methods for machining defects such as

1- Broken edge detection

2- Porosity detection

3- Tool mark classification

1.7. Methodology

This thesis pays special attention to finding suitable features for fuzzy classification. Feature extraction for image processing could be probably the most crucial part in this research. Statistical features and XOR based features are developed for object detection in Chapter 3. Geometric based features are proposed for object orientation identification in Chapter 4. Edge matching based features and derivative based features are constructed for broken edge detection in Chapter 5. Matrix correlation based features are developed for porosity detection in Chapter 6. Statistical features and wavelet based features are proposed for tool mark image classification in Chapter 7.

Fuzzy classifiers such as fuzzy subtractive clustering in the third and fourth chapters, fuzzy c means in the fifth and sixth chapters and adaptive network based fuzzy inference system (ANFIS) in the seventh chapter are adopted for classification throughout this thesis. In short, feature extraction and objective fuzzy clustering are common methodology throughout this dissertation.

1.8. Thesis Organization

This dissertation is organized as follows. Chapter 2 summarizes the background, theory and industrial applications of developmental machine vision based inspection. It addresses the relevant experimental and implemental aspects. Chapter 3 deals with the development of a fuzzy machine vision based inspection to detect the placement of metal fastening clips on a structural member that supports a truck dash panel for an automotive manufacturer. Before we arrive at the proposed model different statistical features are analyzed. In Chapter 4, we propose a two-stage fuzzy model which confirms the placement of above mentioned metal fastening clips after they are inserted by a robot arm at the first stage and it inspects the orientation of clip placements at the second stage. Chapter 5 is devoted to developing a computer aided methodology of detection of the broken edges, which are formed in aluminum alloys during production of water pumps for car engines with the die casting method. It illustrates the process of extracting the amount of comparability of the edges in a noisy context based on a predefined template. Chapter 6 addresses the problem of porosity detection which is a challenging two-class problem on the surface of the mentioned water pumps. The main complexity of this problem arises from the similarity between porosity and tool marks. An unsupervised fuzzy approach is adopted to solve this problem. Chapter 7 presents a fuzzy machine vision based approach for image classification of water pumps into three classes to make acceptable decisions according to their texture. We tackle this problem by constructing new features and adopting adaptive neuro-fuzzy inference systems (*ANFIS*). Finally, Chapter 8 summarizes the thesis and its academic contributions and proposes future research directions.

Chapter 2

Literature Review

2.1. Introduction

Successful fuzzy machine vision based inspection system requires proper integration among different stages such as image acquisition, image registration, image pre-processing, image processing and decision making. In this chapter, a brief survey of relevant literature on conceptual analysis and analytical modeling of fuzzy machine vision based inspection is presented. Meanwhile, we provide concise descriptions of algorithm used in general purpose machine vision systems. Since literature on machine vision based inspection covers many disciplines and topics, we conduct a modest review on more prominent parts of fuzzy machine vision based inspection such as feature selection, fuzzy clustering, fuzzy color processing and fuzzy edge-based models.

2.2. Fuzzy Concept in Machine Vision Based Inspection

Fuzzy sets theory introduced by [Zadeh 1965] is an extension of the classical sets. In other words, it is the generalization of conventional (crisp) set theory. Fuzzy sets can handle the concept of partial degree of belongingness using a membership function. For instance, instead of just saying white or black, the color belonging to a set has a degree of whiteness and a degree of blackness. Since the introduction of fuzzy theory, its applications have started to appear. Applications of fuzzy have been reported in various domains such as remotely sensed images, medical imagery, speech recognition and atmospheric sciences [Bezdek and Pal 1992a, Pal and Majumder 1986]. However, one of the most common applications of fuzzy theory is in machine vision based inspection. The contribution of fuzzy in this field is enormous. As a result, many researchers have found it promising to apply fuzzy techniques to develop more sophisticated automated vision systems. Keller [1997a] outlines the impact which fuzzy set theory has already had in the domain of machine vision based inspection. He also mentions the potential need for new fuzzy approaches to solve the difficult tasks specifically for understanding the image data.

Feigenbaum [1961] have shown that manual inspection is seldom efficient in more than 85% of applications. Some researchers including [Li and Lau 1989, Kouatli and Jones 1990, Nath and Lee 1983] have demonstrated the feasibility of applying the fuzzy algorithms to a wide range of interrelated fields such as machine vision, pattern recognition, pattern matching, image processing and process control in an inspection context. When applied to inspection problems, the fuzzy algorithm is called a fuzzy inspection algorithm.

Uncertainty in the machine vision and image processing exists due to different reasons. Noise, regular or irregular background, changes in lighting and in some cases the boundary of an object are subject to different perceptions. In addition, objects will slightly change from one another, even when they belong to one class. These reasons explicitly show the importance of fuzzy theory in machine vision based inspection since fuzzy theory is a proven tool to deal with uncertainty efficiently. Therefore, a machine vision system should have sufficient provision for representing the uncertainties involved in each stage. For instance, in defining image regions and the related features, the relevant information of the original image for making a decision at the highest level should be considered [Mitra and Pal 2005a]. Fuzzy measures and methodologies such as fuzzy entropy, fuzzy geometry, fuzzy axis transform, fuzzy Hough transform, and other fuzzy processing algorithms, have been developed to deal with uncertainty. For instance, Chen and Hoberock [1996] proposed a new fuzzy network, termed *FUZAMP*, in order to cope with situations where the available training data from a machine vision system includes uncertainty. They reported that it performs well when used to recognize different types of fuzzy objects presented at different locations and orientations in the camera's field of view. Again, *FUZAMP* has been successfully implemented to correlate human evaluations with machine evaluations. In the next sections we give a broader perspective of how some researchers have used fuzzy theory as a tool in machine vision based inspection and image processing.

2.3. A Review on Fuzzy Machine Vision Based Inspection Applications

Fuzzy theory has been applied broadly in machine vision based inspection. Chen [1995a] has developed an integrated fuzzy rule based automatic visual inspection system. In his research, domain knowledge for the visual inspection of integrated circuits *IC*s is considered, using several fuzzy rules. An experimental test with well tuned fuzzy parameters shows better results compared to traditional visual inspection method. He concluded fuzzy machine vision based inspection algorithms have their advantages when compared to heuristic search methods in expert systems.

An effective model based machine vision system using fuzzy logic is designed for use in practical production lines by [Lee and Bien 1997]. In their proposed system, the gray level corner detection problem is formulated as a pattern classification problem to determine whether a pixel belongs to the class of corners or not. The probability density function is estimated by means of fuzzy logic. A gray level corner matching method is developed to minimize the amount of calculations. All available information obtained from the gray level corner detector is used to make the model. From a fuzzy inference procedure, a matched segment list is extracted and the resulted segment list is used to calculate the transformations between the object model and each object in the scene.

Wanga *et al.* [2010] proposed a new lane detection model based on fuzzy rules. A combination of the self-clustering algorithm (*SCA*), fuzzy C-mean (*FCM*) and fuzzy rule is used to detect brightness changes in the video scenes and enhance definite information. Orientation of the vehicle with respect to both lane boundaries is computed and a symmetry measure is also used to detect the lane departure warning system. These orientations are implemented to produce a symmetry measure that correctly indicates tendencies of lane departure in advance. Experimental results indicate that the model provides an accurate fit to lane boundaries and can be used to obtain robust information about their orientation.

Lashkia [2001] developed an efficient method based on fuzzy theory. With the proposed algorithm, images are filtered by applying fuzzy reasoning using local image characteristics. The proposed algorithm was applied to detect internal weld defects from radiographic films, which are taken from steel butt weld parts. Results show success in detecting defects at a similar level to human vision. A comparison between a visual and an automatic evaluation demonstrates the efficiency of this method.

Urena and Berenguel [2001] proposed a machine vision system for seeds germination quality evaluation using fuzzy logic. They present an automatic system for monitoring seed germination by means of a software tool incorporating an artificial vision system and a fuzzy logic based classifier. Image information enters a fuzzy logic based classifier that stimulates the quality grading of seedlings, undertaken by experienced technicians, based on their growth pattern.

Sun and Brosnan [2003] developed a fuzzy pizza quality evaluation system using machine vision. The results confirm that the machine vision based inspection can be used for the analysis of pizza base and tomato sauce spread characteristics. For the pizza base analysis, shape and size related features are constructed to classify the captured images. The overall accuracy of the system was only 13% poorer than the human accuracy. The developed sauce spread analysis was based on the use of fuzzy theory for the classification of acceptable and defective quality samples.

2.4. Feature Selection

Feature selection, also known as feature reduction, attribute selection or variable selection aims to select an optimal set with lower dimensionality of features from the feature pool. Dimensionality reduction is very useful in eliminating of the unnecessary information since only relevant information contributes to establish better boundaries among classes in later classification stage. Moreover, processing less information is time-efficient.

The philosophy and style of machine vision and pattern recognition by features are comprehensively examined in both old and new literatures. In its most general form, the feature selection for different patterns is well presented in [Puig and Garca 2003, 2004, Randen and Husoy 1999b]. Selecting irrelevant features can lead to too many and complex fuzzy rules with less accuracy in recognizing patterns. In feature selection literature, a good feature is a sufficient one [Ozols and Borisov 2001]. A feature is described sufficient if it holds at least one article of class (1) and does not hold any article of class (2) for a training set [Pastorius 2001]. Finding good features is not always an easy task. As a result, we conduct an extensive investigation of suitable feature selection in the later chapters of this dissertation.

Image feature selection is based on the extraction of salient attributes or structures in the images. Significant regions, lines or points are understood as features. They should be distinct, disperse all over the image and efficiently recognizable in all the images. Since a number of features in image processing and machine vision have been proposed by researchers, we find it more convenient to provide a concise look at a few dominant features in the rest of this section.

Statistical based features are a group of features which have been reported successful in analyzing the images on some machine vision based inspection applications. For instance, researchers such as [Chandraratne *et al.* 2007, Smith and Smith 2005] proposed efficient statistical features in this field. In the Chapter 3, we investigate some of these statistical based feature approaches.

Geometric based features are another group of features applying in machine vision based inspection. They should be stable in their place during the whole experiment. Geometric based features usually belong to one of the three following categories.

- *Region features*: The region-based feature category can be the projection of general high contrast closed-boundary regions of an appropriate size [Flusser and Suk 1998, Alhichri and Kamel 2003].

- *Line features*: The line based-feature category is the representation of general line segments [Hsieh and Perlant 1992] and in some cases object contours [Dai and Khorram 1997].

- *Point features*: The point based-feature category consists of methods working with line intersection [Stockman and Benett 1982, Vasileisky and Berger 1998].

We have used the geometric based features in the Chapter 4 of this dissertation.

Multi resolution based features, for instance wavelet domain features, have also been used extensively in machine vision based inspection specifically for texture classification and texture segmentation with encouraging results [Muneeswaran *et al.* 2005, Arivazhagan *et al.* 2006]. Some of the proposed multi resolution texture feature extractor methods are quite successful, but most of the applications of the texture analysis are limited to gray scale images. However, some other researchers like Sengur [2007] reports that wavelet base transform features for machine vision color texture classification are highly effective. He proposes a scheme, composed of a wavelet domain feature extractor in conjunction with entropy and energy features. Different color spaces are considered in his experimental study. The overall success rate is reported about 96%. In the Chapter 7, we use wavelet transform to develop our features.

2.5. Fuzzy Clustering and Its Applications on Machine Vision and Image Processing

In this section we discuss some applications of fuzzy clustering in machine vision and image processing since fuzzy clustering could be considered as an efficient algorithm for inspection. Fuzzy set theory has been extensively used in clustering problems where the task is to provide class labels for input data. In fact, abstraction in fuzzy clustering is the estimation of a membership function of inputs from the training samples to generate clusters. Having obtained the estimate, generalization is performed when this estimate is used to compute the values of the membership for unknown objects not confined in the training clusters. Consideration of linguistic features and fuzzy relations in representing a cluster has been suggested by [Bellman *et al.* 1966]. Fuzzy subtractive clustering [Chiu

1994] and fuzzy C-means [Dunn 1973] which later improved by [Bezdek 1981] are two examples of different fuzzy algorithms for clustering. While the former assumes that each data point is a potential cluster center and it measures the likelihood that each data point would be considered as a new cluster center the latter attempts to partition a finite collection of elements into a collection of c fuzzy clusters with respect to given criterion. Fuzzy clustering has numerous applications in machine vision and image processing. Some of them are explained below.

Hasanzadeh and Ahmadi [2007] applied a density based fuzzy clustering technique for non destructive detection of defects in material. They have utilized human conception for classifying defects by the fusion of fuzzy clustering method and fuzzy logic rules based on the density and the size of the defect. The proposed approach shows that there is less dependency between the variation of density and size of a defect and the variation of noise density and distribution.

Sainarayanan and Yaacob [2007] used a new real time image processing scheme with fuzzy clustering algorithms. The processed image is mapped onto a specially structured stereo acoustic pattern and transferred to the stereo earphones in the system. The main objective of the work is to develop an electronic travel aid to assist the blinds for obstacle identification in their navigation.

Lu and Yahagi [2006] developed a method of face recognition based on fuzzy clustering and parallel neural networks. His research presents a new neuro-fuzzy system for face recognition. In particular, the face patterns are divided into several small-scale parallel neural networks based on fuzzy clustering, and they are combined to obtain the recognition result. The proposed method achieves 98.71% recognition accuracy using 310 frontal face images and 98.06% recognition accuracy using 310 rotated face images corresponding to 31 individuals.

Cinque and Lombardi [2004] applied a clustering fuzzy approach for image segmentation. Their work demonstrates a clustering approach for image segmentation based on a modified fuzzy approach. The goal of the proposed approach is finding a model in order to be able to provide a prototype for each cluster and to avoid complex post processing phases. Qualitative and quantitative metrics are claimed to confirm the validity of their approach.

Yin and Sun [2006] used fuzzy clustering with novel separable criterion. In their work, an improved fuzzy clustering algorithm is developed based on the conventional fuzzy c-means (*FCM*) to obtain better quality clustering results. The update equations for the membership and the cluster center are derived from the altering optimization algorithm. Two fuzzy scattering matrices in the objective function assure the fitness between data points and cluster centers. They also secure the separation between cluster centers, using novel separable criterion. They claim that the numerical simulation illustrates more accurate clustering results than the fuzzy c-means.

Since arrhythmias has been the subject of considerable research effort in recent years, some researchers such as Yuksel and Yildrim [2004] present a fuzzy clustering neural network architecture system for classification of electrocardiography *ECG*. Their research presents a comprehensive study of the classification accuracy of *ECG* and a new fuzzy clustering, using neural network architecture for early diagnosis.

Rueda and Zhang [2006] utilized geometric visualization of clusters obtained from fuzzy clustering algorithms. The scheme is based on a geometric visualization and it works by grouping the objects with similar cluster membership. The proposed method shows clear advantages over the existing methods, demonstrating its capabilities for viewing and navigating intercluster relationships in a spatial manner.

Saracoglu and Allahverdi [2007] have proposed a fuzzy clustering approach for finding similar documents using a novel similarity measure. Their research aims for developing a fast high quality method of searching similar documents based on fuzzy clustering in large document collections. In their system, predefined fuzzy clusters are used to extract feature vectors of related documents. It has been seen in experimental results that the proposed system uses new similarity measure which has better performance compared with conventional similarity measurement systems.

Ozols and Borisov [2001] used fuzzy classification based on pattern projection analysis. They developed a method of measuring the comparative efficiency of features and building decision rules. The method is based on the analysis of the structure of the training sets on the binary shadows of composition on co-ordinate hyper-planes in multi- dimensional space. The experimental results have indicated the efficiency of the proposed criterion application in the problem of fuzzy pattern recognition by features.

Cai and Kwan [1998] implemented fuzzy classification using fuzzy inference networks. They argue that most of the existing fuzzy rule based systems have difficulties in deriving inference rules and membership functions directly from training data. Some approaches use back propagation (*BP*) type learning algorithms to learn the parameters of membership functions from training data. However, *BP* learning algorithms take a long time to converge and require a proper setting of the number of inference rules. In their work, self organizing learning algorithms are proposed for the fuzzy inference networks. In the proposed learning algorithms, the number of inference rules and the membership functions in the inference rules will be automatically determined during the training procedure. The proposed fuzzy inference network possesses both the structure and the learning ability of neural networks, and the classification ability of fuzzy algorithms.

2.6. Fuzzy Color Processing

Since fuzzy color processing is prevalent in the machine vision and image processing literature, we briefly review some researchers' work in this field.

Konstantinidis and Andreadis [2005] used image retrieval based on fuzzy color histogram processing. They propose a collection of techniques for retrieving images on the basis of color features. Color image histogram, as an efficient tool has been broadly used in content-based image retrieval (*CBIR*). Although, the classic method of color histogram creation results in many histograms with large variations among neighboring bins, they propose a new fuzzy linking method of color histogram based on the Lab color space with a histogram containing only ten bins. Histogram linking is the procedure of mapping the 3D histogram onto a one-dimensional histogram. Since comparing and concluding 3D histograms is complicated and computationally costly, histogram linking is necessary in machine vision and image processing.

Fuzzy spatial relationship for machine vision plays a major role in color processing. The notion of imperfect information constitutes a key point in image interpretation, visual understanding and structural recognition. This calls for the framework of fuzzy sets, which exhibits proper features to represent spatial imprecision at different levels for better reasoning. To do so, Caulfield and Yoo [2004] performed logic operations on filters before applying them to scenes. Fuzzy logic and Boolean theory both have been used in their work. They illustrated the strength of fuzzy reasoning by applying fuzzy and Boolean masks to a complex image.

Wang [1997] defined a color imaging process with fuzzy set theory to find the distortion lines of a grid in the experimental mechanics of metal forming. The computed results are used to combine with the model of the finite flow line region's method. Calculated results of the velocity and effective strain fields show that this procedure gives reliable results.

Guo and Hsu [2005] introduced an intelligent image agent based on soft computing techniques for color image processing. The intelligent image agent consists of a parallel fuzzy composition mechanism, a fuzzy system related matrix process and a fuzzy adjustment processor to remove impulse noise from highly corrupted images. The fuzzy mechanism embedded in the filter aims at removing impulse noise without destroying fine details and textures. A learning method based on the genetic algorithm completes the adjustment of the filter parameters from a set of training data.

Stanley and Aggarwal [2003] proposed a fuzzy based histogram analysis technique for skin lesion discrimination in dermatological clinical images. The approach utilizes a fuzzy set for skin lesion color, an alpha cut and uses a support set cardinality for quantifying a fuzzy ratio skin lesion color feature.

Smolka [2003] presented a novel method of noise reduction in color images. This new technique is capable of attenuating impulsive and Gaussian noise, while preserving and even enhancing sharpness of the image edges. In his work, a smoothing operator, based on a random walk model and a fuzzy similarity measure between pixels connected by digital paths is introduced.

Wang [1999] used fuzzy c-means approach to color image processing for evaluation the grid lines in plane extrusion. The fuzzy c-means algorithm is first used to calibrate the colors of specimen, and then three fuzzy sets are constructed to filter the image of specimen, such as in the thinning of stream lines, the connection of broken lines and deletion of blurs.

Plataniotis and Venetsanopoulos [1996] developed fuzzy adaptive filters for multi channel image processing. The proposed methodology constitutes a unifying and powerful framework for multi channel image processing. The new filter uses fuzzy membership functions based on different distance measures among the image vectors to adapt to local data in the image.

Frisch [2006] proposed a framework for the segmentation of color images based on the employment of fuzzy integral to classify the input features. The framework uses a self-organizing feature map to determine the coefficients of the fuzzy measure. The performance of the framework is illustrated by successfully realizing the segmentation of color images. At first, the features of the framework and its parameterization are optimized by segmenting different images. Then, the framework is applied to segmentize different images taken under difficult illumination conditions. The presented framework is claimed to render the segmentation of all test color images successfully.

Bloch [2005] reviewed the main fuzzy approaches for defining image spatial relationships including topological and material relations. Topological relations consider the adjacency and material relations consider the distances and directional relative positions.

Zeng and An [2007] presented a multi-face detection method for color images. The method is based on the assumption that faces are well separated from the background by skin color detection. These faces can be located by the proposed method which is a modified version of the subtractive clustering algorithm. The modified method proposes a new definition of distance for multi-face detection and its parameters can be predetermined adaptively by statistical information of face objects in the image. Down sampling is employed to reduce the computation of clustering and facilitate the process of the method.

2.7. Fuzzy Edge Detection on Machine Vision Based Inspection

Fuzzy edge detection has been applied in various image processing and machine vision applications in recent years. Therefore, here we present some of the applications of fuzzy edge detection methods which are relevant to our work.

Yuksel [2007] proposed a neuro-fuzzy edge detector for digital images corrupted by impulse noise. The structure of the detector composes of four identical neuro-fuzzy sub-detectors and a post-processor. The internal parameters of the detector are determined by training. It is claimed that the detector efficiently extracts edges in digital images corrupted by impulse noise without requiring the filtering of the noise.

Kim and Kweon [2004] developed an automatic edge detection using 3 by 3 ideal binary pixel patterns and fuzzy based edge thresholding. An edge magnitude and direction scheme that uses 3 by 3 window patterns and a lookup table is devised. Final edges are automatically determined using the non-maximum suppression with edge confidence metric and fuzzy-based edge thresholding.

Lim [2006] proposed robust edge detection in noisy images. The method is based on testing whether a r by r window is spatially partitioned into two sub-regions. The sub-regions have differences in local gray level value between adjacent pixel neighborhoods. The method employs an edge height model to extract edges of some adequate height from images corrupted with noises. It is claimed that the performance of the proposed edge detector is fully robust to all tested variations of noise.

Kannappady and Shiri [2006] demonstrated the application of fuzzy edge detection for fast object based image retrieval. Fuzzy edge detection coupled with edge thinning module constructs the primary process of creating edge maps. Images containing the object are retrieved by similarity matching using a multi-resolution feature set containing edge crossings with grid lines. For more efficiency, edge detected multi-resolution representations are also processed for all images in the database.

Kim and Han [1998] developed a representation of edges in blurred images using the concept of fuzzy sets when the images are degraded by various asymmetric and local blurring factors. The proposed representation is expressed by fuzzy membership functions, and it serves as a relative index of blur. The membership function is derived from the distribution of intensity gradients and the symmetric of gradient magnitudes, and the function is directly calculated from the blurred image without identifying the point spread function or restoring image.

Liang and Looney [2003] presented competitive fuzzy edge detection. Their fuzzy classifier detects classes of image pixels corresponding to gray level variation in the various directions. It uses an extended function as a fuzzy set membership function (*FCMF*) for each class, where the class assigned to each pixel is the one with the greatest fuzzy truth of membership. To do so, the classification is done initially. The next step is to thin the edges.

Ozbay and Karlik [2006] constructed an edge detector in noisy images by neuro-fuzzy (*NF*) processing. The performance of the proposed edge detector is evaluated on different test images and compared with popular edge detectors from the literature. Simulation results indicate that the proposed (*NF*) operator outperforms competing edge detectors and offers superior performance in edge detection.

Li and Wang [1997] applied the fuzzy inference rule based edge detection approach to detect white line markings, and [Li and Wang 1998] use the fuzzy inference rule-based edge detection approach to detect the road for robot navigation. Combination of fuzzy aggregators has also been employed to smooth the noise; nevertheless, noise has a significant negative impact on the performance of this method. Li and Wang [1997] employed the fuzzy arithmetic average aggregator to replace the fuzzy minimum aggregator, claiming that the algorithm is robust to noise. Tao and Taur [1993] used the fuzzy inference rules without considering the continuity of the edges. Solaiman and Cavayas [1999] implemented the fuzzy inference rule-based approach for road extraction. A road fuzzy membership value is assigned to each pixel, and a set of fuzzy operators characterizing

2D road structures is defined. The algorithm detects the bright road in an aerial image database.

A fuzzy rule based edge detection approach using fuzzy operators to detect specific patterns of neighboring pixels is described in [Russo and Ramponi 1994, Tao and Taur 1993]. Each neighboring pixel is defined as a linguistic variable, a 'bright' or a 'dark' pixel, based on the corresponding membership functions. Subsequently, a fuzzy aggregator is employed to these linguistic variables using a different set of operators.

Chapter 3

Fuzzy Machine Vision Based Clip Detection

3.1. Introduction

Machine vision based inspection is a system capable of acquiring images using an optical non-contact sensing device competent of processing, analyzing and measuring various characteristics so decisions can be made on part inspection. Machine vision based inspection is the application of computer vision to industry and manufacturing whereas computer vision is mainly focused on machine-based image processing [Albovik 1998]. Machine vision based inspection has been receiving more emphasis in industrial applications during the recent decades [Van de Wouwer *et al.* 1999]. It has been taking a key role for making fast and appropriate decisions in production and service sectors. Machine vision is arguably a mature technology [Pastorius 2001] but there are some difficulties to its widespread use in industry. The operator usually either sets very wide tolerances, or provides multiple master images and finds a way for the system to know which one to use [Killing and Mechefske 2006]. If the changes in light and background are negligible, the system performance is acceptable otherwise in most cases accuracy will deteriorate. For instance, most inspection algorithms are based on fixed thresholds or single master image while in this chapter we propose multiple master images.

The foundation of a successful machine vision based inspection depends on image processing which in turn relies on feature selections, decision making rules and object classifications [Ozols and Borisov 2001, Umbaugh 1998]. Machine vision, usually requires digital input/output devices and computer networks to control other manufacturing equipment such as robotic arms. Proper placement of cameras and light sources is perhaps a crucial step in obtaining high-quality images which can greatly simplify the vision algorithms and improve their reliability [Cowan *et al.* 1992, Mehran *et al.* 2006]. While hardware equipment is very important for image acquisition and storage, the soft computing algorithms will remain as the key players in this field in the foreseeable future. A good example of soft computing tool is fuzzy set (FS) and fuzzy logic (FL). FS and FL theories are two versatile tools for soft computing or computational intelligence, which deal with the design of flexible information processing systems.

A seminal analysis of fuzzy approaches for problem solving in a vague environment is addressed [Bellman *et al.* 1966, Zadeh 1977]. Fuzzy image processing is a collective term used for different fuzzy approaches that can recognize, represent and process images, with their segments and features as fuzzy sets [Palit and Popovic 1999, Grimm and Bunke 2007]. The representation and processing of images depend on the selected fuzzy technique and the nature of the problem in question. This chapter is an attempt to solve a machine vision based inspection problem by the help of fuzzy theory. It pays special attention to finding suitable features for fuzzy classification. Feature identification and extraction for image processing would be probably the most crucial part in this research.

The philosophy and style of pattern recognition by features are comprehensively examined in both old and new literature. In its most general form, the feature selection for different patterns is well presented in [Puig and Garca 2004, Randen and Husoy 1999a]. Selecting irrelevant features can lead to too many and complex rules, and less accuracy in recognizing patterns. In feature selection literature, a good feature is a sufficient one [Ozols and Borisov 2001]. A feature is described sufficient if it holds at least one article of class (1) and does not hold any articles of class (2) for a training set [Pastorius 2001]. Finding good features is not always an easy task [Guo *et al.* 2008]. As a result, we conduct an extensive investigation of suitable feature selection. Reference [Mitra and Sankar 2005] is a useful source to realize the different aspects of fuzzy sets in pattern recognition and machine intelligence.

The remaining part of this chapter is organized as follows. Section 2 briefly describes the nature of the problem considered for analysis. In Section 3, the different image processing and classification approaches being implemented to tackle the problems are discussed. In section 4, the theoretical foundations for the fuzzy system used in the presented study are given. In Section 5, the proposed features, fuzzy model and corresponding experimental results have been discussed. Finally, concluding remarks are made in Section 6.

3.2. Problem Description

A group of industrial images obtained from automotive assembly line is considered for analysis. In this set of images, we attempt to detect the presence of a clip which should be attached to a particular location on the assembly. Neither the location of the clip nor its color on each image is clearly defined. In this sense, the challenging part of the problem is to deal with ill-defined objects. In our database we have 1910 images; 1811 of them are present clip images while the remaining 99 are missing clip images. The original images are captured in color (480 × 640 RGB).

The objective is to develop an algorithm in order to detect whether the clip in an image is present or not. As it is shown in Figure 3.1, the clip is placed in its appropriate position. Color images give more information than gray scale images. Gray scale images can be represented as a scalar function while color images can be represented as a vector valued function. The clip is missing in the particular image shown in Figure 3.2. There are quite large differences in the position of the clips within the images among different samples, largely due to the changes in the position of the camera. One could try to compensate for the differences in the position by intelligently locating the correct sub-image to compare, but this problem is almost as difficult as finding the clip itself. A moving robot arm is also in the background. The lack of consistency could bring more difficulty for automatically finding the zone of interest. Variation of the light and color of clips also should be considered. To see the complexity of this problem, see Figures 3.3 and 3.4. Figure 3.3 presents the variation of present clip images in terms of color and position while Figure 3.4 shows the variation of background color and the movement of the frame

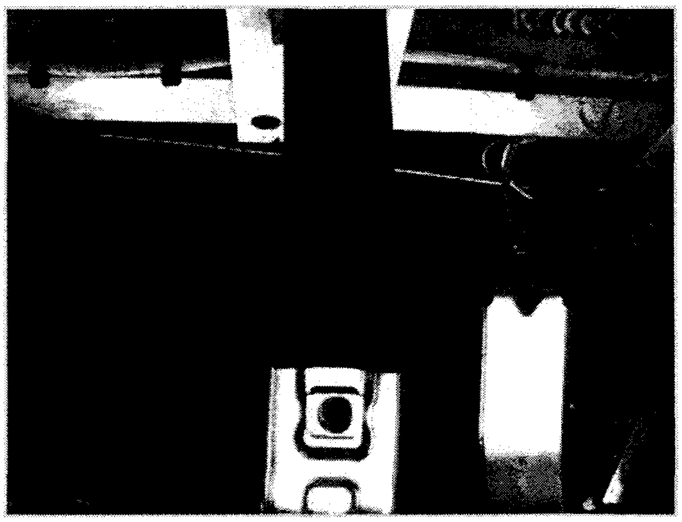


Figure 3.1: The clip is present

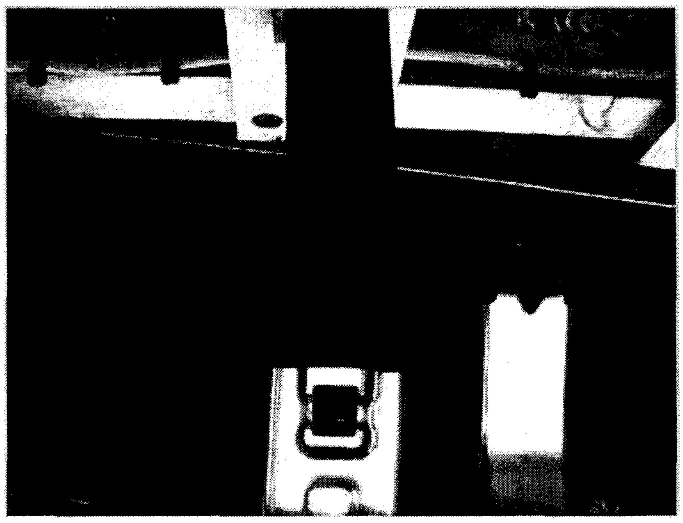


Figure 3.2: The clip is missing

(missing clip images). Finally, we would like to add that orientation in this problem is not an issue since our fuzzy model proposed later in Section 3.5, is orientation invariant to some extent.

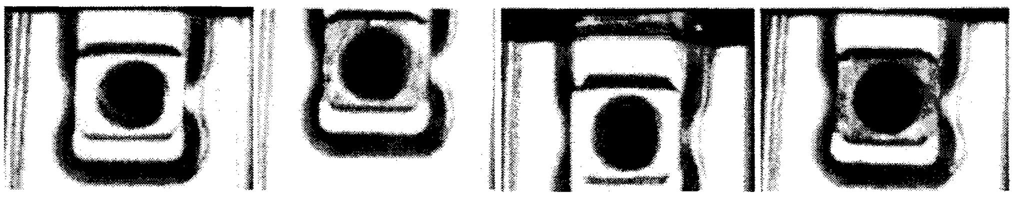


Figure 3.3: Present clip images, variation in color and position

3.3. Various Approaches for Clip Detection

In this section, image processing phases, statistical approaches for feature extraction, and fuzzy modeling are described.

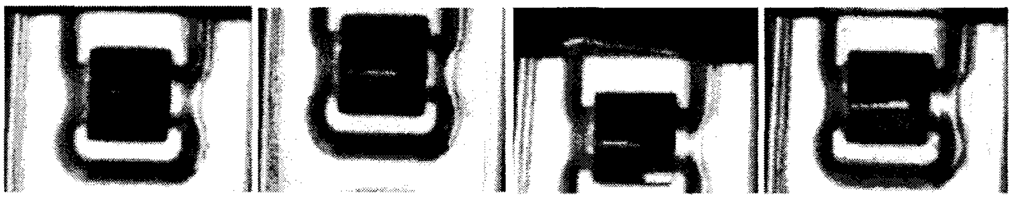


Figure 3.4: Missing clip images, variation due to moving robot arm and camera position

3.3.1. Image Processing Phases

A general pattern recognition system is composed of following phases.

Data Collection

The first stage in any image processing system is data collection. After images are taken they are digitized and consequently vectorized. Generally speaking images are converted into numerical form. The best way for representing an image is in a tensor form. For instance in RGB (red, green, blue) modeling, the dimension of the tensor is three. This kind of representation is also true for HSL modeling where HSL stands for hue, saturation and luminosity [Levkowitzh and Herman 1993].

Registration and Preprocessing

Registration is the process of transforming the different sets of data into one coordinate system. Registration is necessary in order to be able to compare or integrate the data obtained from different measurements. Those parts of the image which are not of interest could be cropped and the image would be reduced to a smaller size. In the registration of data, basic model fitting is performed. The color coordinate system is defined. Typically, this step is usually more than the conversion of the current image to the newly reduced image (zone of interest). This step is crucial and valuable in terms of processing, identification, storage and retrieval of the images in the database. Noise intensifies the natural image variance which causes the images to differ from one another. Real-world input data always holds some amount of noise and certainly preprocessing is needed to reduce this effect. The term noise could be defined as "anything that hinders a pattern recognition system to fulfill its duty, no matter how it inheres to the nature of the data" [Cornelius 1998]. Some attractive properties of the data may also be enhanced or some filtering operations may be applied but in this study, we have avoided the use of any kind of filters due to this fact that our model has the capability to deal with large amount of noise in this application.

Our concern mostly focuses on the variety of light level, clip and moving frame and background changes rather than distinct noise effects. In our database the original image is cropped and a sub-image is taken in order to reduce processing time since the clip is confined to a small area. By adopting Gaussian distribution for all the four corner positions of clip and using the standard deviation of a random sample of size 30 with three sigma limits, one time horizontally and one time vertically, it makes it possible to determine the zone of interest. This operation has been done for north west (x_{nw}, y_{nw}), north east (x_{ne}, y_{ne}), south west (x_{sw}, y_{sw}) and south east (x_{se}, y_{se}) corner points of the clips. The four corner points of the zone of interest (north west (x_w, y_n), north east (x_e, y_n), south west (x_w, y_s) and south east (x_e, y_s) are calculated using the following equations.

$$x_w = min\{\overline{x}_{nw} - 3 \times \sigma_{xnw}, \overline{x}_{sw} - 3 \times \sigma_{xsw}\} \tag{3.1}$$

$$x_e = max\{\overline{x}_{ne} + 3 \times \sigma_{xne}, \overline{x}_{se} + 3 \times \sigma_{xse}\} \tag{3.2}$$

$$y_n = max\{\overline{y}_{nw} + 3 \times \sigma_{y_{nw}}, \overline{y}_{ne} + 3 \times \sigma_{y_{ne}}\} \tag{3.3}$$

$$y_s = min\{\overline{y}_{sw} - 3 \times \sigma_{y_{sw}}, \overline{y}_{se} - 3 \times \sigma_{y_{se}}\} \tag{3.4}$$

where, $(\overline{x}_{nw}, \overline{y}_{nw})$, $(\overline{x}_{ne}, \overline{y}_{ne})$, $(\overline{x}_{sw}, \overline{y}_{sw})$ and $(\overline{x}_{se}, \overline{y}_{se})$ are the average corner clip positions in the sample for north west, north east, south west and south east, respectively. In a similar fashion, $(\sigma_{xnw}, \sigma_{y_{nw}})$, $(\sigma_{xne}, \sigma_{y_{ne}})$, $(\sigma_{xsw}, \sigma_{y_{sw}})$ and $(\sigma_{xse}, \sigma_{y_{se}})$ present the standard deviation of the corresponding corner clip positions in the sample. The special characteristic of number 30 (sample size) arises from the Central Limit Theorem (CLT) which helps us to assume Gaussian distribution for determining the zone of interest in the sample images. Therefore, the image size of the cropped image is reduced from a 480 × 640 matrix to a 51 × 61 matrix. Investigation of the relationship between different zones and yielding can also be done for optimization or post optimality analysis which are not the purpose of this chapter. As a result, the number of pixels of newly reduced image, decreases from 307200 pixels to only 3111 pixels. That is, the zone of interest is about 1 percent of the original image (Figure 3.5). Due to the capability of our model to deal with the changes which result from using a three channel (RGB) fuzzy pixel based approach, we have considered a fixed zone of interest. That is, the place of the zone of interest is always permanent on the original image. However, we choose a larger interest zone to handle all the clip variations discussed earlier in Section 2. In some applications the interest zone is not fixed any more, such as in occasions when we relate the interest zone to the position of another object which can easily be found in the image. Finally, we would also like to mention that one of the advantages of proposed fuzzy model in Section 3.5.2 is its capability to handle to some degree the differences in choosing the zone of interest which

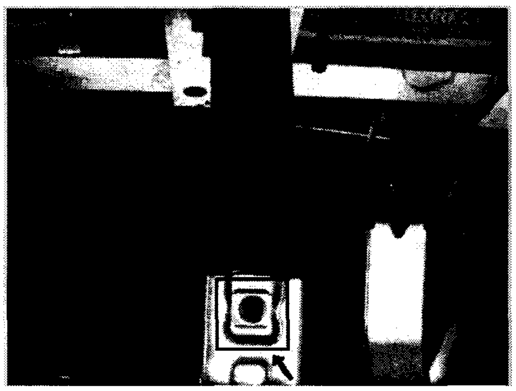


Figure 3.5: Zone of interest

result from composite present matrices, composite missing matrices and XOR feature extractor. However, there are other possible approaches to tackle this problem. Moment of inertia after thresholding, detection of shape using counters are examples, to name a few.

Setting Reference Images and Feature Extraction

Two reference composite images as the representative of both present clip images and missing clip images are introduced. Let us call Ω as the set of all images and Ω_p, Ω_m as the sets of images with present (1811 images) and missing clips (99 images), respectively. Therefore,

$$\Omega = \Omega_p \bigcup \Omega_m, \; \Omega_p \in \Omega \text{ and } \Omega_m \in \Omega \tag{3.5}$$

Now, we take $N_p = 30$ random sample images from Ω_p, and $N_m = 30$ random sample images from Ω_m. As we mentioned earlier in Section 3.3.1 using the three sigma methodology the zone of interest (cropped image) is reduced to a matrix with a size of 51×61. Therefore, the lower and upper limits for the x-axis and the y-axis in the zone of interest for all the images are $(L_x = 1, U_x = 61)$ and $(L_y = 1, U_y = 51)$, respectively. Number of all the pixels in the zone of interest is $N = 3111$. We define the k^{th} present clip sample image as a vector-valued picture function as follows:

$$\mathbf{f}_k(x,y) = [f_{R_k}(x,y), f_{G_k}(x,y), f_{B_k}(x,y)] \tag{3.6}$$

where, f_{R_k}, f_{G_k} and f_{B_k} present the values of the color (red, green, and blue) of the image at the point (x,y). With a similar fashion, we define the k^{th} missing clip sample image as a vector-valued picture function as follows:

$$\mathbf{g}_k(x,y) = [g_{R_k}(x,y), g_{G_k}(x,y), g_{B_k}(x,y)] \tag{3.7}$$

where, g_{R_k}, g_{G_k} and g_{B_k} present the values of the color of the image at the point (x,y). In the next step, we calculate the average pixel values in present clip sample images with respect to their colors and consider them as the representative images of present clip images, $\bar{\mathbf{f}}(x,y)$.

$$\bar{\mathbf{f}}(x,y) = [\bar{f}_R(x,y), \bar{f}_G(x,y), \bar{f}_B(x,y)] \tag{3.8}$$

where,

$$\bar{f}_R(x,y) = \frac{1}{N_p} \sum_{k=1}^{N_p} f_{R_k}(x,y) \tag{3.9}$$

$$\bar{f}_G(x,y) = \frac{1}{N_p} \sum_{k=1}^{N_p} f_{G_k}(x,y) \tag{3.10}$$

$$\bar{f}_B(x,y) = \frac{1}{N_p} \sum_{k=1}^{N_p} f_{B_k}(x,y) \tag{3.11}$$

Similarly, we do the same calculations for missing clip sample images and calculate the missing representative images, $\overline{\mathbf{g}}(x,y)$.

$$\overline{\mathbf{g}}(x,y) = [\overline{g}_{R_k}(x,y), \overline{g}_{G_k}(x,y), \overline{g}_{B_k}(x,y)] \tag{3.12}$$

where,

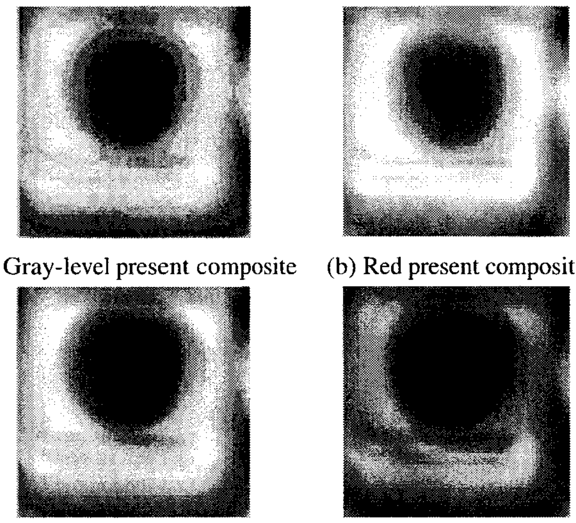
$$\overline{g}_R(x,y) = \tfrac{1}{N_m}\textstyle\sum_{k=1}^{N_m} g_{R_k}(x,y) \tag{3.13}$$

$$\overline{g}_G(x,y) = \tfrac{1}{N_m}\textstyle\sum_{k=1}^{N_m} g_{G_k}(x,y) \tag{3.14}$$

$$\overline{g}_B(x,y) = \tfrac{1}{N_m}\textstyle\sum_{k=1}^{N_m} g_{B_k}(x,y) \tag{3.15}$$

We call these representative images as composite images or equivalently as composite matrices. These composite images are shown in Figures 3.6 and 3.7. One can observe vague images in Figure 3.6 and Figure 3.7 compared to the images in Figure 3.1 and Figure 3.2. This ambiguity is expected since Figure 3.6 and 3.7 are both a linear combination of 30 different images.

In addition, two reference gray-level images with all the pixels to be pure white (W) or pure black (B) are also constructed in order to take care of the variations in brightness of images in the database. All the elements of a white matrix are 255, while all the elements of a black matrix are 0. In this sense, we have three composite images of present clip images for color red, green and blue, three composite images of missing clip images for color red, green and blue, and two white and black reference images. Different features



(a) Gray-level present composite (b) Red present composite

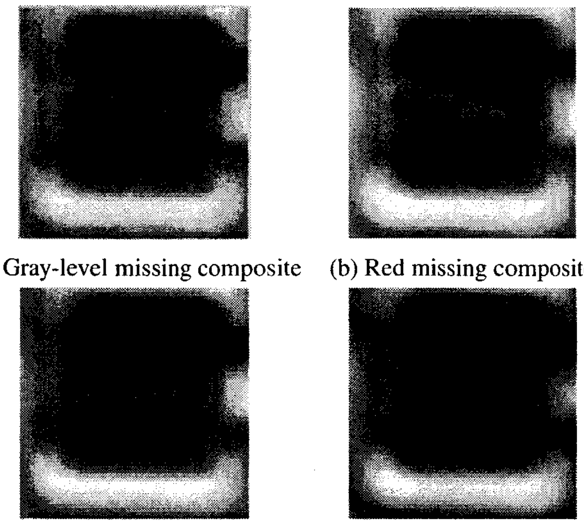
(c) Green present composite (d) Blue present composite

Figure 3.6: Vague composite image of present clip

will be thoroughly investigated in this chapter to find suitable ones for comparing trial images in conjunction with composite images. In reference [Tzanakou 2001, Sklansly 1978] several features have been analyzed.

Image Processing

This stage of processing is sometimes called segmentation. It could be merged with previous steps. This phase plays a main role in image processing as classification and clustering are done in this phase. Classification and clustering are different tasks. In classification, there is some target concept which distinguishes one set from another. In clustering, the idea is merely to produce natural groupings. In this chapter, we use the subtractive clustering method to extract the fuzzy rules.



(a) Gray-level missing composite (b) Red missing composite

(c) Green missing composite (d) Blue missing composite

Figure 3.7: Vague composite image of missing clip

Decision Making

This is the final step. In this phase a decision is assigned to the trial image. In this chapter, we are interested in determining whether the current image belongs to the set of present clip images or missing clip images. Using our sample images and fuzzy rules the parameters and coefficients of the model are set and applied to rest of the images for decision making. Fuzzy logic, Neural networks, Neuro-fuzzy models and statistical approaches can be used as other reasonable alternatives for decision making [Wang and Liu 2006, Binaghi 2007, Peters *et al.* 2002]. However, due to the ability of fuzzy logic in dealing with vague information, poorly defined clips and noisy images, we have adopted fuzzy subtractive clustering for developing a Sugeno model for final decision making.

3.3.2. Statistical Approaches for Feature Selection

We first perform some statistical analysis in order to characterize the variation between present and missing clip images. As we earlier mentioned in Section 3.2, there are 1811 present clip images and 99 missing clip images in our database. We took a random sample of 30 present clip images and a random sample of 30 missing clip images. We constantly use these samples throughout this chapter. We perform mean, variance, Euclidean distance, covariance and histogram analysis (absolute deviation and least square deviation) of the red, green and blue values for both present and missing clip groups. These statistical investigations are carried out in order to extract a suitable feature for differentiation. Basically, in this stage we are looking for a feature to classify the images. Any feature based classification should answer this question: to what extent each property could identify the images? In other words, which feature is the most suitable for classification?

Mean Color Comparison

The first analysis is the investigation of the average of red, green and blue values matrices. We calculate the average RGB values for all the 60 images in both sample groups. The formulas for calculating mean value in the k^{th} present clip sample image ($\mu_{R_k}^{(p)}$, $\mu_{G_k}^{(p)}$, $\mu_{B_k}^{(p)}$) and corresponding present composite which is the average present mean of the sample ($\overline{\mu}_R^{(p)}$, $\overline{\mu}_G^{(p)}$, $\overline{\mu}_B^{(p)}$) with respect to each color are shown in Equation (3.16) to Equation (3.21).

$$\mu_{R_k}^{(p)} = \frac{1}{N}\sum_{x=1}^{N_x}\sum_{y=1}^{N_y} f_{R_k}(x,y) \tag{3.16}$$

$$\mu_{G_k}^{(p)} = \frac{1}{N}\sum_{x=1}^{N_x}\sum_{y=1}^{N_y} f_{G_k}(x,y) \tag{3.17}$$

$$\mu_{B_k}^{(p)} = \frac{1}{N}\sum_{x=1}^{N_x}\sum_{y=1}^{N_y} f_{B_k}(x,y) \tag{3.18}$$

$$\overline{\mu}_R^{(p)} = \frac{1}{N_p}\sum_{k=1}^{N_p} \overline{f}_R(x,y) \tag{3.19}$$

$$\overline{\mu}_G^{(p)} = \frac{1}{N_p}\sum_{k=1}^{N_p} \overline{f}_G(x,y) \tag{3.20}$$

$$\overline{\mu}_B^{(p)} = \frac{1}{N_p}\sum_{k=1}^{N_p} \overline{f}_B(x,y) \tag{3.21}$$

In a similar fashion, the mean value in the k^{th} missing clip sample image ($\mu_{R_k}^{(m)}$, $\mu_{G_k}^{(m)}$, $\mu_{B_k}^{(m)}$) and corresponding missing composite which is the average missing mean of the sample ($\overline{\mu}_R^{(m)}$, $\overline{\mu}_G^{(m)}$, $\overline{\mu}_B^{(m)}$) with respect to each color are shown in Equation (3.22) to Equation (3.27).

$$\mu_{R_k}^{(m)} = \frac{1}{N}\sum_{x=1}^{N_x}\sum_{y=1}^{N_y} g_{R_k}(x,y) \tag{3.22}$$

$$\mu_{G_k}^{(m)} = \frac{1}{N}\sum_{x=1}^{N_x}\sum_{y=1}^{N_y} g_{G_k}(x,y) \tag{3.23}$$

$$\mu_{B_k}^{(m)} = \frac{1}{N}\sum_{x=1}^{N_x}\sum_{y=1}^{N_y} g_{B_k}(x,y) \tag{3.24}$$

$$\overline{\mu}_R^{(m)} = \frac{1}{N_m}\sum_{k=1}^{N_m} \overline{g}_R(x,y) \tag{3.25}$$

$$\overline{\mu}_G^{(m)} = \frac{1}{N_m}\sum_{k=1}^{N_m} \overline{g}_G(x,y) \tag{3.26}$$

$$\overline{\mu}_B^{(m)} = \frac{1}{N_m}\sum_{k=1}^{N_m}\overline{g}_B(x,y) \tag{3.27}$$

As shown in Figure 3.8, the average present mean red value is much greater than the average missing mean red value since the clips are made of copper. If we can also verify that minimum present mean red value is greater than the maximum missing mean red value, then the mean red values of an image can be used as a suitable feature for classifying present and missing clip images. However, the minimum present mean red value which is 150.69, is less than the maximum missing mean red value which is 159.9. That is, there is an overlap between the range of mean red values of present and missing clip images. This creates a problem in choosing mean as a feature for differentiation. Similar results are also true for green and blue values. As a result, mean color values can not be used as a feature for classification in this application.

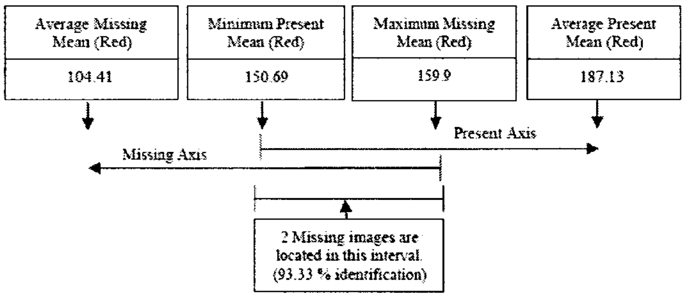

Average Missing Mean (Red)
104.41
Minimum Present Mean (Red)
150.69
Maximum Missing Mean (Red)
159.9
Average Present Mean (Red)
187.13
Present Axis
Missing Axis
2 Missing images are located in this interval. (93.33 % identification)

Figure 3.8: Overlap between mean of present and missing clip images (red values)

Variance Comparison

It is hypothesized that the variance of each value in each image may give sufficient amount of information for classification. The formulas for calculating variance for present and missing clip sample images with respect to each color are shown in Equation (3.28) and Equation (3.33), respectively.

$$var(f_{R_k}) = \frac{1}{N}\sum_{x=1}^{N_x}\sum_{y=1}^{N_y}(f_{R_k}(x,y) - \mu_{R_k}^{(p)})^2 \tag{3.28}$$

$$var(f_{G_k}) = \frac{1}{N}\sum_{x=1}^{N_x}\sum_{y=1}^{N_y}(f_{G_k}(x,y) - \mu_{G_k}^{(p)})^2 \tag{3.29}$$

$$var(f_{B_k}) = \frac{1}{N}\sum_{x=1}^{N_x}\sum_{y=1}^{N_y}(f_{B_k}(x,y) - \mu_{B_k}^{(p)})^2 \tag{3.30}$$

$$var(g_{R_k}) = \frac{1}{N}\sum_{x=1}^{N_x}\sum_{y=1}^{N_y}(g_{R_k}(x,y) - \mu_{R_k}^{(m)})^2 \tag{3.31}$$

$$var(g_{G_k}) = \frac{1}{N}\sum_{x=1}^{N_x}\sum_{y=1}^{N_y}(g_{G_k}(x,y) - \mu_{G_k}^{(m)})^2 \tag{3.32}$$

$$var(g_{B_k}) = \frac{1}{N}\sum_{x=1}^{N_x}\sum_{y=1}^{N_y}(g_{B_k}(x,y) - \mu_{B_k}^{(m)})^2 \tag{3.33}$$

Figure 3.9 shows us that the idea of choosing variance for feature classification is only applicable to some extent. Since the standard deviation is the square root of variance, similar conclusions can be drawn for standard deviation as well. In a similar manner, neither green nor blue variances can overcome this problem and we still have a significant overlap.

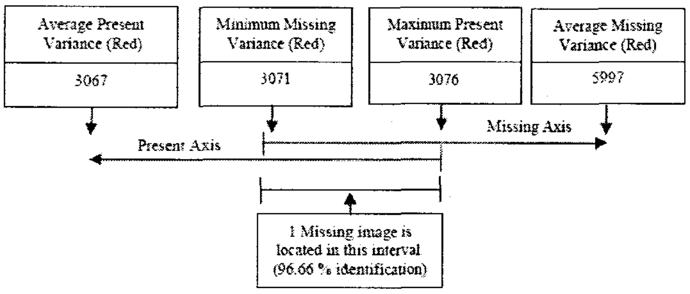

Average Present Variance (Red)
3067
Minimum Missing Variance (Red)
3071
Maximum Present Variance (Red)
3076
Average Missing Variance (Red)
5997
Missing Axis
Present Axis
1 Missing image is located in this interval (96.66 % identification)

Figure 3.9: Overlap between variance of present and missing clip images (red)

Euclidean Distance Comparison

Euclidean distance is investigated in order to find any dominant features between two groups of images. Euclidean distances which are computed for present and missing clip sample images in the dataset are given by Equation (3.34) and Equation (3.35), respectively.

$$E_k^{(p)} = \frac{1}{N}\sum_{x=1}^{N_x}\sum_{y=1}^{N_y}\sqrt{f_{R_k}^2(x,y)+f_{G_k}^2(x,y)+f_{B_k}^2(x,y)} \tag{3.34}$$

$$E_k^{(m)} = \frac{1}{N}\sum_{x=1}^{N_x}\sum_{y=1}^{N_y}\sqrt{g_{R_k}^2(x,y)+g_{G_k}^2(x,y)+g_{B_k}^2(x,y)} \tag{3.35}$$

We observe that the average present Euclidean distance value is greater than the average missing Euclidean distance value. However, in Figure 3.10, it can be seen that the overlap still exists.

Covariance Comparison

Since there could be meaningful information regarding the correlation of color values, we investigate the covariance of red and green; red and blue; green and blue and finally red, green and blue pixel values. The covariance of any two given standard colors for the k^{th}

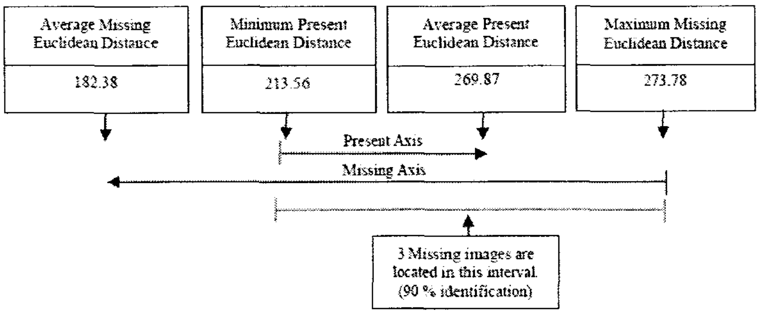

Average Missing Euclidean Distance
182.38
Minimum Present Euclidean Distance
213.56
Average Present Euclidean Distance
269.87
Maximum Missing Euclidean Distance
273.78
Present Axis
Missing Axis
3 Missing images are located in this interval. (90 % identification)

Figure 3.10: Overlap between Euclidean distances of present and missing clip images

present and missing clip image can be determined by Equation (3.36) through Equation (3.41), respectively.

$$cov(f_{R_k}, f_{G_k}) = \frac{1}{N}\sum_{x=1}^{N_x}\sum_{y=1}^{N_y}(f_{R_k}(x,y)f_{G_k}(x,y) - \mu_{R_k}^{(p)}\mu_{G_k}^{(p)}) \tag{3.36}$$

$$cov(f_{R_k}, f_{B_k}) = \frac{1}{N}\sum_{x=1}^{N_x}\sum_{y=1}^{N_y}(f_{R_k}(x,y)f_{B_k}(x,y) - \mu_{R_k}^{(p)}\mu_{B_k}^{(p)}) \tag{3.37}$$

$$cov(f_{G_k}, f_{B_k}) = \frac{1}{N}\sum_{x=1}^{N_x}\sum_{y=1}^{N_y}(f_{G_k}(x,y)f_{B_k}(x,y) - \mu_{G_k}^{(p)}\mu_{B_k}^{(p)}) \tag{3.38}$$

$$cov(g_{R_k}, g_{G_k}) = \frac{1}{N}\sum_{x=1}^{N_x}\sum_{y=1}^{N_y}(g_{R_k}(x,y)g_{G_k}(x,y) - \mu_{R_k}^{(m)}\mu_{G_k}^{(m)}) \tag{3.39}$$

$$cov(g_{R_k}, g_{B_k}) = \frac{1}{N}\sum_{x=1}^{N_x}\sum_{y=1}^{N_y}(g_{R_k}(x,y)g_{B_k}(x,y) - \mu_{R_k}^{(m)}\mu_{B_k}^{(m)}) \tag{3.40}$$

$$cov(g_{G_k}, g_{B_k}) = \frac{1}{N}\sum_{x=1}^{N_x}\sum_{y=1}^{N_y}(g_{G_k}(x,y)g_{B_k}(x,y) - \mu_{G_k}^{(m)}\mu_{B_k}^{(m)}) \tag{3.41}$$

In the same manner, the covariance of the three standard colors in the k^{th} image can be defined for present and missing clip sample images in Equation (3.42) and Equation (3.43), respectively.

$$cov(f_{R_k}, f_{G_k}, f_{B_k}) = \frac{1}{N}\sum_{x=1}^{N_x}\sum_{y=1}^{N_y}(f_{R_k}(x,y)f_{G_k}(x,y)f_{B_k}(x,y) - \mu_{R_k}^{(p)}\mu_{G_k}^{(p)}\mu_{B_k}^{(p)}) \quad (3.42)$$

$$cov(g_{R_k}, g_{G_k}, g_{B_k}) = \frac{1}{N}\sum_{x=1}^{N_x}\sum_{y=1}^{N_y}(g_{R_k}(x,y)g_{G_k}(x,y)g_{B_k}(x,y) - \mu_{R_k}^{(m)}\mu_{G_k}^{(m)}\mu_{B_k}^{(m)}) \quad (3.43)$$

Figure 3.11 shows that identification power of red-green covariance is as low as 87.27%. We arrived at similar results for other correlations as well. These results exhibit that correlations do not contain desirable information by themselves to make 100% differentiation.

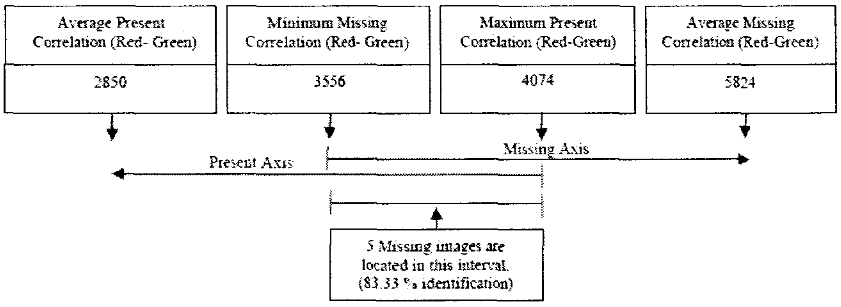

Average Present Correlation (Red- Green)
2850
Minimum Missing Correlation (Red- Green)
3556
Maximum Present Correlation (Red-Green)
4074
Average Missing Correlation (Red-Green)
5824
Missing Axis
Present Axis
5 Missing images are located in this interval. (83.33 % identification)

Figure 3.11: Overlap between red and green correlations

Frequency Comparison

We have divided the range of 0 - 255 into 256 different levels and calculated the frequency of pixels that falls into each corresponding range. Figure 3.12 shows the average frequency for red values in present clip images and missing clip images in our sample of

30 images in each group. By comparing the frequency of trial image with the present and missing average frequency, it appears that we can categorize the test image. This comparison is done for all three colors and the calculations are based on absolute deviation and least square deviation. The rule is simple; the smaller the deviation, the more the similarity. It means that the test image belongs to the group with less deviation which presents more similarity. The frequency plots of green and blue values are presented in Figures 3.13 and 3.14. Figures 3.12, 3.13 and 3.14 show that red is the most important color followed by green and eventually in terms of frequency comparison.

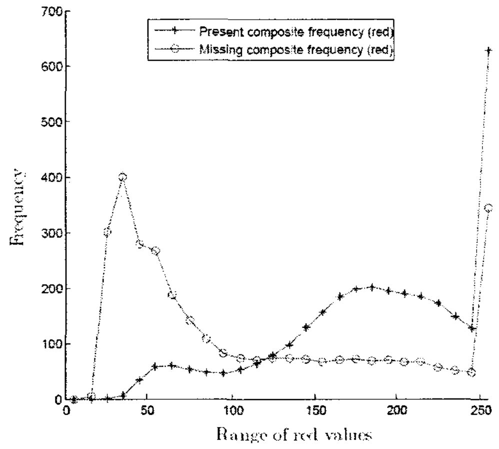


Figure 3.12: Present and missing clip composite frequency (red)

Absolute Frequency Deviation

Absolute frequency deviations (AFD) of the k^{th} present and missing clip sample image with respect to each color from corresponding composite matrices are represented by $(AFD^{(p)}_{R_k}, AFD^{(p)}_{G_k}, AFD^{(p)}_{B_k})$ and $(AFD^{(m)}_{R_k}, AFD^{(m)}_{G_k}, AFD^{(m)}_{B_k})$.
$\mathbf{h}^{(p)}_k(n)$, $\mathbf{h}^{(m)}_k(n)$, $\mathbf{h}^{(\overline{p})}(n)$ and $\mathbf{h}^{(\overline{m})}(n)$ return the number of pixels whose values are n with respect to each color in the k^{th} present clip image, k^{th} missing clip image, present

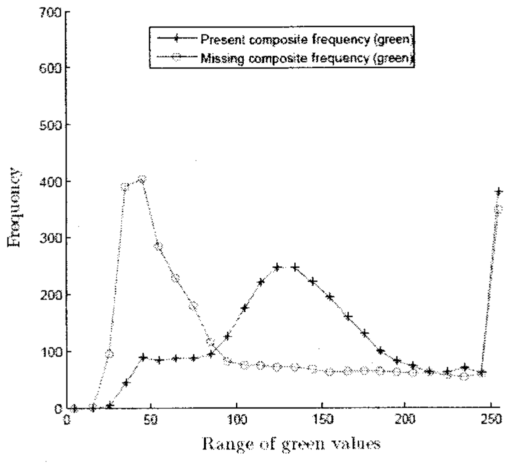


Figure 3.13: Present and missing clip composite frequency (green)

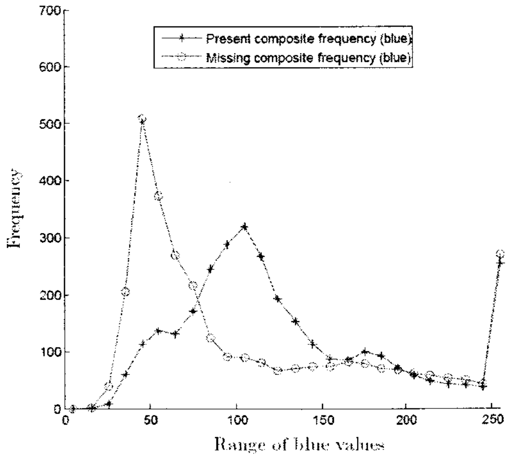


Figure 3.14: Present and missing clip composite frequency (blue)

composite and missing composite, respectively. The vector-valued function of $\mathbf{h}_k^{(p)}(n)$, $\mathbf{h}_k^{(m)}(n)$, $\mathbf{h}^{(\overline{p})}(n)$ and $\mathbf{h}^{(\overline{m})}(n)$ are shown as follows.

$$\mathbf{h}_k^{(p)}(n) = [h_{R_k}^{(p)}(n), h_{G_k}^{(p)}(n), h_{B_k}^{(p)}(n)] \tag{3.44}$$

$$\mathbf{h}_k^{(m)}(n) = [h_{R_k}^{(m)}(n), h_{G_k}^{(m)}(n), h_{B_k}^{(m)}(n)] \tag{3.45}$$

$$\mathbf{h}^{(\overline{p})}(n) = [h_{R_k}^{(\overline{p})}(n), h_{G}^{(\overline{p})}(n), h_{B}^{(\overline{p})}(n)] \tag{3.46}$$

$$\mathbf{h}^{(\overline{m})}(n) = [h_{R}^{(\overline{m})}(n), h_{G}^{(\overline{m})}(n), h_{B}^{(\overline{m})}(n)] \tag{3.47}$$

Therefore, the absolute frequency deviations of each color for each present and missing clip image from corresponding color composite matrices are given by Equation (3.48) and Equation (3.53), respectively.

$$AFD_{R_k}^{(p)} = \sum_{n=0}^{255} \mid h_{R_k}^{(p)}(n) - h_{R_k}^{(\overline{p})}(n) \mid \tag{3.48}$$

$$AFD_{G_k}^{(p)} = \sum_{n=0}^{255} \mid h_{G_k}^{(p)}(n) - h_{G_k}^{(\overline{p})}(n) \mid \tag{3.49}$$

$$AFD_{B_k}^{(p)} = \sum_{n=0}^{255} \mid h_{B_k}^{(p)}(n) - h_{B_k}^{(\overline{p})}(n) \mid \tag{3.50}$$

$$AFD_{R_k}^{(m)} = \sum_{n=0}^{255} \mid h_{R_k}^{(m)}(n) - h_{R_k}^{(\overline{m})}(n) \mid \tag{3.51}$$

$$AFD_{G_k}^{(m)} = \sum_{n=0}^{255} | h_{G_k}^{(m)}(n) - h_{G_k}^{(\overline{m})}(n) | \tag{3.52}$$

$$AFD_{B_k}^{(m)} = \sum_{n=0}^{255} | h_{B_k}^{(m)}(n) - h_{B_k}^{(\overline{m})}(n) | \tag{3.53}$$

As shown in Figure 3.15, the overlap between the two groups of images shows that the idea of classification of images by the help of absolute deviation is fine but not prefect (Only 96.66% identification). Similar results are obtained for green and blue colors. Present clip composite image is set as the reference for comparison with trial image rather than the missing clip composite image.

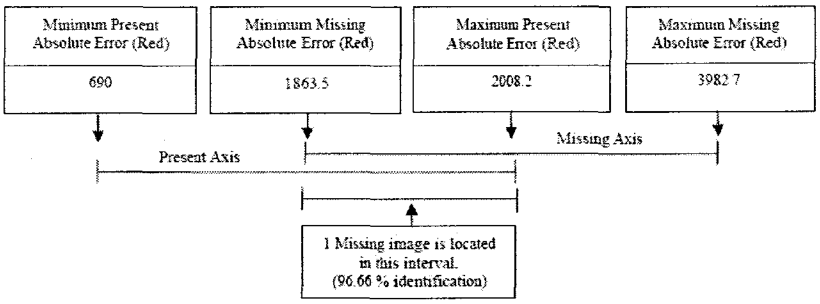

Minimum Present Absolute Error (Red)
690
Minimum Missing Absolute Error (Red)
1863.5
Maximum Present Absolute Error (Red)
2008.2
Maximum Missing Absolute Error (Red)
3982.7
Present Axis
Missing Axis
1 Missing image is located in this interval. (96.66 % identification)

Figure 3.15: Overlap between the absolute deviations of the present and missing clip images

Least Square Frequency Deviation

Least square frequency deviations (LSFD) of the k^{th} present and missing clip sample image with respect to each color are represented by ($LSFD_{R_k}^{(p)}$, $LSFD_{G_k}^{(p)}$, $LSFD_{B_k}^{(p)}$) and ($LSFD_{R_k}^{(m)}$, $LSFD_{G_k}^{(m)}$, $LSFD_{B_k}^{(m)}$). The least square frequency deviations of each color for each present and missing clip image from corresponding color composite matrices are given by Equation (3.54) through Equation (3.59), respectively.

$$LSFD_{R_k}^{(p)} = \sqrt{\sum_{n=0}^{255} (h_{R_k}^{(p)}(n) - h_{R_k}^{(\overline{p})}(n))^2} \tag{3.54}$$

$$LSFD^{(p)}_{G_k} = \sqrt{\sum_{n=0}^{255}(h^{(p)}_{G_k}(n) - h^{(\overline{p})}_{G_k}(n)^2)} \tag{3.55}$$

$$LSFD^{(p)}_{B_k} = \sqrt{\sum_{n=0}^{255}(h^{(p)}_{B_k}(n) - h^{(\overline{p})}_{B_k}(n)^2)} \tag{3.56}$$

$$LSFD^{(m)}_{R_k} = \sqrt{\sum_{n=0}^{255}(h^{(m)}_{R_k}(n) - h^{(\overline{m})}_{R_k}(n)^2)} \tag{3.57}$$

$$LSFD^{(m)}_{G_k} = \sqrt{\sum_{n=0}^{255}(h^{(m)}_{G_k}(n) - h^{(\overline{m})}_{G_k}(n)^2)} \tag{3.58}$$

$$LSFD^{(m)}_{B_k} = \sqrt{\sum_{n=0}^{255}(h^{(m)}_{B_k}(n) - h^{(\overline{m})}_{B_k}(n))^2)} \tag{3.59}$$

As it can be seen in Figure 3.16 the minimum present LSFD error is less than the minimum missing LSFD error, though the maximum present LSFD error is more than the maximum missing LSFD error. The absolute error deviation of the present clip composite image is set as the reference for the comparison with the trial image rather than the missing clip composite image. In short, the large overlap between the two groups of images indicates that the idea of classification of images by this method is weak (63% identification). Similar results are obtained for green and blue colors.

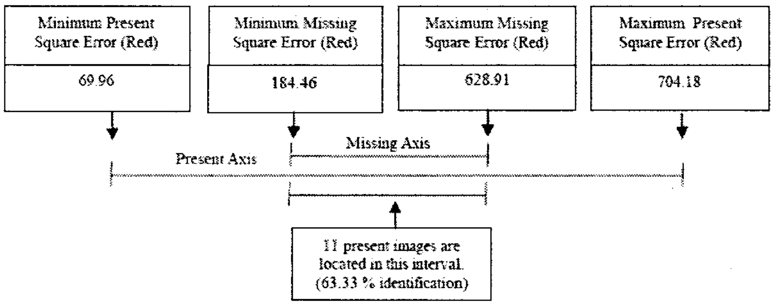

Minimum Present Square Error (Red)
69.96
Minimum Missing Square Error (Red)
184.46
Maximum Missing Square Error (Red)
628.91
Maximum Present Square Error (Red)
704.18
Missing Axis
Present Axis
11 present images are located in this interval. (63.33 % identification)

Figure 3.16: Overlap between least square deviations of present and missing clip images

In general, from the results obtained using the aforementioned approaches, we come to the conclusion that 100% correct identification of present and missing clips cannot be achieved. The best approach misclassified 1 image in 30 images. There might be a possibility that a combination of previous approaches could result in a better identification but it still does not necessarily guarantee full identification.

3.4. Fuzzy Theoretical Foundation

The machinery of fuzzy logic has played and is continuing to play a pivotal role in the computer vision based systems [Zadeh 2008]. Therefore, fuzzy sets and fuzzy logic mainly contribute in classification system and decision making method in this work. In many existing fuzzy systems, fuzzy rules are defined by an expert. Our strategy is to use a technique that would automatically generated such rules based on the sample images introduced in Section 3.3.1. In this section, the theoretical framework for the fuzzy system used in the presented study is briefly given. Eventually, a developed 6-input 1-output fuzzy model will be proposed and applied on the proposed features in Section 3.5. As we mentioned earlier in Section 3.2, the output of the proposed model is to properly detect present and missing clips.

The fuzzy rule-based objective models are constructed from input and output data of the system by using a systematic process with a specific objective function. The constructed fuzzy if-then rules form the fuzzy knowledge-based model of the system. Due to its automatic generation capability of rules, we adopted fuzzy subtractive clustering [Chiu 1994] in identifying the structure of the fuzzy rule base. In addition, first order Sugeno type rules are chosen to represent fuzzy rule outputs and least square estimation method is applied to obtain the optimized output coefficients of the rules. Therefore, an explanation of fuzzy subtractive clustering is summarized in Section 3.4.1. Subsequently, Sugeno model is presented in Section 3.4.2.

3.4.1. Fuzzy Subtractive Clustering Algorithm (FSCA)

In subtractive clustering, a data point x_i is considered to be a cluster center if it possesses the highest potential value, which is a function of the distance measure. The number of fuzzy rules created by a subtractive clustering algorithm and the resulting performance measure (sum of mean square errors) are influenced by the choice of four clustering parameters: $\overline{\varepsilon}$, $\underline{\varepsilon}$, r_a and η. The accept ratio ($\overline{\varepsilon}$) is used to set an upper threshold value above which the algorithm will accept the data as a cluster center. Likewise, the reject ratio ($\underline{\varepsilon}$) is used to set a lower threshold value below which it will reject the data as a new cluster center. If the potential value falls in the gray region between the upper and lower thresholds, we check if the data provides a good trade-off between having a sufficient potential and being sufficiently far from the existing cluster centers. Smaller values of $\overline{\varepsilon}$ and $\underline{\varepsilon}$ will result in a large number of rules and vice versa [Demirli *et al.* 2003]. The positive constant r_a is the Euclidean distance (or radius of cluster hyper-sphere in the data space) defining a neighborhood range of the cluster. Data points outside this radius have little influence on the potential. A large value of r_a results in fewer cluster centers that leads to a coarse model. In the contrary, small value of r_a can produce a large number of rules that may lead to over-defined system. The cluster centers are determined by the following procedure described below:

1. Set the clustering parameters accept ratio ($\overline{\varepsilon}$), reject ratio ($\underline{\varepsilon}$), cluster radius (r_a) and squash factor (η).

2. Consider a collection of n data points $\{x^1, x^2, ..., x^n\}$ in m-dimensional space. The potential of each data point is first estimated by Equation (3.60).

$$P_i = \sum_{j=1}^{n} e^{-\alpha\|x^i - x^j\|^2} \quad \text{for } i = 1....,n \tag{3.60}$$

where, $\alpha = \frac{4}{r_a^2}$, P_i is the potential of the i^{th} data point and n is the total number of data points considered. Select the data point x_i with the maximum potential value as the first cluster center. Set cluster counter $k = 1$, let $c^k = c^1 = \{x^i | P_i \geq P_j, \forall j = 1,...,n\}$ and $P_k^* = P_1^* = P_i$.

3. Once a cluster center c^k is obtained, the data points that are close to c^k are penalized in order to make the emergence of other cluster centers possible. Hence, the potentials of each data point is revised by a subtraction step as shown in Equation (3.61).

$$P_i = P_i - P_k{}^* e^{-\beta \|x^i - c^k\|^2} \tag{3.61}$$

$$\text{for } i = 1,...,n$$

where, $\beta = \frac{4}{r_b{}^2}$ and $r_b = \eta \times r_a$ P_k^* is the potential of the k^{th} cluster center (c^k) and is penalty radius.

4. Now, select x^t with the current maximum potential P_t as the candidate for the next cluster center.

$$x^t = \{x^i | P_t > P_i, \forall i = 1,...,n\} \tag{3.62}$$

5. The process of acquiring new cluster center is based on the comparison of the potential of the candidate cluster center x_t with respect to the upper and lower threshold values, as well as the relative distance criterion. A data point with potential greater than the upper threshold value ($P_t > \overline{\varepsilon}.P_k^*$) is directly accepted as cluster center. However, if the potential P_t is within the gray region ($\underline{\varepsilon}.P_k^* < P_t < \overline{\varepsilon}.P_k^*$) then the acceptance of x^t squarely depends on fulfilling the following criterion:

$$\frac{d_{min}}{r_a} + \frac{P_k^*}{P_t} \geq 1 \tag{3.63}$$

where, d_{min} is the shortest distance between the candidate cluster center x^l and all previously found cluster centers. Under the two mentioned circumstances, if x^l is accepted as a new cluster center, update cluster counter $k \longleftarrow k+1$, cluster center $c^k \longleftarrow x^l$ and repeat the procedure from step 3. Otherwise, reject the candidate cluster x^l and terminate the search for new cluster centers. Once c cluster centers are created in the input and output space, they are projected into each dimension j. Exponential degree of membership functions in the input space are then defined in accordance with Equation (3.64).

$$\mu_j^{ik} = e^{-\alpha \|x_j^i - c_j^k\|^2} \tag{3.64}$$

$$\text{for } i = 1,...,n \;\; j = 1,...,m \;\; k = 1,...,c$$

where, $\alpha = \frac{4}{{r_a}^2}$ $\|x_j^i - c_j^k\|$ is the distance measure between the i^{th} data point and the k^{th} cluster center along the j^{th} dimension.

3.4.2. Sugeno Model

In Sugeno type models, the consequent of a rule can be expressed as a polynomial function of the inputs, and the order of the polynomial also determines the order of the rule. A typical n-input first-order Sugeno model with m rules has the following form:

$$\text{If } X_1 \text{ is } A_{11} \text{ and } X_2 \text{ is } A_{12} \text{ ... } X_n \text{ is } A_{1n} \text{ then} \tag{3.65}$$

$$Z_1 = p_{10} + p_{11}u_1 + p_{12}u_2 + ... + p_{1n}u_n$$

$$\text{If } X_1 \text{ is } A_{21} \text{ and } X_2 \text{ is } A_{22} \text{ ... } X_n \text{ is } A_{2n} \text{ then} \tag{3.66}$$

$$Z_2 = p_{20} + p_{21}u_1 + p_{22}u_2 + ... + p_{2n}u_n$$

$$\vdots$$

$$\text{If } X_1 \text{ is } A_{m1} \text{ and } X_2 \text{ is } A_{m2} \text{ ... } X_n \text{ is } A_{mn} \text{ then} \tag{3.67}$$

$$Z_m = p_{m0} + p_{m1}u_1 + p_{m2}u_2 + ... + p_{mn}u_n$$

where, X_j is defined on U_j in R for $j = 1,2,...,n$. The output level Z_i of each rule is weighted by the firing strength ω_i of the rule. For example, for a min t-norm whose input X_j is u_j, the firing strength is:

$$\omega_i = Min(\mu_{A_{i1}}(u_1), \mu_{A_{i2}}(u_2), ..., \mu_{A_{in}}(u_n)) \tag{3.68}$$

where, $\mu_{A_{ij}}(u_j)$ for $i = 1,2,...,m$ and $j = 1,2,...,n$, is the membership function for input X_j in the i^{th} rule. The final output of the system, W is the weighted average of all rule outputs, computed as Equation (3.69).

$$W = \frac{\sum_{i=1}^{m} \omega_i Z_i}{\sum_{i=1}^{m} \omega_i} \tag{3.69}$$

The optimal consequent parameters ($p_{10},...,p_{n0}$, $p_{21},...,p_{2n}$ and $p_{m2},...,p_{mn}$ coefficients) for a given set of clusters are obtained by the least square estimation (LSE) method. When $Z_k = p_k$ for $k = 1,...,m$, then the model becomes a *singleton consequent* model.

3.5. Proposed Fuzzy Model and Experimental Results

In this section we discuss our proposed features, fuzzy model and corresponding experimental results.

3.5.1. XOR Implication

XOR which is the logical operation exclusive disjunction, also called exclusive, is a type of logical disjunction on two operands that results in a value of true if exactly one of the operands has a value of true. XOR operator is used to generate the 6 features in the antecedent part of the rules. We apply the XOR features to the corresponding digits of two binary strings.

As an example, let $a = 135 = (10000111)_2$ and $b = 150 = (10010110)_2$ representing two pixels values, then $a\, XOR\, b = (00010001)_2 = 17$. If we make a, closer to b such as, $a = 145 = (10010001)_2$ then, $a\, XOR\, b = (00000111)_2 = 7$. So, if the value of pixel a is closer to pixel b, it is likely that $a\, XOR\, b$ gets a smaller value.

As we mentioned earlier in Section 3.3.1, $f_{R_k}(x,y)$ represents the red value of the pixel in the k^{th} present clip sample image at the point (x,y), and $\overline{f}_R(x,y)$ is the red value of the pixel in the present clip composite image at the point (x,y). Now, we can use $f_{R_k}(x,y)\, XOR\, \overline{f}_R(x,y)$ to see if pixels in the present clip sample image are closer to pixels in the present composite image. Similarly, refereing to Section 3.3.1, $g_{R_k}(x,y)$ represents the red value of the pixel in the k^{th} missing clip sample image at the point (x,y), and $\overline{g}_R(x,y)$ is the red value of the pixel in the missing clip composite image at the point (x,y). In a similar fashion, we can use $g_{R_k}(x,y)\, XOR\, \overline{g}_R(x,y)$ to see if pixels in the missing clip sample image are closer to pixels in the missing composite image. The above implication is always true for decimal level and with a minor difference it is "often" true for binary. In other words, in the binary level, it is also possible after reducing the difference of two numbers, the resulting XOR output gets a larger value, but this is less likely. In order to get a better insight to understand to which degree XOR operator is monotonic for a pair of pixels, we ran a simulation test for testing the above approach using triple random numbers. The simulation has been done as follows:

1. Choose three random numbers from 0 to 255 so that $a < b < c$

2. Calculate $(b-a)$, $(c-b)$, $(a\, XOR\, b)$ and $(b\, XOR\, c)$

3.

(a) If $[(b-a)$ is smaller than $(c-b)]$ and $[(a\,XOR\,b)$ is smaller than $(b\,XOR\,c)]$ the experiment is considered as success.

(b) If $[(b-a)$ is smaller than $(c-b)]$ and $[(a\,XOR\,b)$ is larger than $(b\,XOR\,c)]$ the experiment is considered as failure.

(c) If $[(b-a)$ is larger than $(c-b)]$ and $[(a\,XOR\,b)$ is larger than $(b\,XOR\,c)]$ the experiment is considered as success.

(d) If $[(b-a)$ is larger than $(c-b))]$ and $[(a\,XOR\,b)$ is smaller than $(b\,XOR\,c)]$ the experiment is considered as failure.

(e) If $[(b-a)$ is equal to $(c-b)]$ go back to step 1.

After 10 million triple random numbers (a, b and c), the above simulation renders a success rate of 75.0051%. This means that the resulting XOR output for a single pair of pixels (not the whole of image) leads to a right decision 75% of the time. Because, when XOR outputs on pixel pair values are combined using a voting methodology, we expect to obtain good results based on the belief that the majority of pixel pairs are more likely to be correct in their decision. For example, when we have a jury (image) of 12 members (12 pixels) and each member (pixel) separately makes the right decision with a 75% chance, the probability that the majority of members (pixels) in the jury (image) make the right decision increases to

$$C_7^{12}(.75)^7(.25)^5+C_8^{12}(.75)^8(.25)^4+$$

$$C_9^{12}(.75)^9(.25)^3+C_{10}^{12}(.75)^{10}(.25)^2+$$

$$C_{11}^{12}(.75)^{11}(.25)^1+C_{12}^{12}(.75)^{12}(.25)^0=94.44\%$$

Table 3.1: Monotonic region of a typical image

Results	Present image	Missing image
Red channel	84.7 %	70.5 %
Green channel	79.1 %	68.1 %

When we increase the number of pixels to 20, the chance of making the right decisions by the majority of pixel pairs using XOR raises to 98.61%. Statistically speaking, in our application each feature (see Section 3.5.2) is composed of a summation of more than 3011 (51 × 61) pixel pairs (zone of interest) and it doesn't rely only on a single pixel pair. To this end, to provide more details about this application, when a typical present and a typical missing images in different channels (red and green) are taken, the percentages of pixels that fall into monotonic region according to the mentioned test are calculated in Table 3.1. Therefore, we are sufficiently confident that the generated features are the right representative of the image in our application.

3.5.2. Proposed Probabilistic Features

We use XOR operator to generate 6 features. Then, we use these 6 features to develop a 6-input, 1-output fuzzy model. The rational behind these features is to measure the variation among the test images by comparing them against the present and missing composites in different colors. It is also important to note that we are using the same sample which was introduced earlier in Section 3.3.1. The way we have adopted XOR operation and constructed features are explained next.

We perform XOR operation for all the pixels of the present clip sample image with the present clip composite and we sum up the XOR values. This is done for red and green individually (see first and second features′description for present clip sample).

- First feature: the output resulting from implementing the XOR operator on the red present clip composite with present clip sample (red plane) generates the first input (Equation (3.70)).
- Second feature: the output resulting from implementing the XOR operator on the green present clip composite with present clip sample (green plane) generates the second input (Equation (3.71)).

We also perform XOR operation for all the pixels of the present clip sample image with all the pixels in missing clip composite and sum them up. Again, this is done for red and green individually (see third and fourth features′description for present clip sample). It is important to note that the values of red, green are considered separately in their descriptive matrices.

- Third feature: the output resulting from implementing the XOR operator on the red missing clip composite with present clip sample (red plane) generates the third input (Equation (3.72)).
- Fourth feature: the output resulting from implementing the XOR operator on the green missing clip composite with present clip sample (green plane) generates the fourth input (Equation (3.73)).

Since, the blue frequency analysis for present and missing clip images in our sample did not provide any meaningful information to classify the images, we have used only red and green reference colors to determine the pixel values in the input data sets.

We also perform XOR operation for all the pixels of the present clip sample image with white (255) and black (0) image and sum them up (see fifth and sixth features′

description for present clip sample). White and black XOR operators are used as supplements in the clip sample in order to handle situations specially when the clip is missing but still the present clip composite is not significantly larger than the missing clip composite. The reason is that in this case the missing background is white or at least a large part of the background is light gray or white.

- Fifth feature: the output resulting from implementing the XOR operator on one (white page) with present clip sample (three planes) generates the fifth input (Equation (3.74)).
- Sixth feature: the output resulting from implementing the XOR operator on zero (black page) with present clip sample (three planes) generates the sixth input (Equation (3.75)).

Six sets of input data are generated by applying the Equation (3.70) to Equation (3.75) for present clip images in the sample space. To obtain more background on the used notations in the following formulas, the reader is referred to Section 3.3.1.

$$\sum_{x=1}^{N_x} \sum_{y=1}^{N_y} (f_{R_k}(x,y)\ XOR\ \overline{f}_R(x,y)) \tag{3.70}$$

$$\sum_{x=1}^{N_x} \sum_{y=1}^{N_y} (f_{G_k}(x,y)\ XOR\ \overline{f}_G(x,y)) \tag{3.71}$$

$$\sum_{x=1}^{N_x} \sum_{y=1}^{N_y} (f_{R_k}(x,y)\ XOR\ \overline{g}_R(x,y)) \tag{3.72}$$

$$\sum_{x=1}^{N_x} \sum_{y=1}^{N_y} (f_{G_k}(x,y)\ XOR\ \overline{g}_R(x,y)) \tag{3.73}$$

$$\sum_{x=1}^{N_x}\sum_{y=1}^{N_y}((f_{R_k}(x,y)\, XOR\, 255) + (f_{G_k}(x,y)\, XOR\, 255) + (f_{B_k}(x,y)\, XOR\, 255)) \quad (3.74)$$

$$\sum_{x=1}^{N_x}\sum_{y=1}^{N_y}((f_{R_k}(x,y)\, XOR\, 0) + (f_{G_k}(x,y)\, XOR\, 0) + (f_{B_k}(x,y)\, XOR\, 0)) \quad (3.75)$$

Similarly, the inputs from missing clip sample images have been calculated by Equation (3.76) to Equation (3.81).

$$\sum_{x=1}^{N_x}\sum_{y=1}^{N_y}(g_{R_k}(x,y)\, XOR\, \overline{f}_R(x,y)) \quad (3.76)$$

$$\sum_{x=1}^{N_x}\sum_{y=1}^{N_y}(g_{G_k}(x,y)\, XOR\, \overline{f}_G(x,y)) \quad (3.77)$$

$$\sum_{x=1}^{N_x}\sum_{y=1}^{N_y}(g_{R_k}(x,y)\, XOR\, \overline{g}_R(x,y)) \quad (3.78)$$

$$\sum_{x=1}^{N_x}\sum_{y=1}^{N_y}(g_{G_k}(x,y)\, XOR\, \overline{g}_R(x,y)) \quad (3.79)$$

$$\sum_{x=1}^{N_x}\sum_{y=1}^{N_y}((g_{R_k}(x,y)\, XOR\, 255) + (g_{G_k}(x,y)\, XOR\, 255) + (g_{B_k}(x,y)\, XOR\, 255)) \quad (3.80)$$

$$\sum_{x=1}^{N_x}\sum_{y=1}^{N_y}((g_{R_k}(x,y)\, XOR\, 0) + (g_{G_k}(x,y)\, XOR\, 0) + (g_{B_k}(x,y)\, XOR\, 0)) \quad (3.81)$$

We define the output of the training model for present clip sample images as one, and the output for missing clip sample images as zero. Therefore, O_i is characterized as a binary value for the output of the i^{th} image in the model. The corresponding equation is as follows:

$$O_i = \begin{cases} 1; & \text{if the clip is present} \\ 0; & \text{otherwise} \end{cases} \quad (3.82)$$

It should be noted that the six features can also be calculated by using their decimal values which is very time consuming calculations. Our model carries out the calculations in binary level which is computationally less costly. Although, XOR is a very fast binary level operator, it is not entirely a monotonic operator in a pixel by pixel basis for the first four features (inputs) but it is probabilistically monotonic for the overall image.

3.5.3. Proposed Fuzzy Model

Training data from 60 randomly sample images (30 present clips and 30 missing clips) is generated by Equations (3.70) to (3.81). A small portion of our training data is shown in Table 3.2. Then the subtractive clustering algorithm and LSE method are used (as explained in Section 3.4.2) to identify first order Sugeno models. An enumerative search on the subtractive clustering parameters is performed [Demirli and Muthukumaran 2000, Demirli *et al.* 2003] and the following values for the subtractive clustering parameters, which give the following 9-rule model, are chosen: $r_a = 0.2$, $\eta = 3$, $\overline{\varepsilon} = 0.4$, $\underline{\varepsilon} = 0.1$ (see 3.4.1).

$$\text{If } X_1 \text{ is } A_{i1} \text{ and } X_2 \text{ is } A_{i2} \text{ ... } X_n \text{ is } A_{i6} \text{ then} \tag{3.83}$$

$$S_i = p_{i0} + p_{i1}V_1 + p_{i2}V_2 + \ldots + p_{i6}V_6 \quad i = 1,..,9$$

One implementation of this combination is fuzzy tool box in Matlab. This is the primary learning method used in this study. Membership functions are shown in Figure 3.17. The centers of Gaussian input membership functions are given in Table 3.3. The sigma (σ) for all the Gaussian membership functions is chosen as 0.0707. V_j for $j = 1,2,...,6$, are the six features that we have identified in Section 3.5.2. p_{ij} are the consequent parameters which are optimized by LSE and S_i for $i = 1,2,...,9$ are the rule outputs. The consequent parameters are given in Table 3.4. The training time for our fuzzy system is about 15 minutes on a Pentium(R) 4 CPU 2.99GHz, 1.00GB RAM and XP platform.

Table 3.2
Small portion of training data

Image (NO)	Input 1	Input 2	Input 3	Input 4	Input 5	Input 6	Output
Present clip 1	475577	479585	566056	416667	2236987	1620866	1
Present clip 2	330950	751845	359982	680299	1609425	2137788	1
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
Present clip 30	299283	706046	354126	645983	1240609	2441581	1
Missing clip 1	655556	363923	617657	372420	3078893	927389	0
Missing clip 2	655681	467487	761132	477606	1809612	1972894	0
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
Missing clip 30	560678	398962	540826	392979	2483400	1417896	0

Table 3.3
Centers of Gaussian membership functions (μ)

Cluster	First input	Second input	Third input	Fourth input	Fifth input	Sixth input
(No)	(center)	(center)	(center)	(center)	(center)	(center)
Rule 1	0.887	0.089	0.803	0.112	0.769	0.230
Rule 2	0.164	0.943	0.125	0.982	0.189	0.810
Rule 3	0.790	0.317	0.663	0.353	0.961	0.038
Rule 4	0.370	0.723	0.450	0.687	0.041	0.958
Rule 5	0.963	0.849	1	0.893	0.717	0.282
Rule 6	0.799	0.437	0.939	0.539	0.310	0.689
Rule 7	0.459	0.460	0.552	0.413	0.522	0.477
Rule 8	0.127	0.889	0.132	0.889	0.029	0.970
Rule 9	0.620	0.307	0.502	0.364	0.644	0.355

3.5.4. Experimental Results

Figure 3.18 shows the absolute error of the Sugeno model in training phase. The first 30 images are present clip images and the second 30 images are missing clip images. The

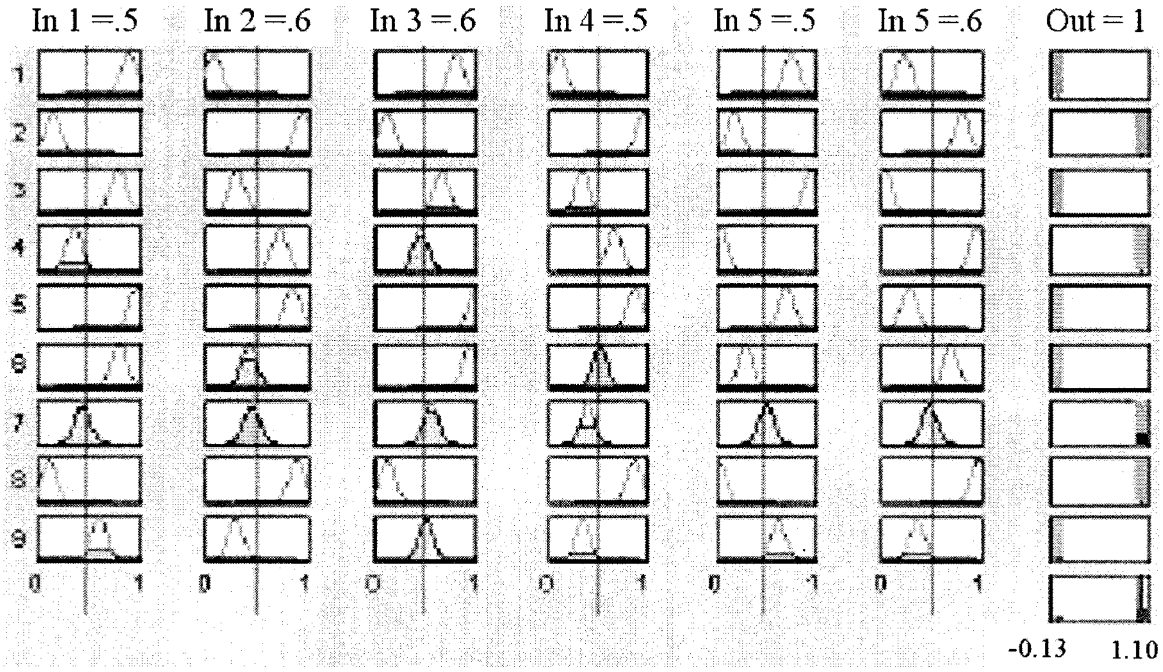


Figure 3.17: Fuzzy rule based with nine rules

largest error in training phase is less than 2×10^{-15} among 60 images. It illustrates that the model can easily detect all the images in the sample space and therefore demonstrates its strong capability to be trained by earlier used features.

Finally, applying the fuzzy first order Sugeno model to the whole database, composed of 1910 images, we get 100% correct identification rate of all the images. The largest absolute error in 1910 images is 8E-3. The average absolute error in 1910 images is obtained as E-5. For the sake of clarification, we also draw your attention to the fact that a distinction should always be made between identification rate (accuracy rate) and error of the model. In the sense that accuracy rate in machine vision, simply reflects the percentage of success while the error of the model is a difference between actual measurement and expected measurement (here 1 for present clips and 0 for missing ones).

With a careful look at the first order Sugeno parameters in Table 3.4, it appears that

Table 3.4
Sugeno parameters

	p_{i0}	p_{i1}	p_{i2}	p_{i3}	p_{i4}	p_{i5}	p_{i6}
Rule 1	1.78E-13	1.50E-14	5.74E-15	6.79E-14	5.84E-14	0	-2.52E-13
Rule 2	1	-1.37E-15	2.68E-14	1.35E-14	8.46E-15	0	-3.06E-14
Rule 3	-2.52E-10	-3.67E-10	1.52E-09	5.61E-10	0	4.37E-10	-1.08E-09
Rule 4	1	-4.45E-14	1.40E-14	4.92E-14	-4.22E-15	0	-2.54E-14
Rule 5	0	0	2.92E-30	0	0	0	0
Rule 6	-4.35E-16	0	0	0	0	0	0
Rule 7	1	0	0	0	0	0	0
Rule 8	1	-1.48E-16	-6.07E-15	-3.60E-16	5.88E-15	0	4.59E-15
Rule 9	-7.26E-03	1.80E-05	-1.64E-02	6.80E-03	0	-2.75E-03	2.22E-02

only p_{i0} parameters have significant values. Furthermore, having an error as small as E-5 for the average absolute error in 1910 images is considered negligible.

We now explore the possibility of using a coarser model (singleton consequent model) to achieve 100% correct identification. For this purpose, we propose a nine-rule singleton consequent fuzzy model as follows.

$$\text{If } X_1 \text{ is } A_{i1} \text{ and } X_2 \text{ is } A_{i2} \ldots X_n \text{ is } A_{i6} \text{ then} \tag{3.84}$$

$$S_i = p_i \quad i = 1,..,9$$

The input membership functions for singleton consequent model are the same as the first order Sugeno model (Figure 3.17). The centers of Gaussian input membership functions are given in Table 3.3. The sigma (σ) of all the Gaussian membership functions is chosen as 0.0707. The consequent parameters of the singleton consequent model are again optimized by using LSE and are given in Table 3.5.

Figure 3.19 shows the error of the singleton consequent model in training phase. The first 30 images are present clip images and the second 30 images are missing clip images. The largest error in training phase 1.3E-3.

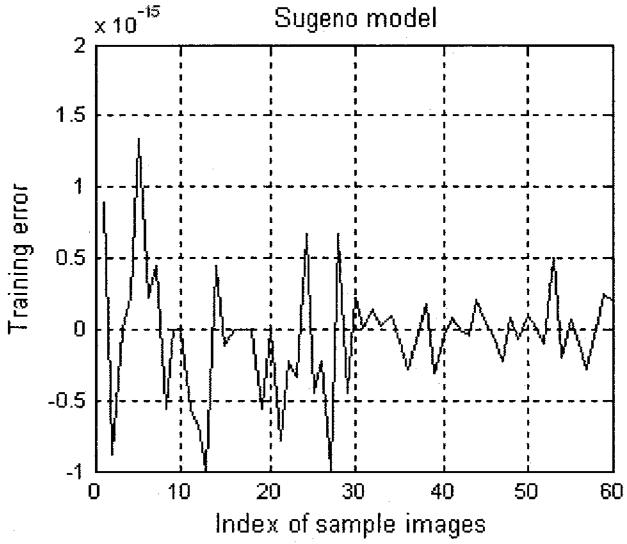


Figure 3.18: Sugeno training error phase for sixty images

Table 3.5
Singleton parameters

Parameters	values
p_1	-3.17e-07
p_2	1.00
p_3	1.93e-09
p_4	1.00
p_5	-3.99e-29
p_6	-1.43e-05
p_7	1.00
p_8	1.00
p_9	-0.01e-01

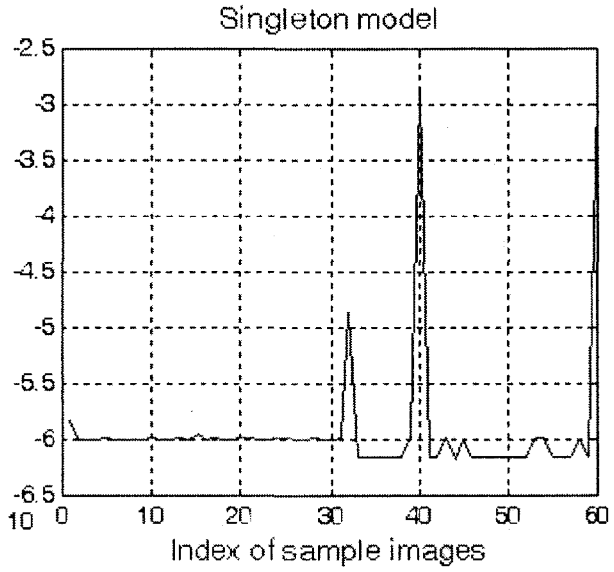


Figure 3.19: Singleton training error phase for sixty images

Lastly, applying the fuzzy singleton model to the whole database composed of 1910 images, we get 100% correct identification rate of all the images. The largest absolute error in 1910 images is 3.2E-3. The average absolute error in 1910 images is obtained as 5E-5.

There are two remarks in terms of comparison between first order Sugeno and singleton consequent model. First, as we expected, singleton consequent model has larger error than the first order Sugeno model. Second, the variance of error in the singleton consequent model is bigger than the variance of error in the first order Sugeno model. The reason is that singleton model has only 9 consequent parameters to be trained to model the system behavior, whereas this number for first order Sugeno model is 63 (both models are composed of nine rules). Therefore, the singleton model is considered a coarse model compared with the first order Sugeno model but it is easier to be trained due to the fewer number of its parameters. However, both models yield 100% correct identification in the entire database.

3.6. Conclusion

In this chapter, we investigated different statistical approaches including mean, variance, Euclidean distance, correlation, absolute error and least square error. We demonstrated that none of these approaches lead to 100% correct identification of missing clips. We proposed a novel fuzzy machine vision based inspection method for clip detection. After determination of the interest zone, which is almost a must for any automated vision system, the rest of the machine vision steps are done automatically. No special setting or direct threshold has been applied in our fuzzy model. Our clip detection model relies on fuzzy color processing using composite images and developing XOR feature extractor. The developed model completely detects present and missing clips among 1910 industrial images from a Canadian automotive assembly line. We believe that our proposed model is a reliable alternative for machine vision based inspection in adverse industrial situations where it is not possible to sufficiently control the environment. When it is required to identify images with a huge amount of changes in light and background, our proposed model conducts a strong performance.

Chapter 4

Fuzzy Clip Position Identification

4.1. Introduction

The majority number of approaches to the orientation detection problem proposed in the last three decades have been developed in strongly controlled environments. However, in presence of more realistic and uncontrolled conditions the performance drastically dropped, making most of the approaches inapplicable in real conditions [Francoa and Nanni 2009]. As a result, automatic image orientation detection is an important, yet extremely challenging task in industry. However, to be attractive, such inspection system should be also cost effective.

Humans use object recognition and contextual information to identify the correct orientation of an image. Their vision system can adapt to varying lighting conditions and ignore irrelevant information. However, humans are quite good at inspection tasks for only short periods of time [Killing *et al.* 2007]. Unfortunately, it is difficult for a computer to perform the task in the same way because current object recognition algorithms are extremely limited in their scope and robustness [Boutell and Luo 2005, Chien 1995]. For instance, methods have been proposed in order to solve the automatic license plate recognition (ALPR) problem. The presence of abundant edges, especially vertical edges, in license plate regions is used to generate the candidate regions for orientation detection and classification in [Jia *et al.* 2007].

Another kind of popular method is Hough Transform. It focuses on detecting the frames which hold the border lines of license plates is extensively used [Yanamura *et al.* 2003]. This method is not suitable for conditions in which borders are not clear due to damages or dirt. Moreover, the computational complexity is another disadvantage of this method [Duan *et al.* 2005]. Color is a distinctive feature which can be used for frame orientation detection [Lee and Kim 1994]. It is a powerful feature for object recognition. However, the sensitivity of this feature to the illumination condition and the quality of imaging system has been limited its usage.

However, the state of the art computer vision techniques still cannot infer the high-level knowledge abstraction of the objects in the real world [Shapiro and Stockman 2000]. Therefore, there is a need to use more sophisticated approaches to assist computers in representing and processing the images. The literature on image orientation detection is rather sparse. Most of the literature has emphasized related topics such as page orientation detection [Caprari 2000, Chaudhuri and Pal 1997, Le *et al.* 1994, Yu and Jain 1996].

The existing orientation detection methods were built upon low-level vision features such as spatial distributions of color and texture relying on learning patterns from a training set [Boutell and Luo 2005, Risinger and Kaikhah 2008]. Although the perception of the orientation in some circumstances is not clear, in this study, due to rectangular shapes of clips, there is a common sense towards the direction of each clip.

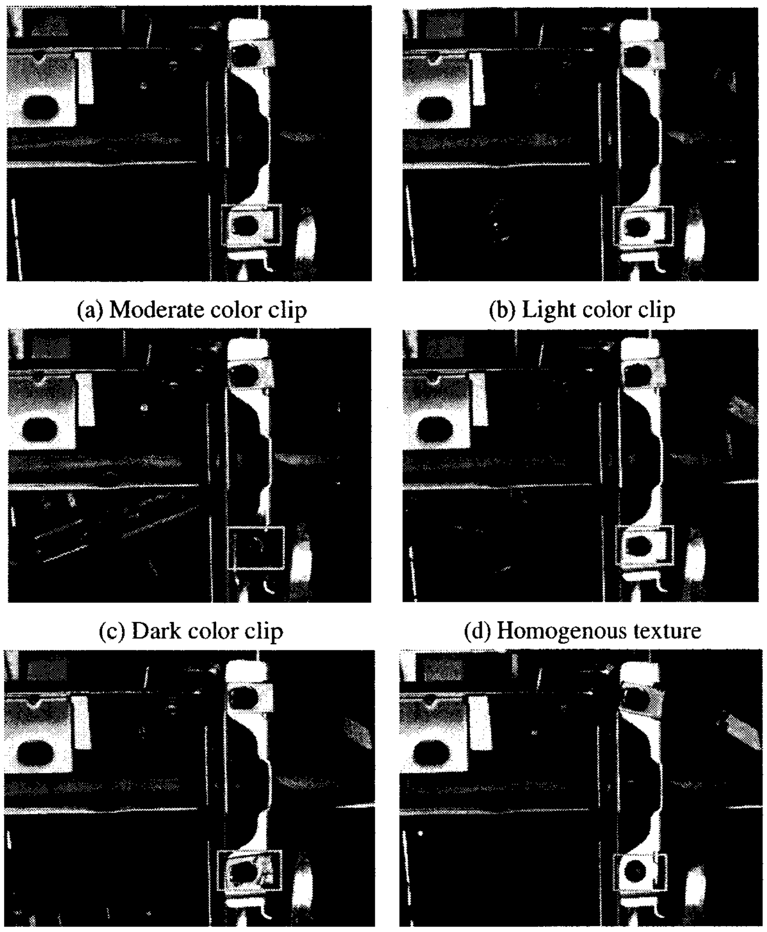
To represent a certain clip orientation in the image as acceptable, we have developed fuzzy models. Fuzzy set and fuzzy logic theory is one of the versatile tools for soft computing intelligence, which deals with the design of flexible information processing center [Chen and Chen 2005, Kulkarni and Cavanaugh 2000, Yager 1992]. Fuzzy logic has also significantly improved the capability of visual inspection techniques [Rao 2001]. In short, fuzzy models due to their summarization and information compression through the use of granulation capabilities [Zadeh 2008], are functional tools not only in quantifying the evidence from partially developed patterns but also combining evidence from different patterns to identify underlying causes. The proposed fuzzy framework in this study is an attempt to bridge the gap between computer and human vision systems. This chapter improves the inspection component of the manufacturing process through the application of fuzzy techniques.

4.2. Clip Orientation Detection Problem

An inspection problem faced by a Canadian automotive parts manufacturer is being used as a case study. The problem involves a vision system that is being used to confirm the placement of metal fastening clips on a structural member that supports a truck dash panel. Five teams from five different universities across Canada were working on this problem [Mehran *et al.* 2006]. It took the manufacturer over eight months to tune the machine vision system to detect missing clip fasteners [Killing and Mechefske 2006]. The trained system is still not able to achieve the performance desired by the company of less than 0.02% false detection rate [Killing and Mechefske 2006].

The objective of this chapter is to develop a model in order to identify whether there are any clips available in the position of interest, and if any, in terms of clip orientation whether it is properly installed or not. For instance, Figure 4.1(a) depicts a situation where two clips are present and correctly installed.

Our emphasis in this work is only on the bottom part of the depicted images. Since there is no significant difference between top clips and bottom ones, the top ones could be analyzed with the same approach. Neither the location of the clip nor its color on each



(a) Moderate color clip

(b) Light color clip

(c) Dark color clip

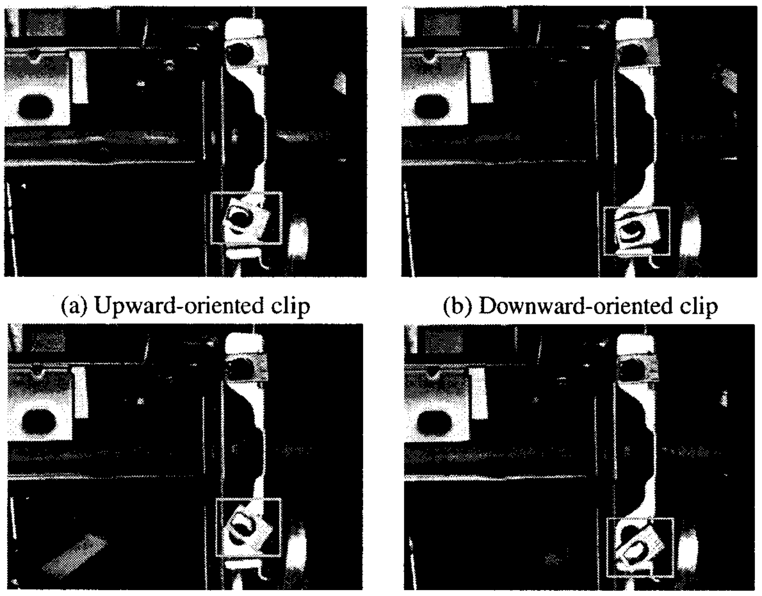
(d) Homogenous texture

(e) Non homogenous texture

(f) Missing clip

Figure 4.1: Properly installed present clips and a missing clip

image is clearly defined. In this sense, the challenging part of the problem is to deal with ill-defined objects. There are quite large differences in the position of the clips within the images among different samples, largely due to the changes in the position of the camera. A moving robot arm is also in the background. The lack of consistency could bring a new problem for any automatic image registration method. Variation of the light and color of clips adds additional difficulties to the problem at hand. To see the complexity of this problem in terms of color variation, Figure 4.1(a) and 4.1(b) show a moderate color clip and a light color clip, respectively, while Figure 4.1(c) illustrates a dark color clip, with



(a) Upward-oriented clip (b) Downward-oriented clip

(c) Extremely upward-oriented clip (d) Extremely downward-oriented clip

Figure 4.2: Improperly installed clips

its slight movement in the frame. Lack of homogeneity in texture of clips also adds to the difficulties of the problem. Figure 4.1(d) and Figure 4.1(e) are only two examples that show homogenous and non homogenous textures for clips.

Nevertheless, the clip sometimes is missing. Figure 4.1(f) shows a typical missing clip image. The most challenging case is when the clip is present but installed improperly. Figure 4.2(a), Figure 4.2(b), Figure 4.2(c) and Figure 4.2(d) show some of the typical improperly installed clip images.

In this case after detection of the existence of the clip, the model should determine whether the ordination of clip is as good as the orientation of clips in Figures 4.1(a) - (e) or not. The original pictures are taken in color (640 × 480 RGB). In our database we have 708 images. While 685 images belong to properly installed clips, 21 of them belong to improperly installed and the rest are missing clips.

4.3. Methodology

In this section, we describe HSL color space (Section 4.3.1), Canny edge detector method (Section 4.3.2) and finally Sugeno model using fuzzy subtractive clustering algorithm (Section 4.3.3), to provide relevant background for the developed fuzzy models.

4.3.1. Color Space

Hue, saturation and light values are three elements of an HSL system. Hue is the wavelength (color) of maximum apparent intensity, though we have to take into account that the eyes see a combination of discrete wavelengths as if they were an intermediate color. Saturation is the purity of the color. A strong red is more saturated than a pink of the same hue, because the pink will contain more white, i.e. light of other wavelengths. Light is the amount of luminance. This represents a wealth of similar color spaces, alternative systems include HSI (hue, saturation and intensity), HSV (hue, saturation and value), HCI (hue, chroma / colorfulness and intensity), HVC (hue, value and chroma) and HSD (hue, saturation and darkness). Most of these color spaces are non-linear transforms from RGB. Their advantage lies in the extremely intuitive manner of specifying color. It is possible to select a desired hue and then modify it slightly by adjustment of its saturation and intensity. The separation of the luminance component from chrominance (color) information is stated to have advantages in applications such as image processing and pattern recognition. However the exact conversion of RGB to HSL system depends entirely on the equipment characteristics. It is important to take into account that there are different transforms of RGB to HSL alternatives, each of which is claimed to be better for specific applications than the others [Levkowitzh and Herman 1993]. Given RGB values, equations for mapping RGB to HSL are:

$$H = \begin{cases} \theta; & \text{if } B \leq G \\ \\ 360 - \theta; & \text{if } B > G \end{cases} \tag{4.1}$$

where,

$$\theta = Cos^{-1}\{\frac{\frac{1}{2}[(R-G)+(R-B)]}{[(R-G)^2+(R-B)(G-B)]^{\frac{1}{2}}}\}\ ; \tag{4.2}$$

$$S = 1 - \frac{3}{R+G+B}\ [min(R,G,B)]\ ; \tag{4.3}$$

$$L = \frac{R+G+B}{3}\ . \tag{4.4}$$

4.3.2. Canny Edge Detection Method

Experiments on the human visual system show that edges in images are extremely important [Lua *et al.* 2003]. Edges are also important features widely used in image processing, pattern recognition, and computer vision. A local edge is a small area in images where the local gray levels change rapidly in a simple way.

A number of methods for edge detection implementing different approaches to the digital calculation of the derivative of the image intensity function are available in the literature. The classical methods such as the Sobel, Prewitt and Kirsch detectors calculate the first directional derivative to determine the locations of the edges [Albovik and Daly 1998]. These detectors are simple to implement but they are usually inaccurate and highly sensitive to noise. The zero-crossing edge detectors use the second derivative along with the Laplacian operator. Though, these detectors have set edge detection characteristics in all directions, they are also very sensitive to noise [Umbaugh 1998]. The Gaussian edge detectors diminish the undesirable negative effects of noise by smoothing the image before performing edge detection. Hence, they exhibit much superior performance over other operators especially in noisy conditions. The Gaussian detectors demonstrate a fairly better performance and they are computationally much more efficient than classical derivative based edge detectors. Canny edge detector, which is a Gaussian edge detector, has been widely used in many applications. It is one of the the most practical used algorithms [Hocenski *et al.* 2006].

The Canny operator works in a multi-stage process [Canny 1986, Deriche 1987]. First of all the image is smoothed by Gaussian convolution. Then a simple 2-D first derivative operator is applied to the smoothed image to highlight regions of the image with high first spatial derivatives. Edges give rise to ridges in the gradient magnitude image. The algorithm then tracks along the top of these ridges and sets to zero all pixels that are not actually on the ridge top so as to give a thin line in the output, a process known as non-maximal suppression. The tracking process exhibits hysteresis controlled by two thresholds: T_1 and T_2 with $T_1 > T_2$. Tracking can only begin at a point on a ridge higher than T_1. Tracking then continues in both directions out from that point until the height of the ridge falls below T_2. This hysteresis helps to ensure that noisy edges are not broken up into multiple edge fragments. The purpose of the Canny edge detection method is to detect edges with noise suppressed at the same time. The Canny method operates as follows:

1. Smooth the image with a Gaussian filter to reduce noise and unwanted details and textures.

$$g(m,n) = G_\sigma(m,n) \times f(m,n) \tag{4.5}$$

where,

$$G_\sigma(m,n) = \frac{1}{\sqrt{2\pi\sigma^2}}\, exp^{[-\frac{m^2+n^2}{2\sigma^2}]} \tag{4.6}$$

2. Compute the gradient of $g(m,n)$ using any of the gradient operators (Roberts, Sobel, Prewitt, etc) to get:

$$M(m,n) = \sqrt{g_m^2(m,n) + g_n^2(m,n)} \tag{4.7}$$

$$\theta(m,n) = Tan^{-1}[\frac{g_n(m,n)}{g_m(m,n)}] \tag{4.8}$$

3. Define $M_T(m,n)$ as

$$M_T(m,n) = \begin{cases} M(m,n); & \text{if } \ M(m,n) > T \\ \\ 0; & \text{otherwise} \end{cases} \tag{4.9}$$

where, T is so chosen that all edge elements are kept while most of the noise is suppressed.

4. Suppress non-maxima pixels in the edges in M_T, obtained above to thin the edge ridges (as the edges might have been broadened in step 1). To do so, check to see whether each non-zero $M_T(m,n)$ is greater than its two neighbors along the gradient direction $\theta(m,n)$. If so, keep $M_T(m,n)$ unchanged, otherwise, set it to 0.

5. Tight the previous result by two different thresholds τ_1 and τ_2 (where $\tau_1 < \tau_2$) to obtain two binary images T_1 and T_2. Note that compared to T_1, T_2 has less noise and fewer false edges but larger gaps between edge segments.

6. Link edges segments in T_2 to form continuous edges. To do so, trace each segment in T_2 to its end and then search its neighbors in T_1 to find any edge segment in T_1 to bridge the gap until reaching another edge segment in T_2.

4.3.3. Fuzzy Theoretical Foundations

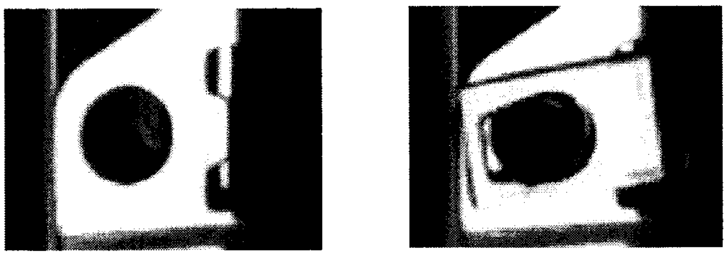
The fuzzy rule-based objective models are constructed from input and output data of the system by using a systematic process with a specific objective function. The constructed fuzzy if-then rules form the fuzzy knowledge-based model of the system. Due to its automatic generation capability of rules, we adopted fuzzy subtractive clustering [Chiu 1994] in identifying the structure of the fuzzy rule base. In addition, first order Sugeno type rules are chosen to represent fuzzy rule outputs and least square estimation method is applied to obtain the optimized output coefficients of the rules. To obtain more background on fuzzy subtractive clustering algorithm (FSCA) and Sugeno model the reader is referred to 3.4.1. and 3.4.2., respectively.

4.4. Fuzzy Models and Experimental Results

Section 4.4.1 has been developed to identify whether there are any clips available in the position of interest or not. In case any clip is available, Section 4.4.2 determines whether the clip is properly or improperly installed.

4.4.1. Fuzzy Clip Existence Detection Model

We took a sample consisting of 30 present clip images and 2 missing clip images (the number of all the missing clip images in our database). In our database the registered image is cropped and a sub-image is taken in order to reduce processing time since the clip is confined to a small area. By adopting Gaussian distribution for the position of clip and using the standard deviation of a sample of size 30 with three sigma limits, one time horizontally and one time vertically, it makes it possible to determine zone of interest. This operation has been done for top left (TL_x, TL_y), top right (TR_x, TR_y), bottom left



(a) A typical missing clip (clip base) (b) A typical present clip

Figure 4.3: Panel of the clips

(BL_x, BL_y) and bottom right (BR_x, BR_y) corner points of clip, using the following Equations.

$$TL_x = \overline{TL}_x - 3 \times TL_{\sigma_x} \tag{4.10}$$

$$TL_y = \overline{TL}_y - 3 \times TL_{\sigma_y} \tag{4.11}$$

where, $\overline{TL}_x$ and $\overline{TL}_y$ are the average of top left corner point clip positions in the sample with respect to x and y axes, respectively. Additionally, TL_{σ_x} and TL_{σ_y} are the standard deviation of top left corner point clip positions in the sample with respect to x and y axes, respectively. In the similar fashion, the other three corner points (top right, bottom left and bottom right) are calculated. The special characteristic of number 30 arises from the Central Limit Theorem (CLT) which helps us to assume that the corner points of clips are normally distributed. Figure 4.3 shows two typical images, one taken from missing clip sample (clip base) and the other from present clip sample. By constructing the histogram of hue value percentage for all the images in the present and missing clip samples, we have observed that the number of pixels whose hue values are less than one percent in any present clip sample image is always less than the number of pixels whose hue values are less than one percent in any missing clip sample image. In other words, the minimum number of pixels with less than one percent hue value in missing clip sample is always larger than the maximum number of pixels with less than one percent hue value in present clip sample. This is a very strong feature for differentiation according to our sample result. Figure 4.4 shows the numbers of pixels with respect to their percentage of hue value for a typical missing clip and a present clip. You may notice that the number of pixels with less than one percent hue value in missing and present clip images are about 3500 and 1600, respectively. Since using only one feature increases the risk of misidentification, we are looking for a second feature. We use the previous sample again. This time, we build the histogram of saturation value for all the images in the sample. We have observed that the number of pixels whose saturation values are less than one percent in any present clip sample image is always less than the number of pixels whose saturation values are less than one percent in any missing clip sample image. In other words, the minimum number of pixels with less than one percent saturation value in missing clip sample is always larger than the maximum number of pixels with less than one percent saturation value in present clip sample. This feature is also a strong feature for present and missing clip differentiation in the entire sample.

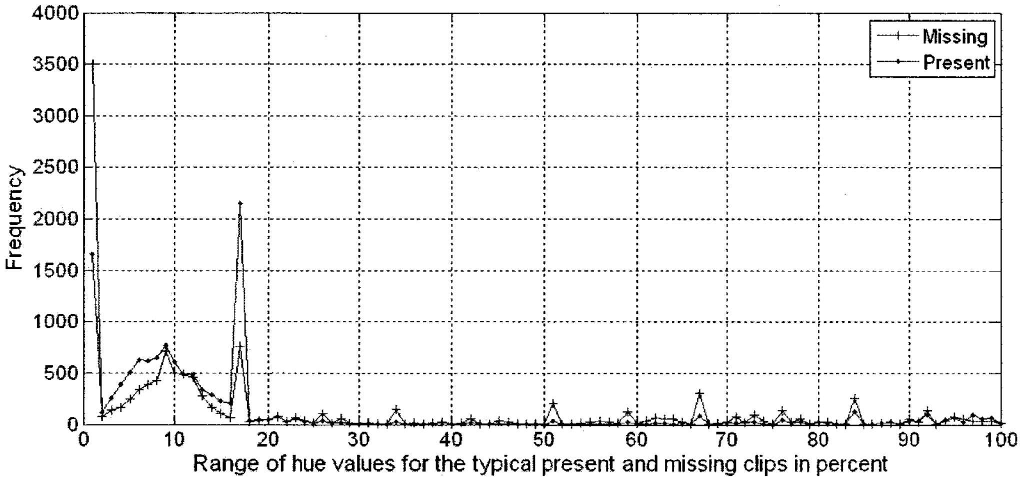


Figure 4.4: Percentage of hue values of a present and a missing clip

Figure 4.5 shows the numbers of pixels with respect to their percentage of saturation value for a typical missing clip and a present clip. It is obvious that the number of pixels with less than one percent saturation value in missing and present clip images are about 3150 and 1400, respectively. Let us call Ω_1, Ω_p and Ω_m as the set of all the present clip sample images, the set of images in present clip sample and the set of images in missing clip sample, respectively.

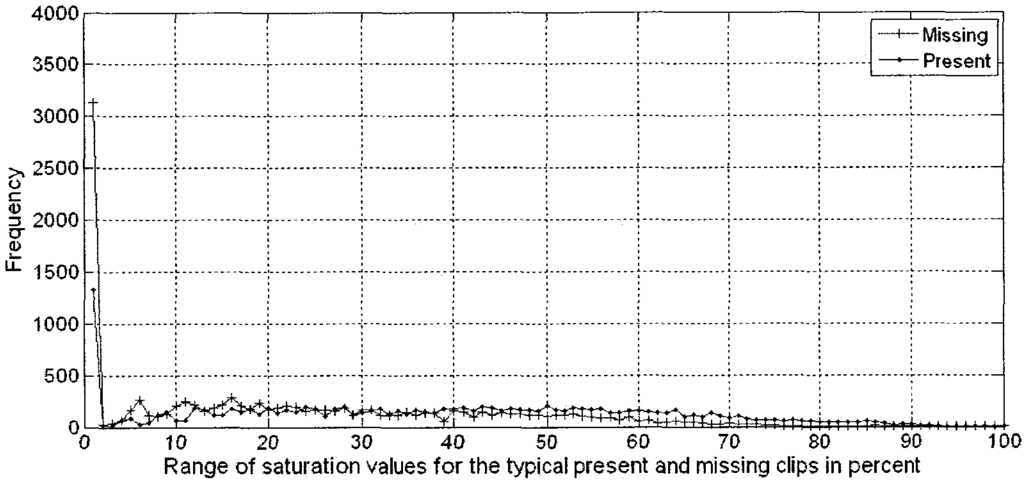


Figure 4.5: Percentage of saturation values of a present and a missing clip

$$\Omega_1 = \Omega_p \bigcup \Omega_m, \tag{4.12}$$

We define H_{ij} and S_{ij} as the percentage of hue and saturation values of the j^{th} pixel in the i^{th} image, respectively. Using Equations (4.13) and (4.14), those pixels that their hue and saturation values is less than one percent are determined for each image, respectively.

$$P_{ij}^{(1)} = \begin{cases} 1; & \text{if} \quad H_{ij} \leq .01 \quad i \in \Omega_1, \; j = 1, ..., N \\ & \\ 0; & \text{if} \quad H_{ij} > .01 \end{cases} \tag{4.13}$$

$$P_{ij}^{(2)} = \begin{cases} 1; & \text{if} \quad S_{ij} \leq .01 \quad i \in \Omega_1, \ j = 1, ..., N \\ \\ 0; & \text{if} \quad S_{ij} > .01 \end{cases} \tag{4.14}$$

where, N is the number of pixel in the zone of interest. Equation (4.15) shows the number of pixels with a hue value less than one percent in the i^{th} image. In a similar fashion, Equation (4.16) shows the number of pixels with a saturation value less than one percent in the i^{th} image, where $X_i^{(1)}$ and $X_i^{(2)}$ are the first and second features in the i^{th} image, respectively.

$$X_i^{(1)} = \sum_{j=1}^{N} P_{ij}^{(1)} \quad i \in \Omega_1 \tag{4.15}$$

$$X_i^{(2)} = \sum_{j=1}^{N} P_{ij}^{(2)} \quad i \in \Omega_1 \tag{4.16}$$

We define the output of training model for present clip images as one, and the output for missing clip images as zero. Therefore, Y_i is characterized as a binary value for the output of the model as follows:

$$Y_i = \begin{cases} 1; & \text{if the clip is present} \\ \\ 0; & \text{if the clip is missing} \end{cases} \tag{4.17}$$

A small portion of our training data is shown in Table 4.1. After using the 32 images in our sample as training data in subtractive clustering based modeling, we identify a fuzzy model with three rules. Our subtractive clustering parameters are: ($r_a = 0.15$, $\eta = 1.8$, $\overline{\varepsilon} = 0.5$, $\underline{\varepsilon} = 0.1$) and the rules are depicted in Figure 4.6.

The centers of Gaussian input membership functions are given in Table 4.2. The sigmas (σ_1 and σ_2) for the first and second features are chosen as 0.058 and 0.064, respectively. a_{ij} for $i = 1, 2, 3$ and $j = 0, 1, 2$ are the consequent parameters which are optimized by least square error (LSE). The consequent parameters are given in Table 4.3.

Table 4.1: Small portion of training data

Image (NO)	Input 1	Input 2	Output
Present clip 1	1741	1441	1
Present clip 2	1720	1354	1
⋮	⋮	⋮	⋮
Missing clip 1	3495	3131	0
Missing clip 2	3278	3002	0
⋮	⋮	⋮	⋮

Table 4.2: Centers of Gaussian membership functions (μ)

Cluster	First input	Second input
(No)	(center)	(center))
Rule 1	0.414	0.046
Rule 2	0.146	0.179
Rule 3	0.549	0.610

Figure 4.7 shows the absolute error of Sugeno model in training phase. The first 30 images are present clip images and the last 2 images are missing clip images. The largest error in training phase is less than 2×10^{-13} among 32 images. It illustrates that the model can easily detect all the images in the sample space and therefore demonstrates its strong capability to be trained by above mentioned features. The training time for our fuzzy system is about 3 minutes on a Pentium (R) 4 CPU 2.99 GHz, 1.00 GB RAM and XP platform.

By applying the model to the whole database, we get 100% correct detection of all 708 images as to whether the clip is present or missing. The largest absolute error in 708 images is 4.4E-1 and the average absolute error in 708 images is less than 3.8E-2.

Table 4.3: Sugeno parameters

Rule i	a_{i0}	a_{i1}	a_{i2}
Rule 1	1	1.92E-13	7.38E-15
Rule 2	1	3.07E-08	1.30E-1
Rule 3	-1.06E-15	3.56	.81E-09

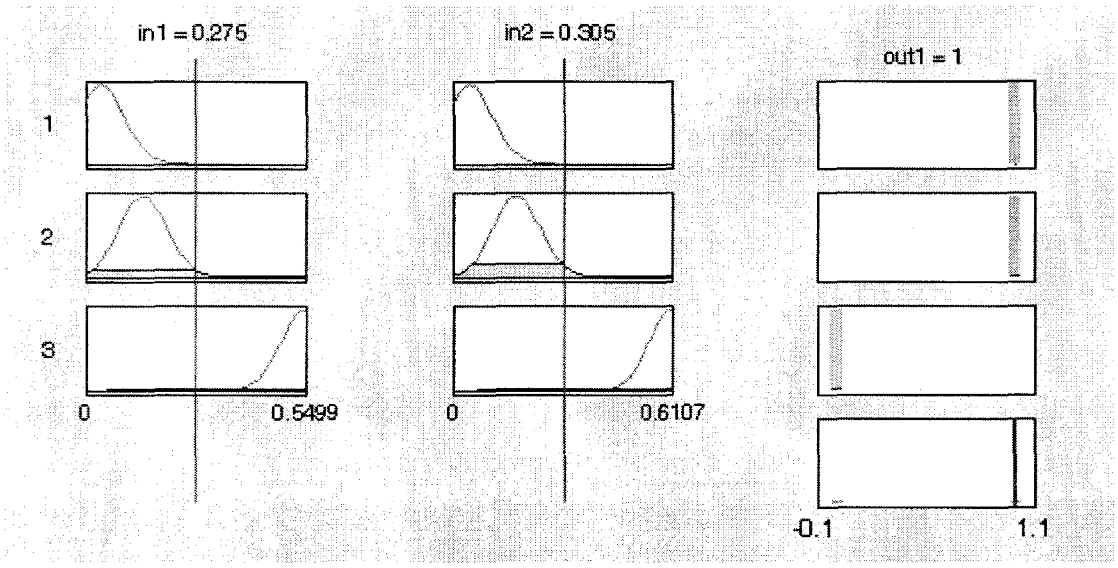


Figure 4.6: The three rules of fuzzy model

4.4.2. Fuzzy Clip Orientation Detection Model

This section is developed to answer the second question. "Is the present clip in the zone of interest installed with proper orientation?" The basic motivation behind the use of the following method is to take care of the uncertainties in the local features of the objects under different orientations in the noisy environment. Our fuzzy model is based on Canny edge detection algorithm. Since Canny algorithm is the most prominent edge detection algorithm (see Section 4.3.2), we have adopted it for the rest of our chapter. However, the use of Canny in our study does not necessarily mean that Canny is the only compatible edge detector algorithm to our fuzzy model but it could be one of the edge detectors

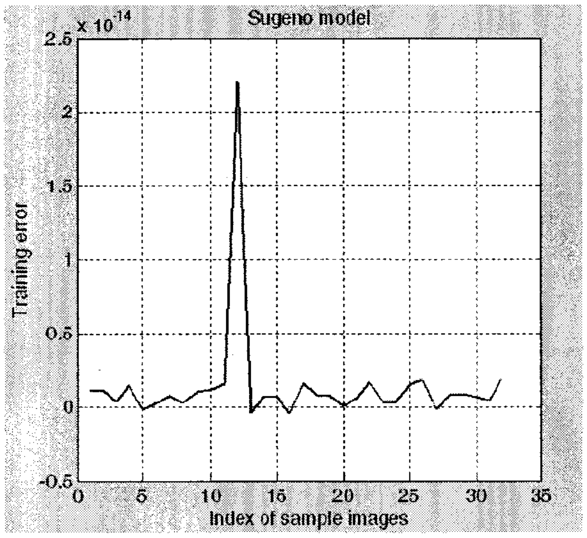


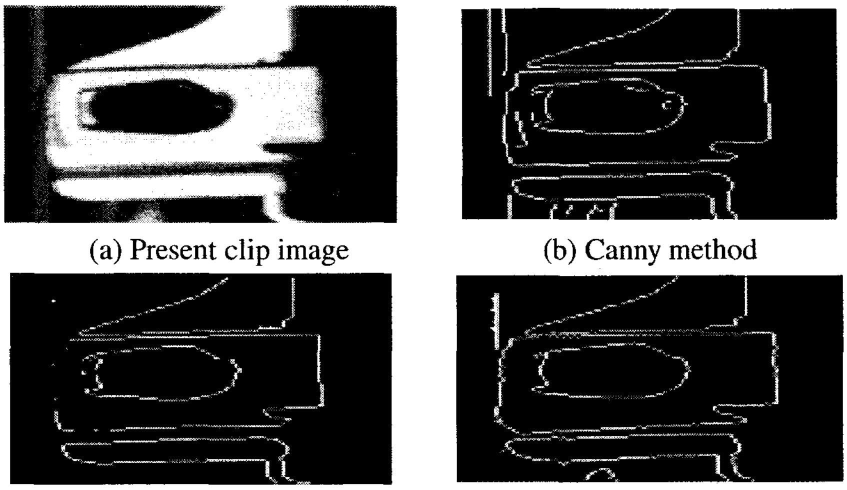
Figure 4.7: Sugeno training error phase for thirty-two images

yielding high performance in our model. This compatibility is considered an advantage to our model. Figure 4.8 shows three different edge detection methods applied to a clip installed with proper orientation. Sobel and threshold based edge detection methods are merely shown to compare the output of different edge detection algorithms.

Let us call Ω_2, Ω_{po} and Ω_{io} as the set of all the sample images, the set of present clip images with proper orientation and the set of present clip images with improper orientation, respectively.

$$\Omega_2 = \Omega_{po} \bigcup \Omega_{io} \tag{4.18}$$

In this model, we divide the surface of the zone of interest into five different regions. Our model can be tuned based on the object of interest. Our regions are determined based on geometric point of view of the application. For instance in this application, we call them as right, left, top, inside and outside rectangles. In short we label them as rectangles R, L, T, I and O, respectively. Figure 4.9 shows their location on the registered image. We take a sample of thirty images with acceptable clip orientation. Using the following



(a) Present clip image

(b) Canny method

(c) Threshold based method

(d) Sobel Method

Figure 4.8: Different edge detection methods

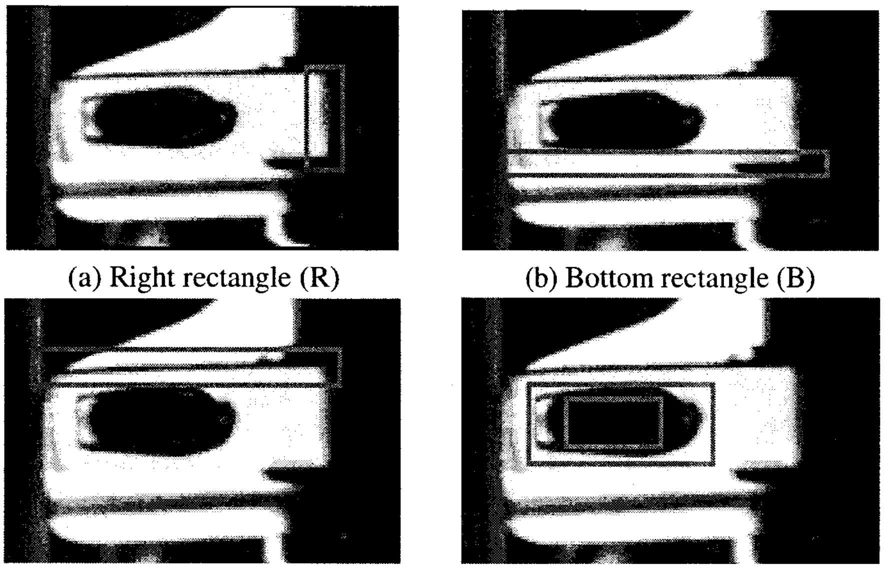
equations, the corner points for rectangle R are determined.

$$R_x^{(i)} = \overline{R^{(i)}}_x - 3 \times R_{\sigma_x}^{(i)} \qquad i = 1,2,3,4 \tag{4.19}$$

$$R_y^{(i)} = \overline{R^{(i)}}_y - 3 \times R_{\sigma_y}^{(i)} \qquad i = 1,2,3,4 \tag{4.20}$$

where, $\overline{R^{(i)}}_x$ and $\overline{R^{(i)}}_y$ are the average of i^{th} corner points of rectangle R in the sample with respect to x and y axes, respectively. Additionally, $R_{\sigma_x}^{(i)}$ and $R_{\sigma_y}^{(i)}$ are the standard deviation of i^{th} corner points of rectangle R in the sample with respect to x and y axes, respectively. In a similar fashion, the position of corner points of rectangles L, T, I and O are determined.

The next step is to count pixel edges which fall in each rectangle as features. Depending on the color in which the edges are traced, their shapes vary, no matter which edge detection method is used. Therefore, another question arises; which color should be the base for edge detection? In Figure 4.10, it can be seen the clip is present but the top



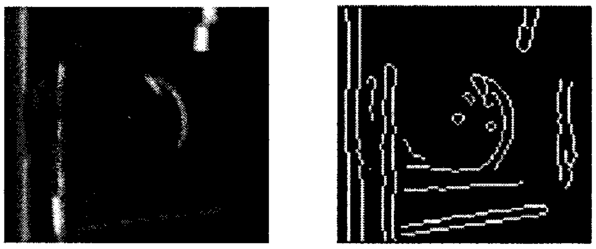
(a) Right rectangle (R)

(b) Bottom rectangle (B)

(c) Top rectangle (T)

(d) Inside/Outside rectangles (I/O)

Figure 4.9: Five different zones



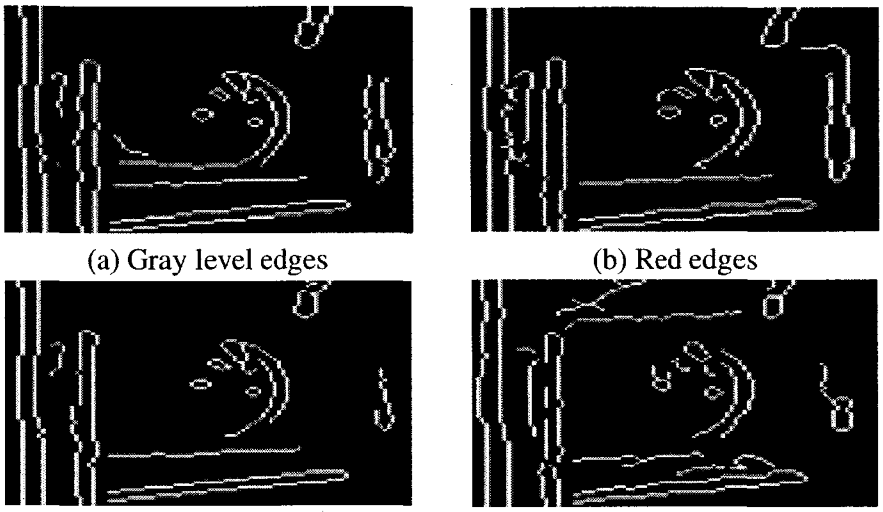
(a) A problematic image

(b) Canny method

Figure 4.10: Poor performance of Canny on a dark clip

edge (top border) is not detected with Canny method in the gray-level. This problem gets more complicated as the clip on the image gets darker .

In Figure 4.11, we illustrate how tracing each color in Canny edge detection results in different edges. In Figure 4.11, it is observed that the top edge which was missing in Figure 4.10 is detected if only blue color is applied for edge detection. So, the objective is finding the color which could be efficient enough to count the edge pixels in each rectangle. As a result, if there is any edge in the position of interest, at least it should be reflected in one of the RGB channels. It should be noted that gray-level could be



(a) Gray level edges

(b) Red edges

(c) Green edges

(d) Blue edges

Figure 4.11: Canny edge detection based on different color tracing

represented as a linear combination of the three RGB colors and is not an independent color. From the stand point of our model, one possible way to count the edge pixels could be applying the maximum function in each region. Therefore, whenever it comes to count the edge pixels in different colors, we get the largest number of pixels as the representing feature.

As the next step, we develop the five-input, one- output Sugeno fuzzy model using subtractive clustering. The features are generated by taking the maximum number of edge pixels in color k at five distinct regions for all the sample images as shown in Equations (4.21) to (4.25).

$$\acute{X}_i^{(1)} = Max_{k\in\{R,G,B\}}\ R_{ik}\ \ i \in \Omega_2 \tag{4.21}$$

$$\acute{X}_i^{(2)} = Max_{k\in\{R,G,B\}}\ B_{ik}\ \ i \in \Omega_2 \tag{4.22}$$

$$\acute{X}_i^{(3)} = Max_{k\in\{R,G,B\}}\ T_{ik}\ \ i \in \Omega_2 \tag{4.23}$$

$$\acute{X}_i^{(4)} = Max_{k\in\{R,G,B\}}\ I_{ik}\ \ i \in \Omega_2 \tag{4.24}$$

Table 4.4: Small portion of training data

Image (NO)	Input 1	Input 2	Input 3	Input 4	Input 5	Output
Properly installed clip 1	304	525	225	27	35	1
Properly installed clip 2	323	450	156	37	45	1
⋮	⋮	⋮	⋮	⋮	⋮	⋮
Improperly installed clip 1	156	125	86	95	51	0
Improperly installed clip 2	23	86	147	86	49	0
⋮	⋮	⋮	⋮	⋮	⋮	⋮

$$\acute{X}_i^{(5)} = Max_{k\in\{R,G,B\}}\ O_{ik}\ \ i \in \Omega_2 \tag{4.25}$$

$$\acute{Y}_i = \begin{cases} 1; & \text{if the clip is installed with proper orientation} \\ 0; & \text{if the clip is installed with improper orientation} \end{cases} \tag{4.26}$$

We define the output of training model for clips with acceptable orientation as one, and the output for clips with unacceptable orientation as zero in Equation (4.26). Therefore, $\acute{Y}_i$ is characterized as a binary value for the output of the model. A small portion of our training data is shown in Table 4.4. After taking a sample of 40 images (20 clips with proper orientation and 20 clips with improper orientation) as training data in subtractive clustering based modeling, we identify a fuzzy model with eight rules. Our subtractive clustering parameters are: ($r_a = 0.4$, $\eta = 1.7$, $\overline{\varepsilon} = 0.5$, $\underline{\varepsilon} = 0.2$). The rules are depicted in Figure 4.12.

The centers of Gaussian input membership functions are given in Table 4.5. The sigma for all the features are chosen as 0.068. $\acute{a}_{ij}$ for $i = 1,2,...,8$ and $j = 0,1,...,5$ are the consequent parameters which are optimized by least square error (LSE). The consequent parameters are given in Table 4.6.

Figure 4.13 shows the absolute error of Sugeno model in training phase. The first 20 images are installed clips with proper orientation and the last 20 images are installed clips with improper orientation. The largest error in training phase is less than 10^{-12}

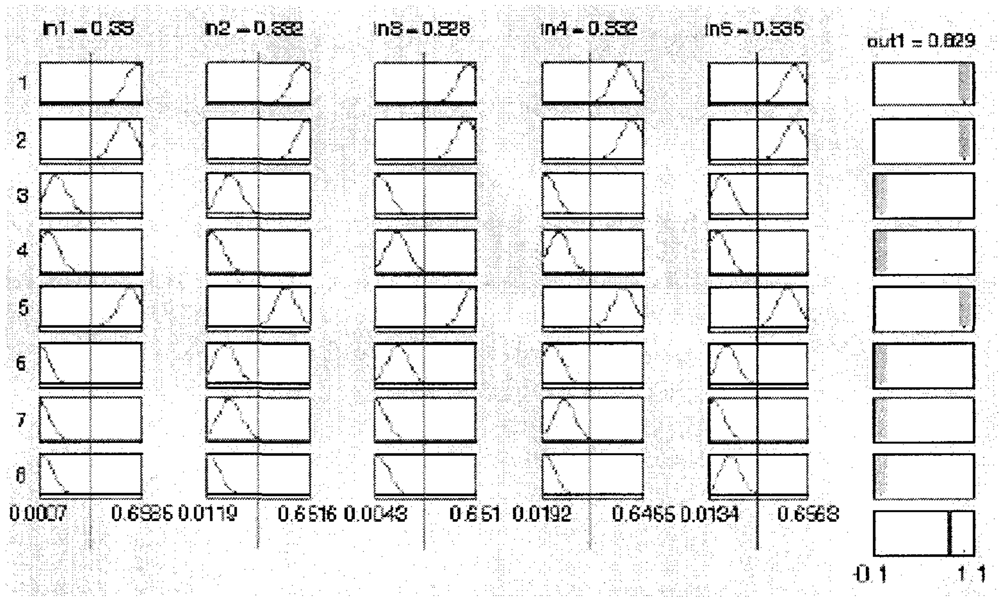


Figure 4.12: The eight rules of fuzzy model

Table 4.5: Centers of Gaussian membership functions (μ)

Cluster (No)	First input (center)	Second input (center)	Third input (center)	Fourth input (center)	Fifth input (center)
Rule 1	0.628	0.608	0.609	0.526	0.572
Rule 2	0.545	0.644	0.593	0.577	0.571
Rule 3	0.118	0.157	0.029	0.040	0.106
Rule 4	0.063	0.045	0.144	0.123	0.074
Rule 5	0.577	0.503	0.634	0.529	0.527
Rule 6	0.004	0.130	0.153	0.082	0.139
Rule 7	0.002	0.159	0.004	0.161	0.049
Rule 8	0.030	0.027	0.024	0.033	0.159

Table 4.6: Sugeno parameters

Rule i	$\acute{a}_{i0}$	$\acute{a}_{i1}$	$\acute{a}_{i2}$	$\acute{a}_{i3}$	$\acute{a}_{i4}$	$\acute{a}_{i5}$
Rule 1	1	3.61E-17	1.06E-11	2.70E-10	3.88E-08	8.69E-14
Rule 2	1	5.72E-13	2.21E-13	1.28E-09	3.39E-10	9.04E-14
Rule 3	-1.64E-16	0.38	.90	0.18	0.33	0.64
Rule 4	-2.92E-17	0.83	.48	0.99	0.97	0.89
Rule 5	1	1.68E-14	1.97E-07	2.23E-11	2.07E-08	1.10E-11
Rule 6	7.06E-17	0.96	.99	.99	0.69	0.37
Rule 7	-2.88E-18	0.96	0.88	.09	0.94	0.99
Rule 8	1.19E-16	0.99	0.34	1.16	0.28	0.23

among 40 images. It illustrates that the fuzzy model can easily identify the orientation of all the images in the sample space and therefore demonstrates its strong capability to be trained by earlier used features. The training time for fuzzy orientation detection is about 6 minutes on a Pentium (R) 4 CPU 2.99 GHz, 1.00 GB RAM and XP platform.

By applying our fuzzy model to the whole database, we get 100% correct orientation identification of all 706 images (2 images were missing clip images). The largest absolute error in 706 images is 0.1437 and the average absolute error in 706 images is less than 0.0178.

4.5. Conclusions

Object orientation detection is one of the most challenging task in machine vision. A two-stage fuzzy model is developed in this work. In the first stage, the existence of the clip in the zone of interest is examined by using two features from HSL color space. In the second stage, a flexible machine vision based inspection system has been proposed which uses a fuzzy subtractive clustering to perform clip orientation detection. In this stage, a new method of grouping and counting points (pixels) on pre-selective regions is proposed. Since this method counts the exact number of edge pixels in the image using a number of

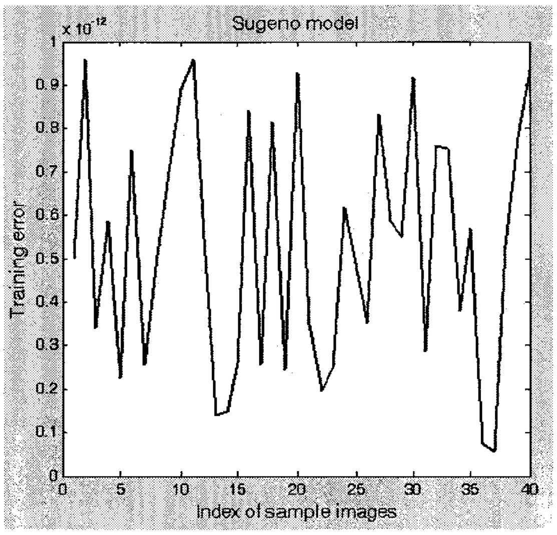


Figure 4.13: Sugeno training error phase for forty images

different region sets, it significantly reduces the probability of false orientation inspection. The method can mostly eliminate the disturbance of artificial lights since we are tracking the orientation of objects in separated RGB channels. Different numbers of pixels in different regions ultimately determine whether the orientation of the clip is acceptable or not, based on the images in the training phase. In addition, this method can reduce the execution time due to the fact that only small portions of pixel edges in the image are examined. The method can be easily tuned and parameterized based on the shape and geometric positions of the test components in the assembly line. Experimental results which have been presented show reliable performance for the application of this method in machine vision inspection. The results demonstrate that the orientation of 100% of the clips in the entire database with a size of 706 images is correctly identified. It is clear that adopting such system will make automated fuzzy machine vision based orientation detection a more feasible alternative in many applications.

Chapter 5

Automated Fuzzy Visual Broken Edge Detection

5.1. Introduction

Inspection in general is the process of determining whether the products deviate from a given set of specifications [Newman and Jain 1995]. A desirable machine vision system is expected to differentiate products into acceptable and non-acceptable categories with the same performance or even better than a human does.

However, the range and scope of machine vision based inspection applications is obviously diverse and widespread. In this paper an attempt is made to detect broken edges using a developed fuzzy machine vision based inspection which is considered one of the promising areas which can significantly benefit the industry.

By definition edges are the boundary between two regions with relatively distinct gray level properties [Gonzalez and Wintz 1977]. An edge indicates discontinuity in image intensity function. They often carry important information about an object, when shown as large gradient magnitude. Geometric shape, boundaries of objects, marks, shadow and shading of a part are presented by edges.

Since most of the information in an image lies on the boundaries between different image regions, the edge based features play an important role in computer vision. These important features frequently are used in automated visual inspection. In machine vision, a feature is a locally detectable pattern of pixels from an image which may represent a piece of higher level information about the image [Yibo 2004]. An accurate extraction of features will lead to the success of classification and recognition of the objects on the image.

Feature recognition can be regarded as an important task that bridges computer aided design (CAD) and computer aided manufacture (CAM). The purpose of feature recognition is to convert the CAD data of a part into a set of features required by downstream manufacturing functions. In certain manufacturing applications such as process planning, inspection, vision system, casting and NC codes generation extensive research has been done [Allada and Anand 1995, Dereli and Filiz 2002, Gao *et al.* 2004, Hu *et al.* 1986, Pal and Kumar 2002]. Common to all the feature recognition techniques reported here is that they have mostly been confined to using edge based models.

This chapter does not intend to propose a new edge detection method. Rather, we develop a step by step method for verifying edges for visual inspection purposes. In spite of the fact that much effort has been made in designing edge operators that are specialized in detecting certain types of edges such as step edge, roof edge, pulse edge, etc., there are very few papers on edge inspection and broken edge detection. This fact indicates the necessity of more work to be done in this field. Although our research is motivated by the requirements in visual water pump surface inspection, the approach proposed here could also be applicable for more general applications.

Due to the fact that fuzzy systems are suitable tools to deal with uncertainty involved in the process of extracting useful information from noisy images, there has been a growing research interest in the applications of fuzzy based methods in recent decades [Byun and Lee 2004, Choi and Krishnapuram 1997, Pham and Prince 1999, Venkata Rao 2006, Yuksel 2006]. Fuzzy approaches has significantly improved the capability of image processing techniques [Chen 1995b, Edinbarough *et al.* 2005, Keller 1997b, Killing *et al.* 2009, Ozbay *et al.* 2006, Park and Kim 2005]. In short, fuzzy models due to their summarization and information compression through the use of granulation capabilities are functional tools not only in quantifying the evidence from partially developed patterns but also combining evidence from different patterns to identify underlying causes [Zadeh 2008]. In this research, we extend our preliminary research on broken edge detection and present a novel fuzzy method for edge inspection in digital images corrupted by tool marks.

The remaining part of this chapter is organized as follows. Section 2 is devoted to the nature of the problem considered for analysis. Conceptual design of the system is briefly described in Section 3. In Section 4, we explain the image pre-processing step. In Section 5, the way the edges (curves) have been processed is described. We explain the proposed features for classification in Section 6. In Section 7, our developed fuzzy classification model and corresponding experimental results have been discussed. Finally, we draw the conclusions in Section 8.

5.2. Broken Edge Detection Problem

In our database, we have 150 water pump images, 100 of which have no significant broken edges while the rest have significant broken edges. We quantify the broken edge as four pixels or more deviation from the reference (master) image. Figure 5.1 shows a generic non broken edge image whereas Figure 5.2 illustrates a typical broken edge image. The white arrows in the Figure 5.2 point to the broken edge areas. Figure 5.3 gives a close-up view of broken edge areas.

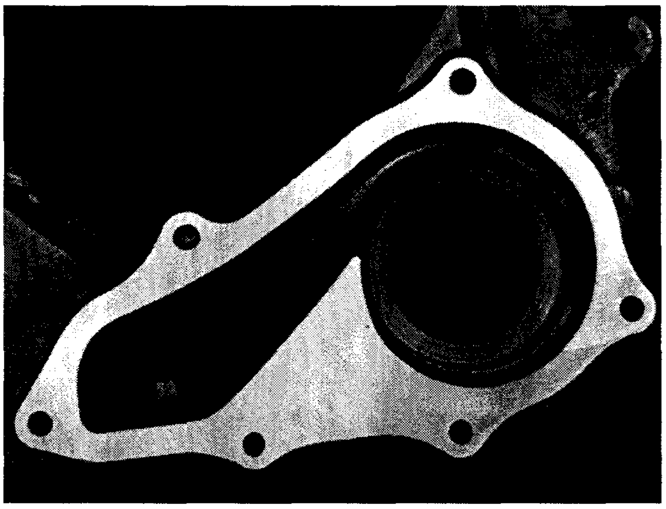


Figure 5.1: Non broken edge water pump (reference image)

It should be noted that the nature of broken edge detection problem is extremely sensitive to noise. This arises a defying situation since we are dealing with rapid changes in light conditions from one image to another and from one area of image to another. In addition to broken edge detection problem, reflection is another specific problem of machined metallic surfaces. This in turn presents variations in the appearance of defect-free parts as well as in the appearance of the parts with broken edges. In this work we approach the broken edge detection problem as a classification problem and use fuzzy theory for decision making.

The main objective of this work is to develop an efficient fuzzy machine vision method (using required pre-processing steps) to correctly identify parts with broken edges. Parts are water pumps in this application.

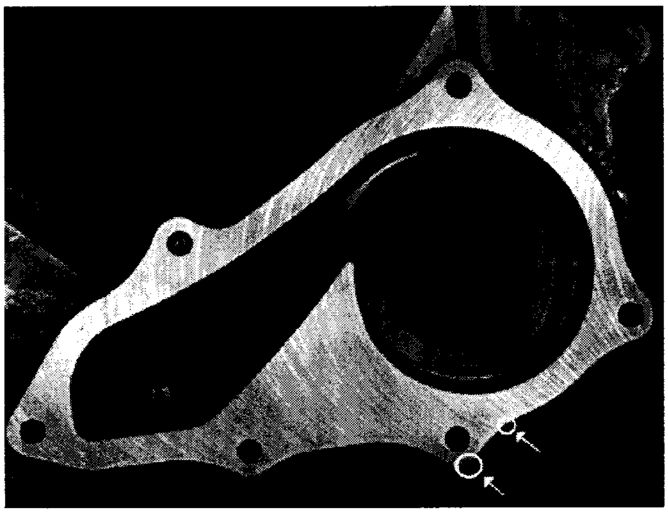


Figure 5.2: Broken edge water pump

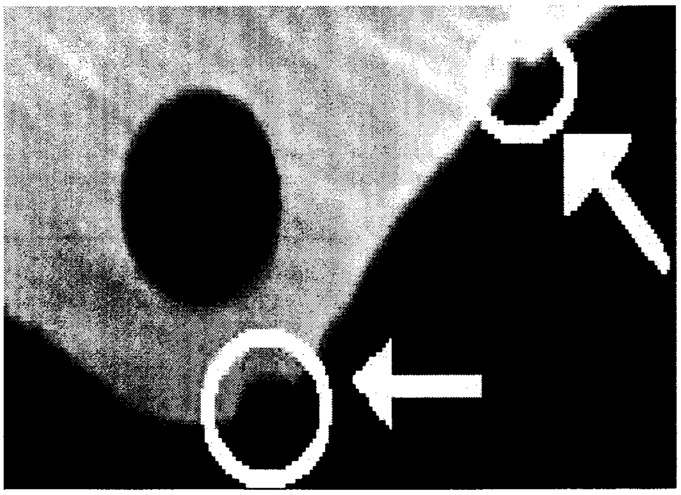


Figure 5.3: Close up broken edge image

5.3. Conceptual Design of the System

In the industry, we usually witness casting problems which result in a phenomenon called broken edge. Broken edges have direct impact on the quality of the work in process and/or final product. It is very natural that in many processes broken edge parts are considered defective. These broken edges could occur in many products with any material, ranging from cast iron water pumps to fancy crystal products.

Nevertheless, manufacturing totally defect-free parts is almost impossible, due to labor fatigue, limited engineering abilities, or more importantly economical concerns. Therefore, the essential point is the inspection to prevent broken edge parts move to the next station or reach the market. Consequently, designing an efficient machine vision based broken edge detection model has a key role to manage such kind of operations. Figure 7.5 shows the conceptual design of the system. It is composed of eight components described as follows.

- The first component is a fixed image acquisition system (camera on a mobile platform) supplying the capture of the surface of the water pump images in this application. All the initial images have been captured in standard Tagged Image File Format (Tiff). They are all gray level images with a size of 960 by 1280.

- A digital reference image, also called master image, that has been captured and processed at the highest practicable quality acts as a guide for image pre-processing and later as an edge or curve template for broken edge feature development.

- Image pre-processing phase is the next component. It is necessary to align and/or integrate the water pump images obtained from different parts at different times from slightly different views (mobile platform).

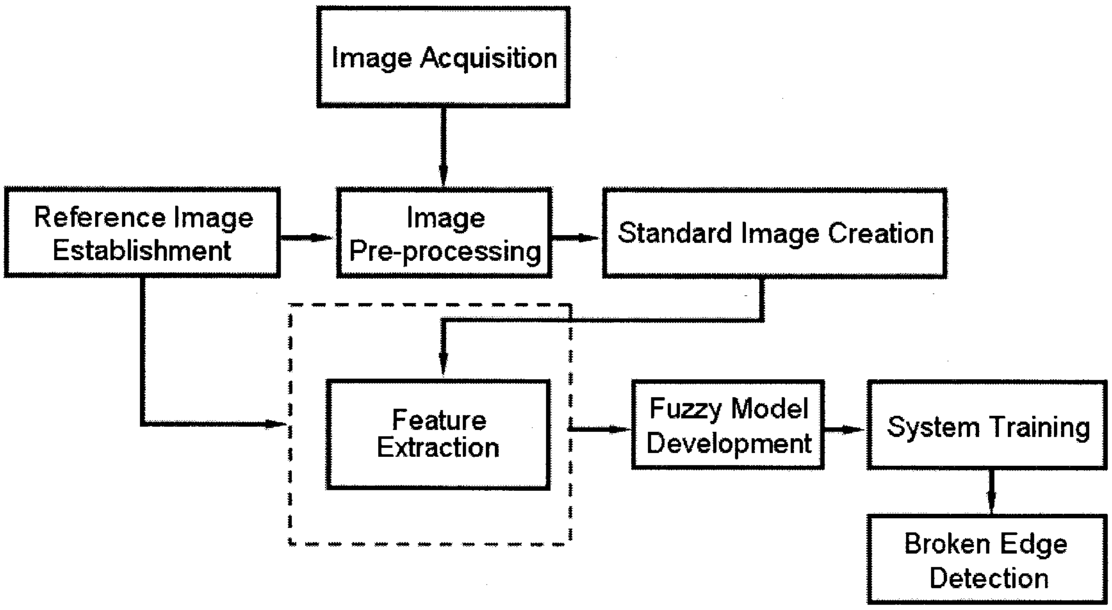


Figure 5.4: Scheme of the system

- Since there is a need to develop a standard image to be comparable with reference image, a component is devoted to create the standard image.

- In the next component, it is time to construct the features. Since feature extraction is a special form of dimensionality reduction in order to distinguish broken edge and non broken area, we propose a set of broken edge detection features to highlight the distinctive characteristics of these areas. As an attempt to maintain the generality of this research the constructed features are also application-independent.

- A fuzzy model is developed to classify the features and label the water pump images either as acceptable or as rejected parts. It can assign degrees of membership to the both of the above mentioned classes.

- The fuzzy membership and consequent parameters should be tuned (optimized) in the system training component to make a final determination about the fuzzy model.

- Broken edge detection is the last component. The decision making process takes place in this step. Detection of broken edges is based on the contents of feature vectors and fuzzy rules.

5.4. Image pre-processing

In this section, we introduce the steps that should be taken in order to reach to the pre-processed image. First, we create a binary image, then we clear all the noises and redundant objects in order to detect the main object (water pump surface). Afterward, the test image is adjusted horizontally and vertically, based on the reference image, and its location is set (displacement and orientation adjustment). Finally, the zone of interest for each image is automatically determined. These steps are explained in details below.

5.4.1. Creating A Binary Image

The first step in this process is moving from a gray level space to a binary space. However, a large amount of care should be taken with respect to determination of the threshold. The threshold should be adjusted from case to case. It means that each single image should have a certain threshold. Otherwise, noise would have a destructive impact on some images. Since the water-pump images were acquired on a dark background, thresholding-based segmentation technique was applied to distinguish the region of water-pump from the background with an adaptive threshold. Being a computationally efficient method, Otsu's method [Otsu 1979] automatically obtains the optimal threshold that minimizes the intra-class variance between dark and light regions for each image individually. To eliminate the gaps within image, three morphological operators were applied to the binary image obtained by the thresholding-based segmentation method. An erosion operation was applied on the binary image to remove the noises. After that a dilation operation was implemented to fill the gaps within the water-pump. To remove extra layer created by the dilation operation, another erosion operation was implemented.

Let $X_{i,j,n}$ the gray level value of the pixel in the i^{th} row, the j^{th} column and in the n^{th} image. where, $i = 1,...,960$, $j = 1,...,1280$ and $n = 1,2,...,105$. The corresponding tensor for $X_{i,j,n}$ is $\mathbf{X}_n$. It should be noted that the origin in pixel coordinates is located at the top left of the image while the origin in spatial coordinates is located at the bottom left. The relationship between pixel and spatial coordinate systems is simply defined in Equations (5.1) and (5.2).

$$x = j - 1 \tag{5.1}$$

$$y = 1280 - i \tag{5.2}$$

where, pixel coordinates and spatial coordinates, are denoted by (i, j) and (x,y), respectively. $B^{(1)}_{i,j,n}$ is defined as the binary value of the pixel in the i^{th} row and the j^{th} column after implementing thresholding-based segmentation method on the n^{th} image ($\mathbf{X}_n$). The corresponding tensor for $B^{(1)}_{i,j,n}$ is $\mathbf{B}^{(1)}_n$. Among all the images in our database we choose the image shown in Figure 5.1(a) as the reference image since it doesn't have any broken-edge and has also a smooth surface. We define $G_{i,j}$ as the pixel value in the i^{th} row and the j^{th} column of the gray level reference image. $\mathbf{G}$ is the corresponding matrix for $G_{i,j}$. In a similar fashion, $R^{(1)}_{i,j}$ represents the binary value of pixel in the i^{th} row and the j^{th} column after implementing thresholding-based segmentation method on the reference image ($\mathbf{G}$). $\mathbf{R}^{(1)}$ is the corresponding matrix for $R^{(1)}_{i,j}$. Figure 5.5 shows the binary output of Figure 5.1(a) through this process.

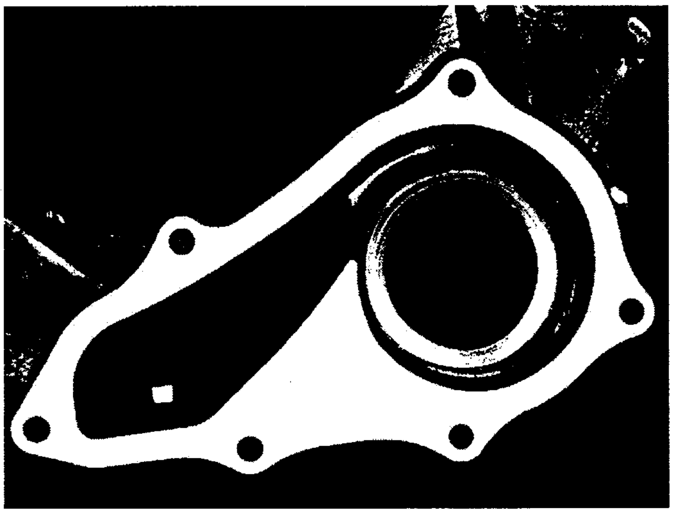


Figure 5.5: Binary output of reference image Figure 5.1(a)

5.4.2. Object Detection

After creation of the binary image, it is time to clear the noise from the image and remove other redundant objects. Therefore, we need to take all the irrelevant objects out of the image. Basically, this step consists of two sub-processes. In the first sub-process, we detect all the objects. In the second one, the largest object will be extracted as the main object (the water-pump) and the rest is eliminated. To do so, we find all the objects in each image with a recursive procedure. At first, a white pixel is randomly marked and its neighbors are determined. From its found (white) neighbors, we explore the rest of pixels that belong to the object which the first pixel belongs to. This algorithm repeats until there is no white pixel belonging to the current object. In the next step, the algorithm tries to find the second object (if any). The algorithm is terminated if no new object is found and all the pixels are examined. When no new object has been found, it measures the surface (counting the pixels) of each object and determines the largest object. In this case, the largest object which is desirable for us is the body of water-pump. After applying the noise removal process of Figure 5.5, we obtain Figure 5.6 where it presents an image without noise and redundant objects. The corresponding tensor of the n^{th} image is $\mathbf{B_n^{(2)}}$, where $B_{i,j,n}^{(2)}$ represents the binary value of pixel in the i^{th} row and the j^{th} column in the n^{th} cleared image. The corresponding matrix of reference image is $\mathbf{R}^{(2)}$, where $R_{i,j}^{(2)}$ represents the binary value of pixel in the i^{th} row and the j^{th} column in the cleared reference image.

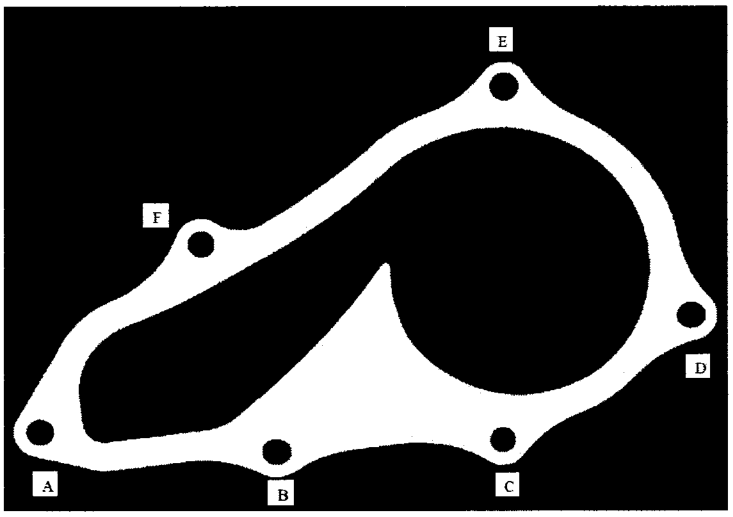


Figure 5.6: Extraction of the main object in the reference image

5.4.3. Horizontal, Vertical and Orientation Adjustments

The water-pump images have different orientations and there are some displacements with respect to x axis and y axis. It is important to note that there is no significant orientation towards z axis. Therefore, finding two reliable points is sufficient to pre-process all the images. In this chapter, we use the centers of two circles as the reliable points to perform horizontal, vertical and orientation adjustments. There are six small circles around the water-pump (see Figure 5.6). Starting with the left most circle, we call them A,B,C,D,E

and F in the counter clockwise direction. Therefore, we need to find the two most reliable centers among these six centers. To do so, the algorithm firstly needs to identify these six objects (circles) and calculate their corresponding centers. Analogous to what we did in Section 5.4.2, it is better to switch the background and foreground by using the following equations to identify these circles. This conversion is merely taken place to be consistent with the typical definition of object as the connected of white pixels in a black environment in the machine vision literature and has nothing to do with the logic of pre-processing step.

Hence, we define $\mathbf{B_n^{(3)}}$ as the inverse cleared image of $\mathbf{B_n^{(2)}}$ and $\mathbf{R}^{(3)}$ as the inverse cleared image of $\mathbf{R}^{(2)}$. $B_{i,j,n}^{(3)}$ represents the binary value of pixel in the i^{th} row and the j^{th} column in the n^{th} inverse cleared image and $R_{i,j}^{(3)}$ represents the binary value of pixel in the i^{th} row and the j^{th} column in the inverse cleared reference image. Therefore, $\mathbf{B_n^{(3)}}$ is the corresponding tensor of $B_{i,j,n}^{(3)}$ and $\mathbf{R}^{(3)}$ is the corresponding matrix of $R_{i,j}^{(3)}$. $B_{i,j,n}^{(3)}$ and $R_{i,j}^{(3)}$ are defined in Equations (5.3) and (5.4).

$$B_{i,j,n}^{(3)} = \begin{cases} 1; \text{ if } B_{i,j,n}^{(2)} = 0 \\ \\ 0; \text{ if } B_{i,j,n}^{(2)} = 1 \end{cases} \tag{5.3}$$

$$R_{i,j}^{(3)} = \begin{cases} 1; \text{ if } R_{i,j}^{(2)} = 0 \\ \\ 0; \text{ if } R_{i,j}^{(2)} = 1 \end{cases} \tag{5.4}$$

Figure 5.7 shows the $\mathbf{R}^{(3)}$. In Figure 5.7, excluding the exterior object, we have seven objects, six circles and one large object in the middle of image. Given the $\mathbf{B_n^{(3)}}$ as the n^{th} inverse cleared image, using the described methodology in Section 5.4.2, seven sets of objects are extracted $(O_{k,n}, k = 1,2,...,7)$, where $O_{k,n}$ is the k^{th} found object in the n^{th} inverse cleared image. In a similar fashion, $O_{k,n}^{(2)}$ is the corresponding set to $\mathbf{R_n^{(3)}}$, where

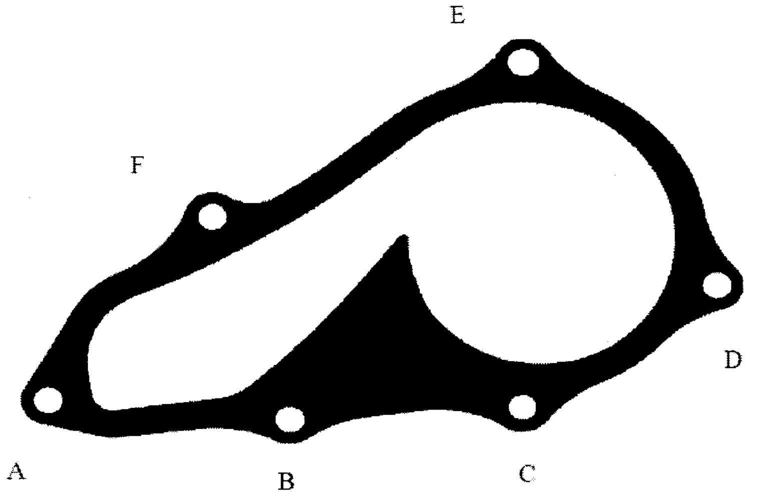


Figure 5.7: Inverse cleared reference image

$O^{(2)}_{k,n}$ is the k^{th} found object in the inverse cleared reference image. It is important to note that because of displacement and rotation of objects, $O_{k,n}$ is not necessarily corresponding to $O^{(2)}_{k,n}$.

At this point it is clear that the set with the most number of members belongs to the largest object in the n^{th} image. The largest object in the n^{th} image is represented by $\acute{O}_n$. In a similar fashion, the largest object in the reference image is represented by $\acute{O}^{(2)}$. For the rest six objects, we characterize the k^{th} object in the n^{th} image by $\hat{O}_{k,n}$, where $(k = 1,2,...,6)$. Similarly, the k^{th} object in the reference image is also characterized by $\hat{O}^{(2)}_k$, where $(k = 1,2,....6)$.

However, by now, the set of objects corresponding to the circles are unknown. It means that we still do not know which value of k refers to which circle in Figure 5.7. Therefore, center of circles should be calculated. Using Equations (5.5) and (5.6), each set of pixels (objects) is converted to a new binary image.

$$B^{(4)}_{i,j,k,n} = \begin{cases} 1; \text{ if } (i,j)\ \varepsilon\ \hat{O}_{k,n}\ \forall i,j,k,n \\ \\ 0; \text{ otherwise} \end{cases} \tag{5.5}$$

where, $B^{(4)}_{i,j,k,n}$ represents the pixel value of i^{th} row and j^{th} column of the k^{th} object in the n^{th} cleared inverse image. The corresponding tensor is called $\mathbf{B^{(4)}_{k,n}}$.

$$R^{(4)}_{i,j,k} = \begin{cases} 1; \text{ if } (i,j)\ \varepsilon\ \hat{O}^{(2)}_k\ \forall i,j,k \\ \\ 0; \text{ otherwise} \end{cases} \tag{5.6}$$

where, $R^{(4)}_{i,j,k}$ represents the pixel value of the i^{th} row and the j^{th} column of the k^{th} object in the cleared inverse reference image. The corresponding matrix is $\mathbf{R^{(4)}_k}$. The corresponding images to the $\acute{O}_n$ and $\acute{O}^{(2)}$ are defined in Equations (5.7) and (5.8).

$$B^{(5)}_{i,j,n} = \begin{cases} 1; \text{ if } (i,j)\ \varepsilon\ \acute{O}_n\ \ \forall i,j,n \\ \\ 0; \text{ otherwise} \end{cases} \tag{5.7}$$

where, $B^{(5)}_{i,j,n}$ represents the pixel value of the i^{th} row and the j^{th} column of the largest object in the n^{th} cleared inverse image. The corresponding tensor is called $\mathbf{B^{(5)}_n}$.

$$R^{(5)}_{i,j} = \begin{cases} 1; \text{ if } (i,j)\ \varepsilon\ \acute{O}^{(2)}\ \ \forall i,j \\ \\ 0; \text{ otherwise} \end{cases} \tag{5.8}$$

where, $R^{(5)}_{i,j}$ represents the pixel value of the i^{th} row and j^{th} column of the largest object in the cleared inverse reference image. The corresponding matrix is called $\mathbf{R}^{(5)}$.

The concept of the center of mass is described as an average of the masses factored by their distances from a reference point in one plane. The reference points in this case are the centers of circles. Therefore, the center of mass of each object in the n^{th} image with respect to x axis and y axis, is determined in Equations (5.9) and (5.10), respectively. The center of mass of each object in the reference image with respect to x axis and y axis, is also determined in Equations (5.11) and (5.12), respectively.

$$C_{X_{k,n}} = \frac{\sum_{j=1}^{n_2} B^{(4)}_{i,j,k,n} \times j}{\sum_{i=1}^{n_1} \sum_{j=1}^{n_2} B^{(4)}_{i,j,k,n}} \; \forall k,n \tag{5.9}$$

$$C_{Y_{k,n}} = \frac{\sum_{i=1}^{n_1} B^{(4)}_{i,j,k,n} \times i}{\sum_{i=1}^{n_1} \sum_{j=1}^{n_2} B^{(4)}_{i,j,k,n}} \; \forall k,n \tag{5.10}$$

where, $(C_{X_{k,n}}, C_{Y_{k,n}})$ show the center of mass of the k^{th} object (circle) in the n^{th} image. Center of circles A,B,C,D,E and F in n^{th} image are shown by $(C_{X_{A_n}}, C_{Y_{A_n}})$, $(C_{X_{B_n}}, C_{Y_{B_n}})$,$(C_{X_{C_n}}, C_{Y_{C_n}})$, $(C_{X_{D_n}}, C_{Y_{D_n}})$, $(C_{X_{E_n}}, C_{Y_{E_n}})$ and $(C_{X_{F_n}}, C_{Y_{F_n}})$, respectively.

$$C^{(2)}_{X_k} = \frac{\sum_{j=1}^{n_2} R^{(4)}_{i,j,k} \times j}{\sum_{i=1}^{n_1} \sum_{j=1}^{n_2} R^{(4)}_{i,j,k}} \; \forall k \tag{5.11}$$

$$C^{(2)}_{Y_k} = \frac{\sum_{i=1}^{n_1} R^{(4)}_{i,j,k} \times i}{\sum_{i=1}^{n_1} \sum_{j=1}^{n_2} R^{(4)}_{i,j,k}} \; \forall k \tag{5.12}$$

where, $(C^{(2)}_{X_k}, C^{(2)}_{Y_k})$ show the center of circles in the reference image. Center of circles A,B,C,D,E and F in the reference image are shown by $(C^{(2)}_{X_A}, C^{(2)}_{Y_A})$, $(C^{(2)}_{X_B}, C^{(2)}_{Y_B})$, $(C^{(2)}_{X_C}, C^{(2)}_{Y_C})$, $(C^{(2)}_{X_D}, C^{(2)}_{Y_D})$, $(C^{(2)}_{X_E}, C^{(2)}_{Y_E})$ and $(C^{(2)}_{X_F}, C^{(2)}_{Y_F})$, respectively.

Table 5.1 shows the centers of circles in the reference image. The name of circle in reference image (first column), the name of circle center with respect to x axis (second column), the corresponding k index to the name of circle center with respect to x axis (third column), circle center value with respect to x axis (fourth column), the name of

Table 5.1: Centers of reference image

circle	$C_{X_i}^{(2)}$	$C_{X_k}^{(2)}$	x center	$C_{Y_i}^{(2)}$	$C_{Y_k}^{(2)}$	y center
A	$C_{X_A}^{(2)}$	$C_{X_2}^{(2)}$	64	$C_{Y_A}^{(2)}$	$C_{Y_2}^{(2)}$	150
B	$C_{X_B}^{(2)}$	$C_{X_3}^{(2)}$	477	$C_{Y_B}^{(2)}$	$C_{Y_3}^{(2)}$	112
C	$C_{X_C}^{(2)}$	$C_{X_5}^{(2)}$	877	$C_{Y_C}^{(2)}$	$C_{Y_5}^{(2)}$	135
D	$C_{X_D}^{(2)}$	$C_{X_6}^{(2)}$	1213	$C_{Y_D}^{(2)}$	$C_{Y_6}^{(2)}$	372
E	$C_{X_E}^{(2)}$	$C_{X_4}^{(2)}$	880	$C_{Y_E}^{(2)}$	$C_{Y_4}^{(2)}$	807
F	$C_{X_F}^{(2)}$	$C_{X_1}^{(2)}$	339	$C_{Y_F}^{(2)}$	$C_{Y_1}^{(2)}$	589

circle center with respect to y axis (fifth column), the corresponding k index to the name of circle center with respect to y axis (sixth column) and circle center value with respect to y axis (seventh column) are shown in Table 5.1.

Using Table 5.1, it is obvious that $C_{X_A}^{(2)} < C_{X_F}^{(2)} < C_{X_B}^{(2)} < C_{X_C}^{(2)} < C_{X_E}^{(2)} < C_{X_D}^{(2)}$. Therefore, the generated results from Equations (5.11) and (5.12) could be assigned as corresponding circle centers in the reference image.

Therefore, depending on the value of k, whenever we use $(C_{X_k}^{(2)}, C_{Y_k}^{(2)})$, we know the center of which object (circle) in reference image we are referring. However, when it comes to test images, the mentioned information is unknown. In other words, we do not know to which circle in corresponding test images, the index of k in $(C_{X_{k,n}}, C_{Y_{k,n}})$ refers to. Consequently, in order to automatically assign centers of circles in test images to the reference images, the Euclidean distance between centers of circle k and circle $\acute{k}$ in n^{th} test image is calculated in Equation (5.13).

$$D_{k,\acute{k},n} = \sqrt{(C_{X_{k,n}} - C_{X_{\acute{k},n}})^2 + (C_{Y_{k,n}} - C_{Y_{\acute{k},n}})^2} \tag{5.13}$$

The Euclidean distance between centers of circle k and circle $\acute{k}$ in reference image is also calculated in Equation (5.14).

$$D^{(2)}_{k,\acute{k}} = \sqrt{(C^{(2)}_{X_k} - C^{(2)}_{X_{\acute{k}}})^2 + (C^{(2)}_{Y_k} - C^{(2)}_{Y_{\acute{k}}})^2} \tag{5.14}$$

We have 6 values for k and 6 circles in the test images. We can assign the value of k ($k = 1,2,..,6$) to these six circles (A,B,C,D,E and F) with 6! different possibilities. $k_1, k_2, ..., k_6$ represent a set of assignments to circles A,B,C,D,E and F, respectively, which means that ${k_1}^{th}$ is corresponding to circle A, ${k_2}^{th}$ is corresponding to circle B and so on.

We define the $A_{k_1,k_2,k_3,...,k_6,n}$, the absolute difference between the line segments connecting the centers of circles (A,B),(B,C),(C,D),(D,E),(E,F) and (F,A) in the n^{th} test image with its corresponding line segments in reference image, as shown in Equation (5.15), in order to know which circle (object) in the n^{th} image is corresponding to which circle (object) in the reference image. $\overline{AB}$, $\overline{BC}$, $\overline{CD}$, $\overline{DE}$, $\overline{EF}$ and $\overline{FA}$ are the line segments for the reference image.

$$A_{k_1,k_2,k_3,...,k_6,n} = |D_{k_1,k_2,n} - \overline{AB}| + |D_{k_2,k_3,n} - \overline{BC}|$$

$$+ |D_{k_3,k_4,n} - \overline{CD}| + |D_{k_4,k_5,n} - \overline{DE}|$$

$$+ |D_{k_5,k_6,n} - \overline{EF}| + |D_{k_6,k_1,n} - \overline{FA}| \tag{5.15}$$

If the following condition is satisfied, we assign k_1 to circle A, k_2 to circle B, k_3 to circle C, k_4 to circle D, k_5 to circle E and k_6 to circle E.

$$A_{k_1,k_2,k_3,...,k_6,n} = Min_{k_1,k_2,k_3,...,k_6}\{A_{k_1,k_2,k_3,...,k_6,n}\} \tag{5.16}$$

In other words, the assignment with minimum absolute distance difference with the number resulting from the above formula, is the right assignment.

Now that we have found six circles (objects) in the n^{th} inverse cleared image with their corresponding circles (objects) in the reference image, we need to determine which of the two circles are the most reliable ones for pre-processing. Therefore, we define circle circularity factor for the n^{th} test image as a ratio of 4π times of area to the square of perimeter. Consequently, the circularity factor of a perfect circle is equal to one.

Circularity factor of a circle with radius $R = \frac{4\pi(\pi R^2)}{(2\pi R)^2} = \frac{4\pi^2 R^2}{4\pi^2 R^2} = 1$. In a binary image, edge can be defined as the place where white and black pixels meet. We define edge pixel $E_{i,j,k,n} = 1$ in the i^{th} row and the j^{th} column for the k^{th} object in the n^{th} image in Equation (5.17).

$$E_{i,j,k,n} = \begin{cases} 1; & \text{if} \quad \text{pixel } (i,j) \text{ is an edge pixel on object } k \\ \\ 0; & \text{otherwise} \end{cases} \tag{5.17}$$

A pixel could not be considered on the edge of an object if its eight neighbors are all black or all white. Hence, we obtain Equation (5.18).

$$E_{i,j,k,n} = \begin{cases} 1; & \text{if} \quad 0 < \sum_{k_1=i-1}^{i+1} \sum_{k_2=j-1}^{j+1} B^{(4)}_{k_1,k_2,k,n} < 8 \\ \\ 0; & \text{otherwise} \end{cases} \tag{5.18}$$

The circularity factor of the k^{th} object in the n^{th} image is determined using the Equation (5.19).

$$f_{k,n} = \frac{4\pi \sum_{i=1}^{n_1} \sum_{j=1}^{n_2} B^{(4)}_{i,j,k,n}}{(\sum_{i=1}^{n_1} \sum_{j=1}^{n_2} E_{i,j,k,n})^2} \tag{5.19}$$

The next step is to determine the average circularity factor of the k^{th} object in the whole database using the Equation (5.20).

$$C_k = \sum_{n=1}^{N} |f_{k,n} - 1| \tag{5.20}$$

Table 5.2: Average circularity factor

circle	k	C_k	chosen Object	$C^{(2)}_{X_k}$	x reference center	$C^{(2)}_{Y_k}$	y reference center
A	2	1.121	-	-	-	-	-
B	3	1.082	-	-	-	-	-
C	5	1.053	$*$	$C^{(2)}_{X_5}$	877	$C^{(2)}_{Y_5}$	135
D	6	1.035	$*$	$C^{(2)}_{X_6}$	1213	$C^{(2)}_{Y_6}$	372
E	4	1.083	-	-	-	-	-
F	1	1.160	-	-	-	-	-

Table 5.2 shows the average circularity factor of the k^{th} object. The name of circle in reference image (first column), the corresponding k index to the name of circle center (second column), the average circularity factor of k^{th} circle (third column), chosen circles based on the smallest circularity factor (fourth column), the corresponding k index to the name of chosen circle center with respect to x axis (fifth column), the chosen circle center value with respect to x axis (sixth column), the corresponding k index to the name of chosen circle center with respect to y axis (seventh column) and the chosen circle center value with respect to y axis (eighth column) are shown in Table 5.2. We choose the k indices of the two smallest C_k as the most reliable circles which are circle C and circle D.

Now, the next step is to determine the ϕ_n which is the angle between the x axis and the line which connects the centers of circles C and D in the n^{th} test image. In a similar way, the same angle in the reference image is represented by θ. ϕ_n and θ are calculated using Equations (5.21) and (5.22).

$$\phi_n = ArcTan(\frac{C_{Y_{D_n}} - C_{Y_{C_n}}}{C_{X_{D_n}} - C_{X_{C_n}}}) \tag{5.21}$$

$$\theta = ArcTan(\frac{C^{(2)}_{Y_D} - C^{(2)}_{Y_C}}{C^{(2)}_{X_D} - C^{(2)}_{X_C}}) \tag{5.22}$$

Consequently, $\Delta\phi_n$ represents the difference angle between reference image and the n^{th} test image which is called rotation angle, as shown in Equation (5.23).

$$\Delta\phi_n = \phi_n - \theta \tag{5.23}$$

In terms of displacement adjustment, we calculate the d_{x_n} and d_{y_n} which measure the average circle center positions from $x = 0$ and $y = 0$ in the n^{th} image, respectively. In a similar fashion, we represent these values in the reference image for x and y displacement as $d_x^{(2)}$ and $d_y^{(2)}$.

Equations (5.24), (5.25), (5.26) and (5.27) calculate the values of d_{x_n}, d_{y_n}, $d_x^{(2)}$ and $d_y^{(2)}$, respectively.

$$d_{x_n} = \frac{C_{X_{C_n}} + C_{X_{D_n}}}{2} \tag{5.24}$$

$$d_{y_n} = \frac{C_{Y_{C_n}} + C_{Y_{D_n}}}{2} \tag{5.25}$$

$$d_x^{(2)} = \frac{C_{X_C}^{(2)} + C_{X_D}^{(2)}}{2} \tag{5.26}$$

$$d_y^{(2)} = \frac{C_{Y_C}^{(2)} + C_{Y_D}^{(2)}}{2} \tag{5.27}$$

Consequently, Δx_n and Δy_n represent the difference distances between the reference image and the n^{th} test image which are called the shifting distance for x and y axes, as shown in Equations (5.28) and (5.29), respectively.

$$\Delta x_n = d_{x_n} - d_x^{(2)} \tag{5.28}$$

$$\Delta y_n = d_{y_n} - d_y^{(2)} \tag{5.29}$$

Using the rotation angle and shifting distance, appropriate transform is calculated. The values of i_n and j_n are the two new indices that store the transformed object position for x and y axes in the n^{th} transformed object, respectively. $\acute{i}_n$ and $\acute{j}_n$ are the corresponding pixel coordinates of i_n and j_n. Δx_n and Δy_n are the shifted coordinated system and ϕ_n is the rotation angle for the n^{th} test image. The relationship between these variables is shown in Equations (5.30), (5.31) and (5.32).

$$\begin{bmatrix} i_n \\ j_n \end{bmatrix} = \begin{bmatrix} \cos(\overline{\phi}_n) & -\sin(\overline{\phi}_n) \\ \sin(\overline{\phi}_n) & \cos(\overline{\phi}_n) \end{bmatrix} \begin{bmatrix} j-1 \\ 1280-i \end{bmatrix} + \begin{bmatrix} \Delta x_n \\ \Delta y_n \end{bmatrix} \tag{5.30}$$

$$\acute{i}_n = 1280 - j_n \tag{5.31}$$

$$\acute{j}_n = i_n + 1 \tag{5.32}$$

In order to extract the water-pump images, the proper mask is calculated using Equation (5.37). $B^{(6)}_{\acute{i}_n,\acute{j}_n,n}$ shows the transform of binary reference image to the position of the n^{th} test image. The corresponding tensor is shown by $\mathbf{B}_\mathbf{n}^{(\mathbf{6})}$.

$$B^{(6)}_{\acute{i}_n,\acute{j}_n,n} = \begin{cases} 1; & \text{if } R^{(2)}_{i,j} = 1 \\ 0; & \text{otherwise} \end{cases} \tag{5.33}$$

$B^{(7)}_{\acute{i}_n,\acute{j}_n,n}$ shows the corresponding gray value of $B^{(6)}_{\acute{i}_n,\acute{j}_n,n}$. The corresponding tensor is shown by $\mathbf{B}_\mathbf{n}^{(\mathbf{7})}$. $B^{(7)}_{\acute{i}_n,\acute{j}_n,n}$ is calculated using Equation (5.34). Finally, Figure 6.3 shows the pre-processed image of Figure 5.1(a).

$$B^{(7)}_{i'_n,j'_n,n} = \begin{cases} B^{(1)}_{i,j,n}; & \text{if } R^{(2)}_{i,j} = 1 \\ 0; & \text{otherwise} \end{cases} \tag{5.34}$$

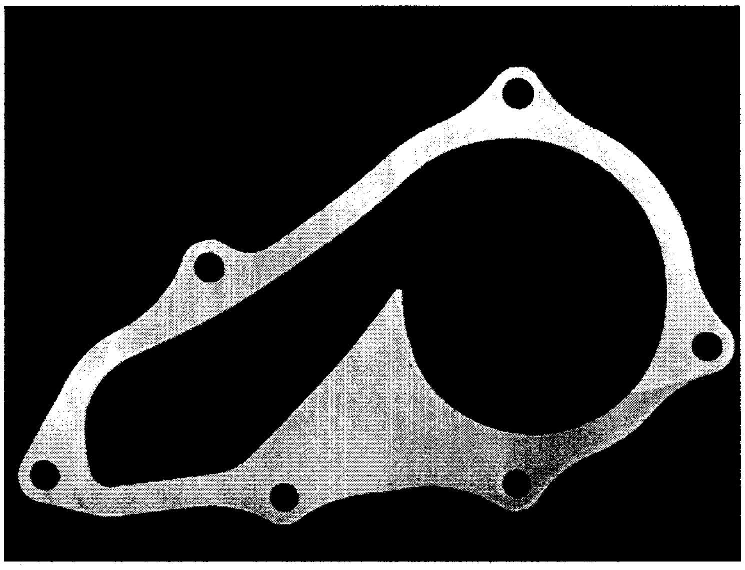


Figure 5.8: Pre-processed image of Figure 5.1(a)

5.5. Edge Extraction and Edge Function Development

In this section, we extract edges from the transformed reference image to the position of n^{th} test image and present them into the form of two separate functions. Section 5.5.1 is devoted to external edge extraction. Edge functions are constructed in Section 5.5.2.

5.5.1. Edge Extraction

$E^{(2)}_{i',j',n}$ is defined as an edge pixel in the i'^{th} row and the j'^{th} column of the transformed reference image to the position of the n^{th} test image. It is calculated using Equation

(5.35). $\mathbf{E}_\mathbf{n}^{(2)}$ is the corresponding tensor of $E^{(2)}_{i',\,j',\,n}$. Figure 5.9 shows edge extraction of the transformed reference image to the position of Figure 5.2.

$$E^{(2)}_{i',\,j',\,n} = \begin{cases} 1; \text{ if } \; 0 < \sum_{k_1=i'-1}^{i'+1} \sum_{k_2=j'-1}^{j'+1} B^{(6)}_{i',\,j',\,n} < 8 \\ \\ 0; \; \text{otherwise} \end{cases} \tag{5.35}$$

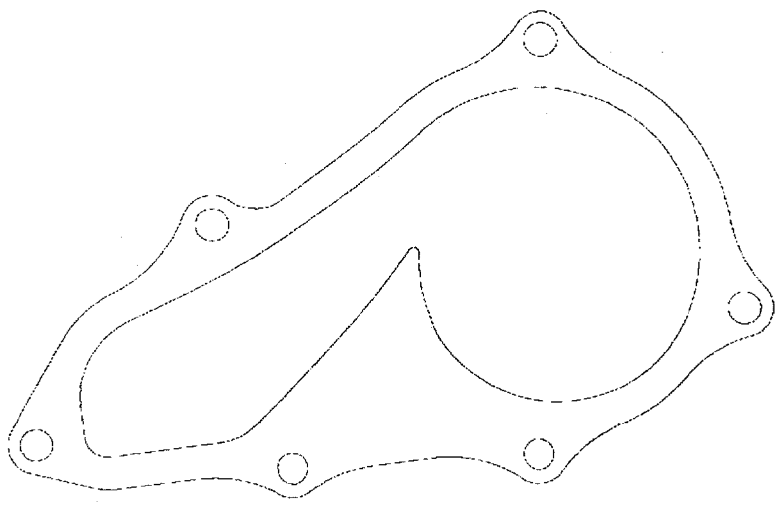


Figure 5.9: Edges of the transformed reference image for Figure 5.2

However, for the sake of simplicity in this chapter, we only focus on the detection of broken edges on the outer edge of the water pumps while the algorithm still remains robust to handle inner edge of water pumps or other products. We define $M_{i,j}$ as a one-piece object without any holes for the i^{th} row and the j^{th} column in the reference image using Equation (5.36). The corresponding matrix is **M**. Figure 5.10 illustrates **M**.

$$M_{i,j} = max\{R^{(2)}_{i,j}, R^{(4)}_{i,j,k}\} \tag{5.36}$$

where, k is the circle number in Figure 5.7. See Section 5.4.3 for the definition of $R^{(2)}_{i,j}$ and $R^{(4)}_{i,j,k}$. $B^{(8)}_{i',\,j',\,n}$ shows the repositioning of mask $\mathbf{M}$ upon the n^{th} test image. The corresponding tensor is shown by $\mathbf{B}^{(8)}_{\mathbf{n}}$.

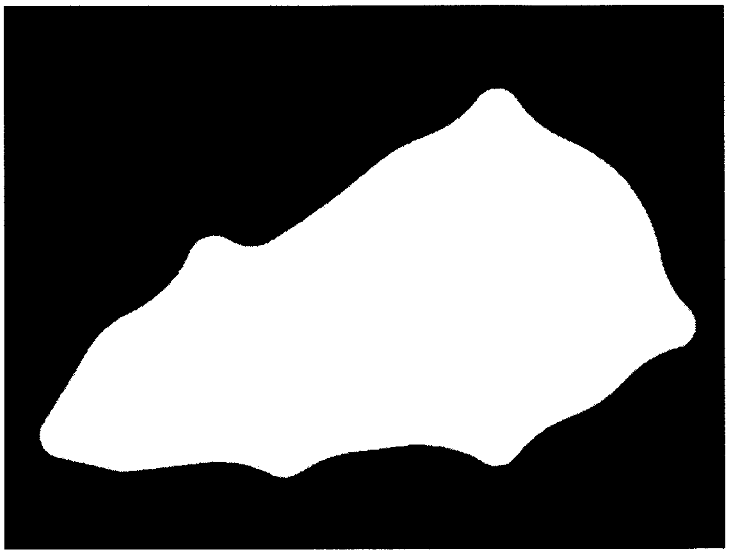


Figure 5.10: One piece reference object

$$B^{(8)}_{i',\,j',\,n} = \begin{cases} 1; & \text{if } M_{i,j} = 1 \\ 0; & \text{otherwise} \end{cases} \tag{5.37}$$

To extract external edges from the transformed reference image, $E^{(3)}_{i',\,j',\,n}$ is defined as an edge pixel in the i' row and the j' column of the transformed reference image to the position of the n^{th} image and is calculated by Equation (5.38). $\mathbf{E}^{(3)}_{\mathbf{n}}$ is the corresponding tensor of $E^{(3)}_{i',\,j',\,n}$. As a result, $\mathbf{E}^{(3)}_{\mathbf{n}}$ plays the role of a proper mask for edge extraction in this application. Figure 5.11 shows external edge extraction in the transformed reference image for the Figure 5.2.

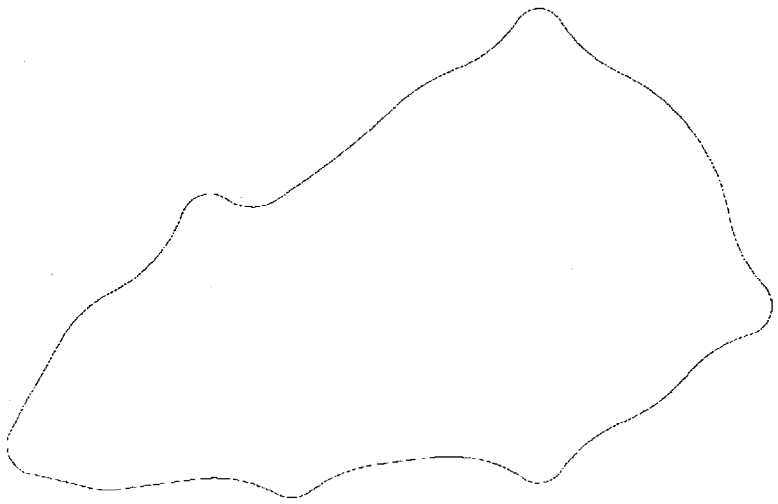


Figure 5.11: External edges of the transformed reference image for the Figure 5.2

$$E^{(3)}_{i',\,j',\,n} = \begin{cases} 1; \text{ if } & 0 < \sum_{k_1=i'-1}^{i'+1} \sum_{k_2=j'-1}^{j'+1} B^{(8)}_{i',\,j',\,n} < 8 \\ \\ 0; & \text{otherwise} \end{cases} \tag{5.38}$$

5.5.2. Edge Function Development

In this section, function development process is done to ensure that with a certain x value on the x axis, at most one pixel exists on y axis. This process is necessary in order to break the external edges of the image into two separate edge functions. Therefore, from the top curve only maximum y (lowest i') is taken whereas from the bottom curve only the minimum y (highest i') is taken. Figure 5.12 illustrates a simplified graphical view of this process. Equation (5.39) presents top edge function for the n^{th} test image whereas bottom edge function for the n^{th} test image is characterized in Equation (5.40).

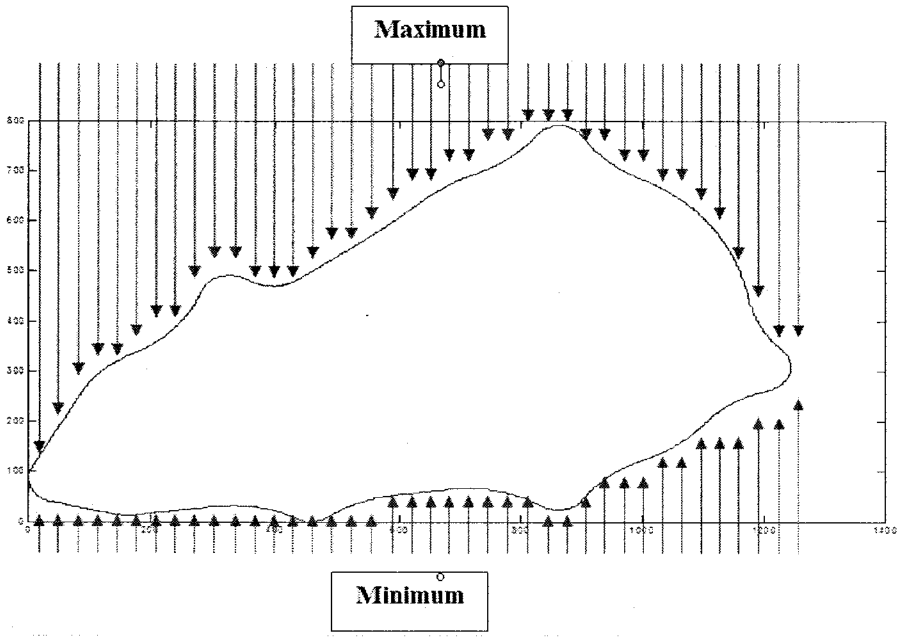


Figure 5.12: Breaking transformed reference image edges to the position of the n^{th} test image to two separate functions

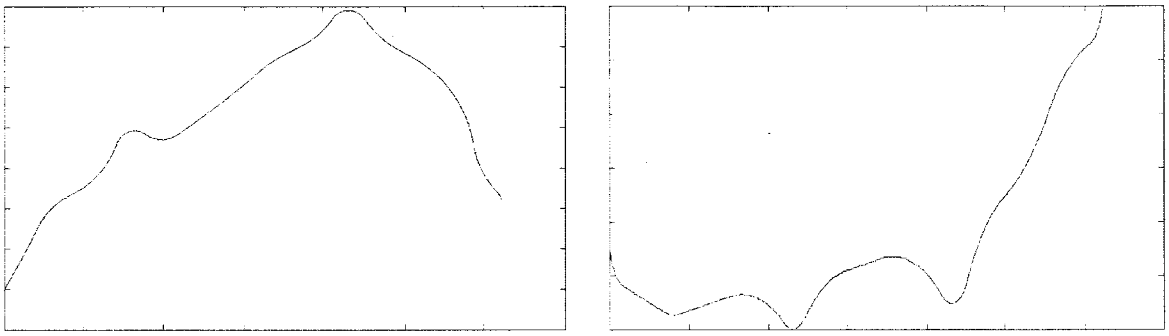
$$T_n(j') = Min_{i'}\{\text{if } E^{(3)}_{i',\,j',\,n} = 1\} \tag{5.39}$$

$$B_n(j') = Max_{i'}\{\text{if } E^{(3)}_{i',\,j',\,n} = 1\} \tag{5.40}$$

Now, let L_n be defined as the lower bound for the j' value which lies on the edges of the transformed reference image to the position of the n^{th} test image. Equation (5.41) presents L_n bound value.

$$L_n = Min_{j'}\{\text{if } E^{(3)}_{i',\,j',\,n} = 1\} \tag{5.41}$$

By converting $j' \longrightarrow j' - (L_n - 1)$, the lower bound of the two functions starts from one. Let U_n be defined as the higher bound of both functions. Therefore, we know the new range for both top and bottom functions is $k = [1, U_n]$.



(a) Top function

(b) Bottom function

Figure 5.13: Two separate functions of the edges of transformed reference image to the position of Figure 5.2

Figure 5.13(a) and 5.13(b) show top and bottom functions of the edges of transformed reference image to the position of Figure 5.2, respectively. In this stage, we know that k is the independent column value of the transformed reference image to the position of n^{th} test image, $T_n(k)$ and $B_n(k)$ are dependent row values of corresponding k for the top and bottom edge functions, $V_{T_n(k),k,n}$ and $V_{B_n(k),k,n}$ are dependent gray level values of corresponding k for the top and bottom edge functions, respectively. In a similar fashion, the gray level value of the edges of the reference image for column k in row $T_n(k)$ and row $B_n(k)$ are shown with $W_{T_n(k),k,n}$ and $W_{B_n(k),k,n}$ for top and bottom functions, respectively.

5.6. Broken Edge Feature Extraction

Feature extraction is carried out by mapping of a multidimensional space into a space of fewer dimensions. If the extracted features are deliberately chosen, it is expected that the feature set extracts the relevant information from the input data to perform the desired task using this reduced representation instead of the full size input. In this section we construct three features based on comparing test and reference functions. These features help us to make a distinction between broken edge and non broken edge parts. The best way to present the features is a vector. Hence, a vector corresponding to each pixel on the edges is needed to investigate whether a pixel lies on a broken edge or not. Here, each vector has three elements. The three features that make up the $\mathbf{g}_n(k)$ and $\mathbf{h}_n(k)$ vectors are explained in detail below. We define $\mathbf{g}_{n(k)}$ as the feature vector of k^{th} pixel on the top curve in the n^{th} image. Equation (5.42) shows the three features of the aforementioned vector. In a similar fashion, we define $\mathbf{h}_{n(k)}$ as the feature vector of k^{th} pixel on the bottom curve in the n^{th} image. $\mathbf{h}_{n(k)}$ is defined using Equation (5.43).

$$\mathbf{g}_n(k) = \left[\begin{array}{ccc} g_n^{(1)}(k) & g_n^{(2)}(k) & g_n^{(3)}(k) \end{array} \right] \tag{5.42}$$

$$\mathbf{h}_n(k) = \left[\begin{array}{ccc} h_n^{(1)}(k) & h_n^{(2)}(k) & h_n^{(3)}(k) \end{array} \right] \tag{5.43}$$

5.6.1. Pair Wise Edge Based Feature

The first feature captures the gray level difference of test and reference functions which means that if there is no significant broken edge then the value of this feature should be small. In order to reduce the amount of noise, each feature also takes into account the adjacent pixel values on its left and right sides. Therefore, the wider a broken edge, the larger the first feature. Equation (5.44) calculates the value of the first feature for the k^{th} pixel of the top function in the n^{th} test image. Similarly, Equation (5.45) calculates the value of the first feature for the k^{th} pixel of the bottom function in the n^{th} image.

$$g_n^{(1)}(k) = (\sum_{k'=k-3}^{k+3} (W_{T_n(\acute{k}),\acute{k},n} - V_{T_n(\acute{k}),\acute{k},n})/7) \tag{5.44}$$

$$\text{for } k = [4, U_n - 3]$$

$$h_n^{(1)}(k) = (\sum_{k'=k-3}^{k+3} (W_{B_n(\acute{k}),\acute{k},n} - V_{B_n(\acute{k}),\acute{k},n})/7) \tag{5.45}$$

$$\text{for } k = [4, U_n - 3]$$

5.6.2. Derivative Based Feature

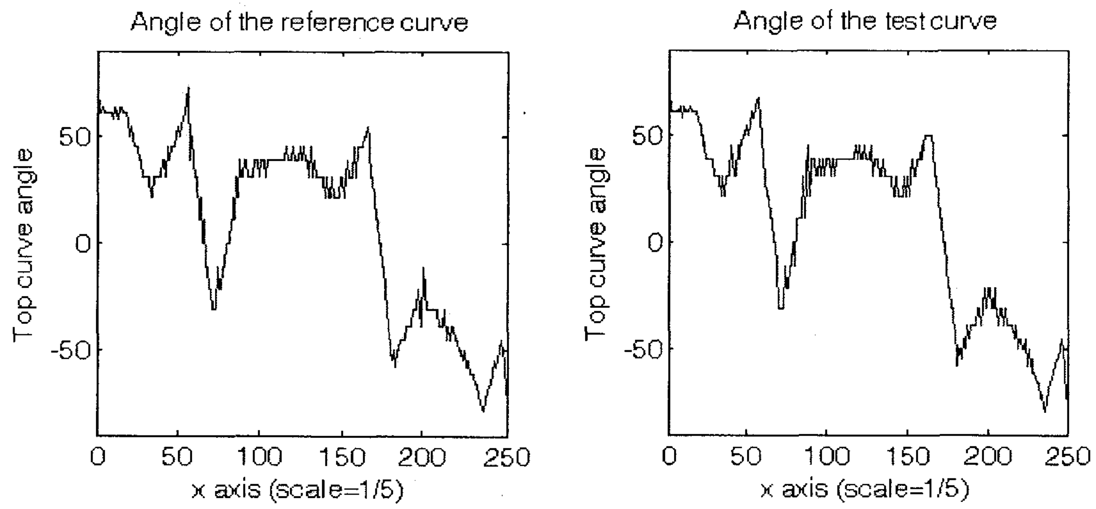
Using described methodology in Section 5.5, we assume k is the independent column value of the edges of the n^{th} image and $\acute{T}_n(k)$ and $\acute{B}_n(k)$ are dependent row values of corresponding k for the top and bottom test functions. It is important to note that to calculate $\acute{T}_n(k)$ and $\acute{B}_n(k)$ there is no need to have any transform of reference image for the n^{th} test image, anymore. So, we refer to $\acute{T}_n(k)$ and $\acute{B}_n(k)$ before their transformation. The whole idea in this section is to make a feature, based on the difference of the curve angle of the reference and test images. To do so, the first derivative function is needed.

There are two different ways to calculate the first derivative function. One way is to fit a polynomial on the top of each curve. For example spline fitting, which is basically a piecewise polynomial function, could be one way. Although this method is precise but it is computationally costly. The other way is to calculate the derivative of the curve directly from the available discrete function with acceptance some insignificant error. Since we are dealing with large sized images, from a computational point of view, we use the second way.

Let us call $\alpha_{k,n}^{(1)}$ the angle of top curve in the column k for the n^{th} image and $\alpha_{k,n}^{(2)}$ the angle of top curve in the column k for transformed reference image in the n^{th} image. Equations (5.46) and (5.47) show how $\alpha_{k,n}^{(1)}$ and $\alpha_{k,n}^{(2)}$ are calculated. We take a span of four to reduce the effect of noise on derivative calculations. Figure 5.14(a) and Figure 5.14(b) show the angle of top curve for reference and test images. As it is seen, there is no significant difference between these curves.

$$\alpha_{k,n}^{(1)} = Arctan((\acute{T}_n(k+2) - \acute{T}_n(k-2))/4) \;\; k = [4, U_n - 3] \tag{5.46}$$

$$\alpha_{k,n}^{(2)} = Arctan((T_n(k+2) - T_n(k-2))/4) \;\; k = [4, U_n - 3] \tag{5.47}$$



(a) Top reference curve

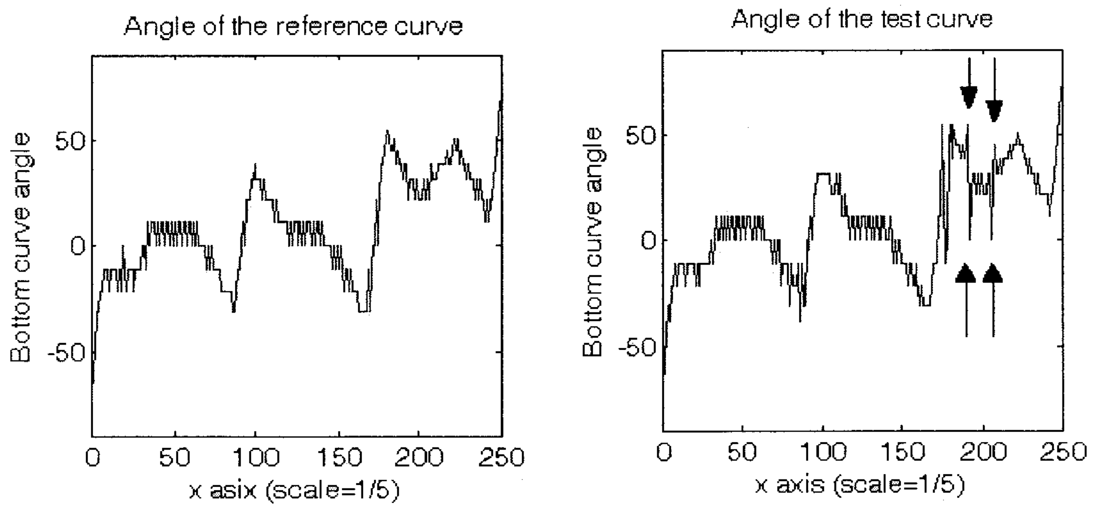
(b) Top test curve

Figure 5.14: Top function comparison

In a similar fashion, let us call $\beta_{k,n}^{(1)}$ the angle of bottom curve in the column k for the n^{th} test image and $\beta_{k,n}^{(2)}$ the angle of bottom curve in the column k for transformed reference image in the n^{th} image. Equations (5.48) and (5.49) show how $\beta_{k,n}^{(1)}$ and $\beta_{k,n}^{(2)}$ are calculated. We take a span of four to reduce the effect of noise on derivative calculations. Figure 5.15(a) and Figure 5.15(b) show the angle of bottom curve for reference and test images. As it is seen, there are significant differences between these curves exactly in the location of broken edges. For better understanding compare Figure 5.15(b) with the location of broken edges in Figure 5.2.

$$\beta_{k,n}^{(1)} = Arctan((\acute{B}_n(k+2) - \acute{B}_n(k-2))/4) \quad k = [4, U_n - 3] \tag{5.48}$$

$$\beta_{k,n}^{(2)} = Arctan((B_n(k+2) - B_n(k-2))/4) \quad k = [4, U_n - 3] \tag{5.49}$$



(a) Bottom reference curve

(b) Bottom test curve

Figure 5.15: Bottom function comparison

The second feature for the top and bottom functions is calculated by Equation (5.50) and Equation (5.51), respectively.

$$g_n^{(2)}(k) = |\alpha_{k,n}^{(1)} - \alpha_{k,n}^{(2)}| \quad k = [4, U_n - 3] \tag{5.50}$$

$$h_n^{(2)}(k) = |\beta_{k,n}^{(1)} - \beta_{k,n}^{(2)}| \quad k = [4, U_n - 3] \tag{5.51}$$

5.6.3. Band Edge Based Feature

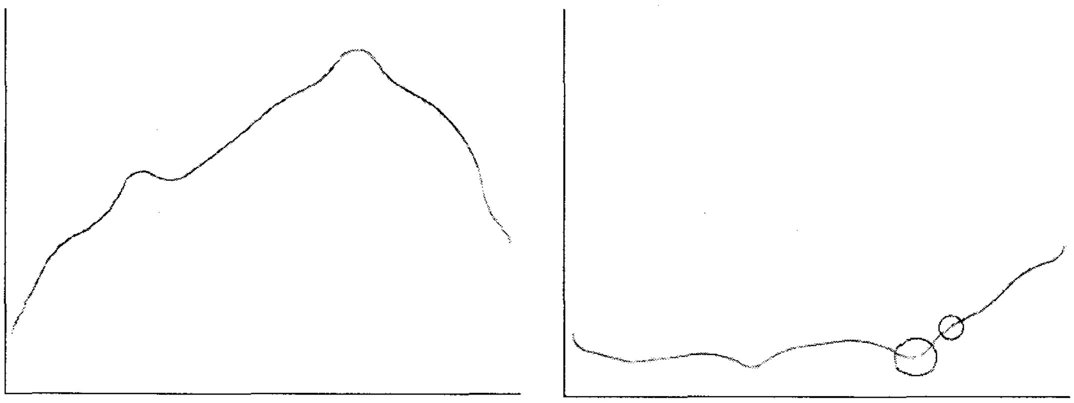
While the first feature targets the width of the broken edge, the third feature measures the depth of the broken edge. The value of $\alpha_{k,n}^{(2)}$ calculated in Section 5.6.2 gives a valuable information for broken edge classification. It says that if $-45 \leq \alpha_{k,n}^{(2)} \leq 45$ then the possible broken edge in the k^{th} column of the n^{th} image occurs vertically. Otherwise, the possible broken edge in the k^{th} column of the n^{th} image occurs horizontally. Therefore, we measure the depth of broken edges on a band with a depth of five pixels which aligns with the edges of the transformed reference images for the n^{th} image. Equation (5.52) shows the way by which the third feature is calculated for top function.

$$g_n^{(3)}(k) = \begin{cases} \sum_{k'=T_n(k)-2}^{T_n(k)+2} (W_{\acute{k},k,n} - V_{\acute{k},k,n})/5; & \text{if } |\alpha_{k,n}^{(2)}| \leq 45 \\ \\ \sum_{k'=k-2}^{k+2} (W_{T_n(k),\acute{k},n} - V_{T_n(k),\acute{k},n})/5; & \text{otherwise} \end{cases} \tag{5.52}$$

Similarly, Equation (5.53) shows by which the way the third feature is calculated for the bottom function.

$$h_n^{(3)}(k) = \begin{cases} \sum_{k'=B_n(k)-2}^{B_n(k)+2} (W_{\acute{k},k,n} - V_{\acute{k},k,n})/5; & \text{if } |\beta_{k,n}^{(2)}| \leq 45 \\ \\ \sum_{k'=k-2}^{k+2} (W_{B_n(k),\acute{k},n} - V_{B_n(k),\acute{k},n})/5; & \text{otherwise} \end{cases} \tag{5.53}$$

Figure 5.16(a) depicts the top gray level of the test band. As it is seen there is no significant gray level change in the band pixels. However, Figure 5.16(b) depicts the bottom gray level of test band. It is seen that the intensity of the gray level pixels of the band fades away indicating the existence of a broken edge. Figure 5.17 provides a close-up image. For better comparison, compare the location of broken edges in Figure 5.16(b) with Figure 5.2.



(a) Top gray level band

(b) Bottom gray level band

Figure 5.16: Top and bottom bands

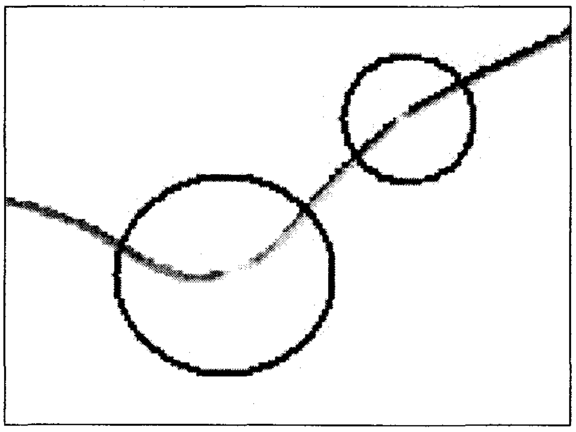


Figure 5.17: Close-up image of Figure 5.16(b)

5.7. Fuzzy Classification and Experimental Results

In order to reduce the impact of noise in decision making, we have calculated features from different methods. Therefore, there is a need to combine the obtained information to make the appropriate decision. Fuzzy logic which is a powerful problem solving methodology with a myriad of applications in information processing, provides such frame work for decision making. In this section, we describe fuzzy C-means clustering, the way it has been adopted in this research and the corresponding experimental results.

5.7.1. Fuzzy C-Means Clustering

Fuzzy C-means (FCM) algorithm, developed by [Dunn 1973] and later improved by [Bezdek 1981], is a technique of clustering which allocates one piece of data to two or more clusters. It is based on the minimization of the following objective function:

$$J_m = \sum_{i=1}^{N} \sum_{j=1}^{C} U_{ij}^{m} \| x_i - c_i \|^2 \qquad 1 \leq m \tag{5.54}$$

where, m is any real number greater than 1, u_{ij} is the degree of membership of x_i in the cluster j, x_i is the i^{th} d-dimensional measured data, c_j is the d-dimension center of the cluster, and $\| * \|$ is any norm expressing the similarity between any measured data and the center. Fuzzy partitioning is carried out through an iterative optimization of the objective function shown above, with the update of membership u_{ij} and the cluster centers c_j by:

$$u_{ij} = \frac{1}{\sum_{k=1}^{C} \left(\frac{\| x_i - c_j \|}{\| x_i - c_s \|} \right)^{\frac{2}{m-1}}} \tag{5.55}$$

$$c_j = \frac{\sum_{i=1}^{N} U_{ij}^{m} . x_i}{\sum_{i=1}^{N} U_{ij}^{m}} \tag{5.56}$$

This iteration will stop when $max_{ij}\{|u_{ij}^{(s+1)} - u_{ij}^{(s)}|\} < \varepsilon$, where ε is a termination criterion between 0 and 1 and s is the iteration step. This procedure converges to a local minimum or a saddle point of J_m. The algorithm is composed of the following steps:

1. Initialize $U = [u_{ij}]$ matrix, $U^{(0)}$

2. At s-step: calculate the centers' vectors $C^{(s)} = [c_j]$ with $U^{(s)}$ and $c_j = \frac{\sum_{i=1}^{N} U_{ij}^{m}.x_i}{\sum_{i=1}^{N} U_{ij}^{m}}$

3. Update $U^{(s)}, U^{(s+1)}$ and $u_{ij} = \frac{1}{\sum_{k=1}^{C} (\frac{\|x_i - c_j\|}{\|x_i - c_s\|})^{\frac{2}{m-1}}}$

4. If $\| U^{(s+1)} - U^{(s)} \| < \varepsilon$ then stop, otherwise return to step 2.

5.7.2. Broken Edge Candidate Clustering and Experimental Results

The features $g_n^{(1)}(k)$, $g_n^{(2)}(k)$ and $g_n^{(3)}(k)$ are numerical and the rules are extracted from the center of clusters. The similar framework can be constructed for the $h_n^{(1)}(k)$, $h_n^{(2)}(k)$ and $h_n^{(3)}(k)$ features. However, in order to decide whether a certain point is located on a broken edge or not, we need to determine the cluster centers. Pixels are classified into three different clusters as follows:

- First cluster: significant broken edge (defective part)
- Second cluster: insignificant broken edge (defective part)
- Third cluster: non broken edge (non defective part)

We define class 1 as significant broken edge parts, class 2 as insignificant broken edge parts and class 3 as non broken edge parts. The fuzzy rules are constructed as follows. The first cluster includes those pixels that can strongly be considered as part of a significant broken edge (defective part). The third cluster includes those pixels that can be considered as part of a non broken edge (non defective part). We define the second

Table 5.3: Cluster centers

	First cluster	Second cluster	Third cluster
First feature	0.882	0.523	0.295
Second feature	0.768	0.432	0.173
Third feature	0.653	0.343	0.054

cluster as a cluster which lies between the first and the third cluster. In other words, we may have one or more pixels located on insignificant broken edges in one image. If we are strict about the broken edges then this cluster can also be considered as a defective part, the same as the first cluster. In this chapter, we consider edges falling into the second cluster as a defective cluster to make the model more strict on broken edge defects.

Ten thousand points from ten different images are used in training step to find the aforementioned cluster centers. Using $\mathbf{g}_n(k)$ and $\mathbf{h}_n(k)$ vectors in Section 5.6 and the described methodology in Section 5.7.1, we find the cluster centers after $s = 13$ iterations $(\varepsilon = \frac{1}{1000})$.

Table 5.3 shows the cluster center values. The corresponding fuzzy C-means clustering objective function is shown in Figure 5.18. The next task is to measure the Euclidean distance between each point and the cluster centers. Each point (vector) falls into the category (cluster) from which it has the minimum Euclidean distance.

Consequently, 150 images have been tested with 55 broken edges with the following results: 52 broken edges have been recognized (about 95%), 3 broken edges have not been recognized (false negative about 5%) and 8 tool marks have been mistakenly recognized as broken edge (false positive about 5%). Table 5.4 shows broken edge chart used to score the fuzzy recognizer performance against the truth field defined by the human expert. The TP represents true positive results, FN is false negative results, FP is false positive results

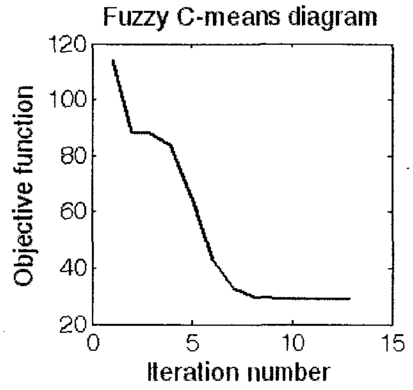


Figure 5.18: Fuzzy C-means clustering objective function

Table 5.4: Fuzzy machine vision based broken edge detection model performance

Results	Model truth (T)	Model truth (F)
Expert truth (T)	TP = 94.55 %	FN = 5.45 %
Expert truth (F)	FP = 5.33 %	TN = 94.67 %

and TN is true negative results. The processing time per image was about 150 seconds on a Pentium 4 (3 GHz), 1 GB RAM and a Windows XP platform (90 seconds for image pre-processing and 60 seconds for broken edge detection).

5.8. Conclusions

The current research outlines experience with one industrial example which involves the broken edge inspection of die cast aluminum automotive water pump housings. In this research, we have developed an objective fuzzy approach for fast and accurate machine vision based broken edge detection. The proposed method not only detects broken edge defects as long as they break the homogeneity of textures but also it can determine the types of broken edge defects according to their levels of severity, developed during casting the parts from aluminum alloy. Three new features for broken edge detection are also proposed. The first feature, pair wise edge based feature aims to compare the width of broken edge of test and reference images. The second feature, derivative based feature compares the derivative of test and reference function images and finally the third feature, band edge based feature measures the depth of broken edge of test and reference images. The fuzzy decision making on broken edge detection adopted and presented in this chapter, adds great value to the whole production system by increasing the confidence of the inspectors in the machined based performance of the real-time broken edge verification. The results demonstrated that the developed system can provide precise result of perceptual edge features in a computationally efficient way. The results demonstrated that about 95% of the broken edges in the entire database is correctly identified. It is worth mentioning that the proposed model can easily be tuned and implemented for inspection purpose in many assembly lines.

Chapter 6

Fuzzy Machine Vision Based Porosity Detection

6.1. Introduction

Current world trends are connected generally with the development of the automotive and aircraft industries, where lightweight and safe designs are in high demand. The aluminum alloys technologies, those currently used and of the future, consist mostly of casting to metal moulds. As a result, significant growth of the demand for castings from the aluminum alloys is of paramount importance.

However, gas porosity in molten aluminum alloys is a common problem. Porosity defects in metallic parts specially aluminum alloys result from gas bubbles trapped during casting or sintering processes. Once the surface is machined away, the pores are revealed on the surface and may be observed visually. Defect porosity parts should be identified before the assembly process because they may cause sealing failures or stress concentrations. The variation in porosity, average pore size and pore size distribution significantly influences the mechanical and textural characteristics of products [Huang and

Clayton 1990]. Pores also affect sensory properties and have a direct impact on the other physical properties, such as mass diffusion coefficient, thermal conductivity, and thermal diffusivity [Rahman 2001]. In the literature, much research effort has been done directed at studying the porosity of different kinds of material, such as [Du and Sun 2005, Hangai and Kitahara 2009].

Porosity defects are traditionally detected through manual inspection which includes creation of vision concentration to a candidate pore. The human inspector may rotate the part to create multiple view points and by doing so alters the reflection properties. In spite of the fact that humans are quite good at inspection tasks for short periods of time [Killing *et al.* 2007], variations in performance associated with human inspection quality and limited speed have motivated the use of automatic solutions. Repeatability and reproductivity of the analysis results, reduction of the time needed for labor-consuming examinations and also extension of the investigation capability, should be highlighted among the benefits of employing the machine vision based methods [Barth and Barrows 2003, Dobrzanski *et al.* 2005, Cucchiara *et al.* 2007]. As a result, it was reported that about 60-90% of all existing machine vision applications were classified as automated visual inspection [Awcock and Thomas 1996].

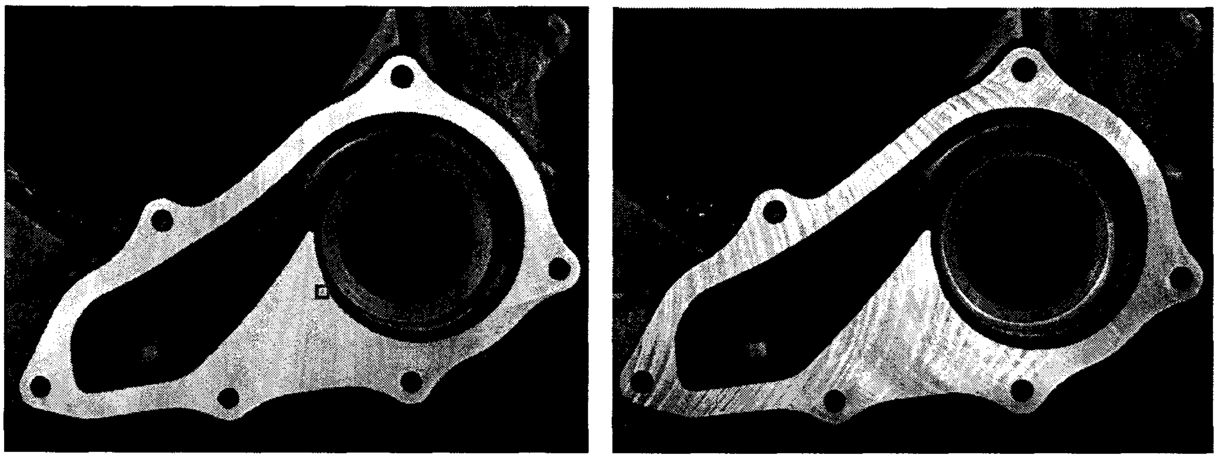
Recent developments in image processing, artificial intelligence, fuzzy logic and other related fields have significantly improved the capability of visual inspection techniques [Fletcher 1996, Rao 2001, Ong and Wang 1995, Peters *et al.* 2002]. The contribution of fuzzy sets and logic to image processing, pattern recognition and machine intelligence is significant [Tunstel *et al.* 2002, Iravani Tabrizipour and Toyserkani 2007, Mitra and Pal 2005b]. Fuzzy set theory has been also extensively used in clustering problems where the task is to provide class labels to input data (partitioning of feature space) under unsupervised mode based on certain criterion. A seminal contribution to cluster analysis was Ruspini's concept of a fuzzy partition [Ruspini 1969]. This was followed by the design of fuzzy c-means, fuzzy ISODATA, fuzzy DYNOC and other possibilistic clustering algorithms [Bezdek 1981, Bezdek and Pal 1992b].

This chapter describes a project whose objective is to improve the parts inspection component of manufacturing in the automotive parts industry through the application of intelligent fuzzy systems. The presented methodology may be crucial, for manufacturers of car subassemblies, where meeting the strict quality aspects ensures the required service life of the manufactured products. Hence, regarding the general observations and/or assumptions of pore properties, this chapter describes a fuzzy modeling framework for detecting the pores by clustering, using machine vision technology in the car subassembly castings from aluminum alloys. The current research will outline experience with one industrial example which involves the porosity inspection of die cast aluminum automotive water-pump housings. Some developments have been previously reported by [Killing *et al.* 2007].

The remaining part of this chapter is organized as follows. Section 2 briefly describes the nature of the problem considered for analysis. In Section 3, we describe the pre-processing process. In Section 4, the way the pore candidates which have been selected is determined. We explain the proposed features in Section 5. In Section 6, the proposed fuzzy clustering model and corresponding experimental results have been discussed. Finally, we draw the conclusions in Section 7.

6.2. Problem Definition

In our database, we have 105 water-pump images where, 35 of them have no pores while the rest have significant or semi significant pores. All the initial images have been captured in standard Tagged Image File Format (Tiff). They are all gray level images with a size of 960 by 1280. Figure 6.1(a) shows a generic porosity defect image with one pore without considerable tool marks whereas Figure 6.1(b) illustrates a typical non-porosity image with a lot of tool marks. The pore is depicted within the black rectangle in Figure 6.1(a). Figure 6.2 gives a closer look at the porous area.



(a) Image with one pore

(b) Image with no pores

Figure 6.1: A defect and a non-defect part

The main complexity of this problem results from the similarity of porosity and tool marks. Therefore, making a distinction between porosity and tool marks could be a ,very challenging task. It should also be noted that porosity detection is extremely sensitive to noise. This arises a defying situation since we are dealing with rapid changes in light conditions from one image to another as well as swift changes in light conditions from one area of image to another. In addition to porosity detection problem, reflection is another specific problem of machined metallic surfaces. This in turn introduces variations in the appearance of defect-free surfaces as well as in the appearance of the porus surfaces. In this work we approach to porosity detection problem as a clustering problem and refer to fuzzy c-means algorithm for decision making. Using required preprocessing steps, the main objective of this work is to develop an efficient fuzzy machine vision algorithm to correctly identify parts with porosity. In the following sections, we introduce the steps

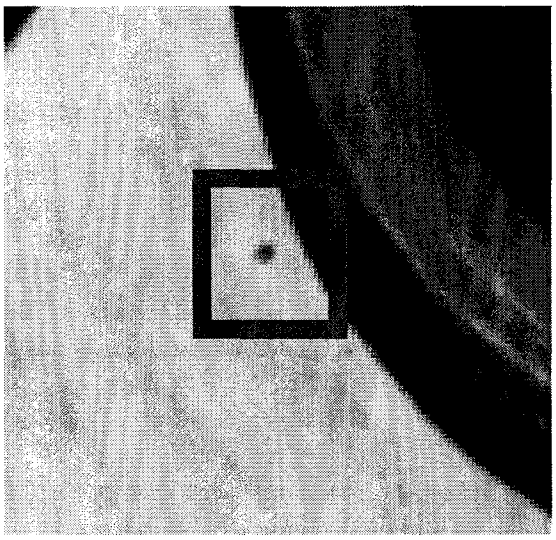


Figure 6.2: A closer look at the porous area in Figure 6.1(a)

that should be taken in order to reach the desirable results. The machine vision has four main steps [Davies 2005a] as follows:

1. *Image acquisition*: The acquired digital image is quantized in both the spatial coordinates and levels. Initial images in our database are the outputs of this step.

2. *Pre-processing*: Pre-processing is carried out on the image to align images for getting a consolidated scene, such as noise filtering, image segmentation and the application of geometric corrections. Using industrial water-pump images, we illustrate this part in Section 3.

3. *Feature extraction*: In this step, the properties of each image are translated to feature vector since such representations facilitate processing and statistical analysis. Sections 4 and 5 describe our proposed features for porosity detection.

4. *Recognition*: Classification or clustering is made based on feature vectors for decision making. This step is explained in Section 6 with details. In the context of this

study, we classify porosity into three categories as follows:

(a) Parts with strong pores (strong porosity)

(b) Parts with weak pores (weak porosity)

(c) Parts without pores (no porosity)

6.3. Pre-processing

In this section, we introduce the steps that should be taken in order to reach to the registered image. First, we create a binary image, then we clear all the noises and redundant objects in order to detect the main object (water-pump surface). Afterward, the test image is adjusted horizontally and vertically, based on the reference image, and its location is set (displacement and orientation adjustment). Finally, the zone of interest for each image is automatically determined. Since similar steps were explained in details in Chapter 5, we don't explain them here. Readers can refer to Section 5.4.

As a result, Figure 6.3 shows the registered image of Figure 6.1(a). Notably, Figure 6.1(a) has 1228800 pixels (960 by 1280) for consideration while the zone of interest of registered test image of Figure 6.1(a) has 244531 pixels (80.1% reduction).

6.4. Pore Candidate Selection

In this section, the way the pore candidates have been selected is described. At first, we explain some observations and considerations about the pores. Then, our strategy to screen those pixels located on porosity spots is described.

On a 2-D projection, pores cannot be expressed according to a warped, scaled or rotated texture. Moreover, porosity spots are typically darker than their background but

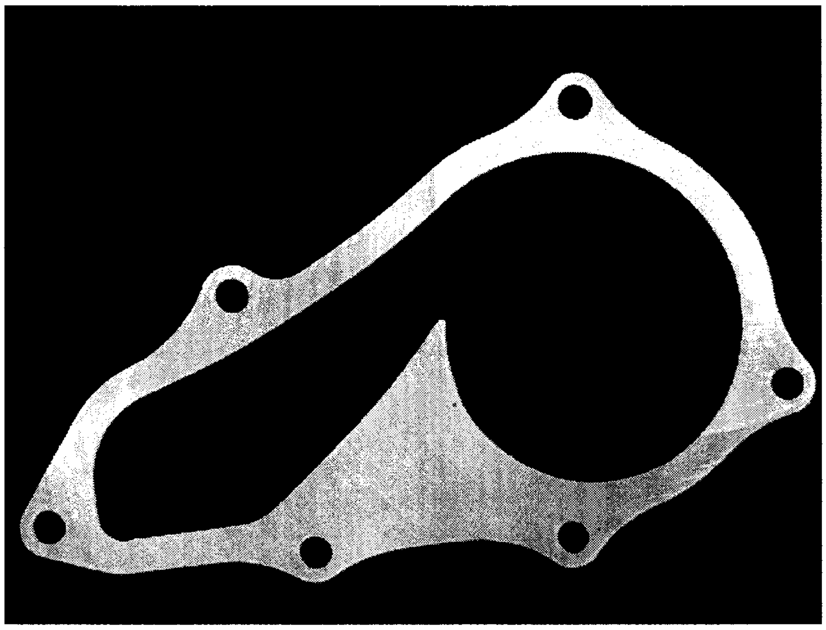


Figure 6.3: Registered image of Figure 6.1(a)

there may be other dark spots. They do not have homogenous textural signature thus cannot serve as a part of the textural representation. Furthermore, pores are too small and they are scattered at random locations. They are rarely like polyhedrons and also their shapes are random. In addition, images have pores and tool marks with different and random densities. As a result, porosity defect detection strategies, based directly on object recognition, are not applicable.

Moreover, it is important to note that it is not practical to check all the pixels on the surface of the registered water-pump image ($\mathbf{B}_{\mathbf{n}}^{(7)}$) to calculate their corresponding features (as explained in Section 5). Therefore, we need to filter out as many of the pixels on the surface of the image as possible. Only, those pixels that pass the filter will be qualified as pore potential candidates. To do so, our strategy includes the following three steps:

- Step 1: identify pixels that are the core of porosity as the preliminary pore candidates
- Step 2: apply a suitable threshold to eliminate light spots
- Step 3: apply certain minimum condition to get fewer candidates as the final pore candidates

In the first step, the core of porosity is found. We define core of porosity as a pixel on the porosity which its gray level value is less than or equal to all its eight neighbors' gray level values. It is clear from this definition that a porosity may have more than one core. Therefore, preliminary pixel candidates that we are looking for are local minimum pixels located on the surface of water-pump image. Equation (6.1) calculates local minimum pixels in the n^{th} test image.

$$P^{(1)}_{\acute{i}_n,\acute{j}_n,n} = \begin{cases} 0; & \text{if } B^{(7)}_{\acute{i}_n,\acute{j}_n,n} \le B^{(7)}_{a_n,b_n,n} \\ \\ 1; & \text{otherwise} \end{cases} \tag{6.1}$$

where, a_n and b_n are integers, $\acute{i}_n - 1 \le a_n \le \acute{i}_n + 1$ and $\acute{j}_n - 1 \le b_n \le \acute{j}_n + 1$. Therefore, if a pixel on the $\acute{i}_n$ row and the $\acute{j}_n$ column of the n^{th} registered test image is a local minimum pixel, then $P^{(1)}_{\acute{i}_n,\acute{j}_n,n}$ is zero. $\mathbf{P}^{(1)}_{\mathbf{n}}$ is the corresponding tensor of $P^{(1)}_{\acute{i}_n,\acute{j}_n,n}$. Equation (6.1) clearly shows that $P^{(1)}_{\acute{i}_n,\acute{j}_n,n}$ has a binary value. Gray level values of local minima are calculated using Equation (6.2), which means that the gray level value of a local minimum on the $\acute{i}_n$ row and the $\acute{j}_n$ column of the n^{th} registered test image is shown by $P^{(2)}_{\acute{i}_n,\acute{j}_n,n}$. $\mathbf{P}^{(2)}_{\mathbf{n}}$ is the corresponding tensor of $P^{(2)}_{\acute{i}_n,\acute{j}_n,n}$. Figure 6.4 shows the local minimum pixels of Figure 6.1(a).

$$P^{(2)}_{\acute{i}_n,\acute{j}_n,n} = \begin{cases} B^{(7)}_{\acute{i}_n,\acute{j}_n,n}; & \text{if } P^{(1)}_{\acute{i}_n,\acute{j}_n,n} = 1 \\ \\ 255; & \text{otherwise} \end{cases} \tag{6.2}$$

Notably, the zone of interest of the registered test image of Figure 6.1(a) has 244531 pixels for pore candidate consideration while the local minimum pixel image of Figure 6.1(a), Figure 6.4 has only 14563 pixels (94.1% reduction in the first step).

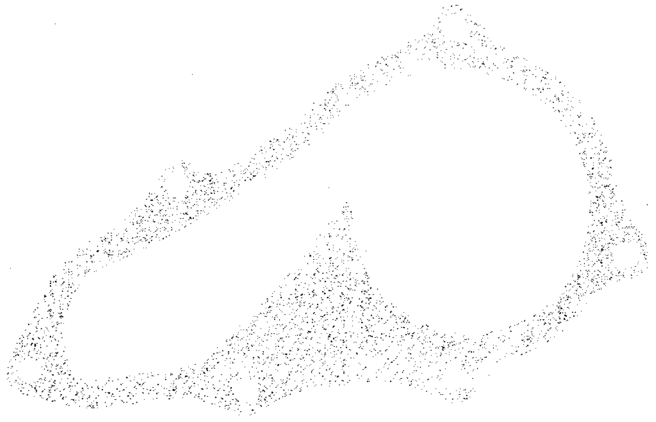


Figure 6.4: Local minimum pixels of Figure 6.1(a)

In the second step, a proper threshold is applied to eliminate light spots. Obviously, pores cannot be white spots. This fact results in a more reduction in the number of pore candidates. Finding the optimal value of aforementioned threshold highly depends on the database of images which is being dealt with and there is always a trade off between keeping a proper set of pore candidates and processing time. However, searching for the optimal threshold is not the purpose of this research. In this application, we choose a conservative threshold as large as $\tau_2 = 160$. In other words, it means that if the local minimum in a pixel neighborhood is larger than 160, that pixel could not be considered as a pore candidate in this step. Equation (6.3) calculates minimum pixels on the n^{th} test image with the applied threshold ($\tau_2 = 160$). It means that if a pixel on the $\acute{i_n}$ row and the $\acute{j_n}$ column of the n^{th} registered test image is a local minimum pixel with the applied threshold, then $P^{(3)}_{\acute{i_n},\acute{j_n},n}$ is zero. $\mathbf{P_n^{(3)}}$ is the corresponding tensor of $P^{(3)}_{\acute{i_n},\acute{j_n},n}$. Equation (6.3) evidently shows that $P^{(3)}_{\acute{i_n},\acute{j_n},n}$ has a binary value. Gray level values of local minima with the applied threshold are calculated using Equation (6.4), which means that the gray level value of a local minimum with the applied threshold in the $\acute{i_n}$ row and $\acute{j_n}$ column of the n^{th} registered test image is shown by $P^{(4)}_{\acute{i_n},\acute{j_n},n}$. $\mathbf{P_n^{(4)}}$ is the corresponding tensor of $P^{(4)}_{\acute{i_n},\acute{j_n},n}$.

$$P^{(3)}_{\acute{i_n},\acute{j_n},n} = \begin{cases} 0; & \text{if } P^{(2)}_{\acute{i_n},\acute{j_n},n} \leq 160 \\ 1; & \text{otherwise} \end{cases} \tag{6.3}$$

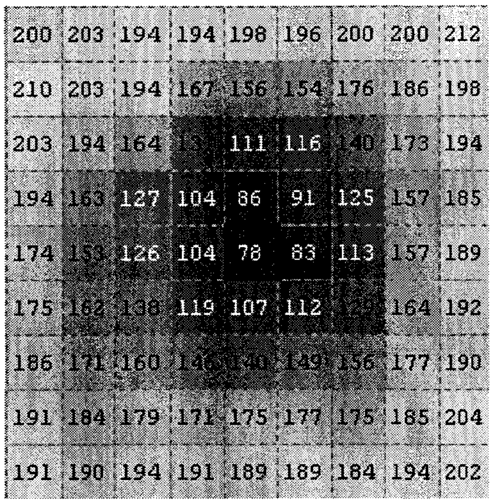
$$P^{(4)}_{\acute{i_n},\acute{j_n},n} = \begin{cases} P^{(2)}_{\acute{i_n},\acute{j_n},n}; & \text{if } P^{(3)}_{\acute{i_n},\acute{j_n},n} = 0 \\ 255; & \text{otherwise} \end{cases} \tag{6.4}$$

Now, it is possible to count the number of pore candidates in the second step. We choose $m_n^{(2)}$ as the number of pore candidates in the n^{th} image in the second step. $P^{(5)}_{1,k_n,n}$, $P^{(5)}_{2,k_n,n}$ and $P^{(5)}_{3,k_n,n}$ are also considered as the corresponding row, column, and gray level value of the ${k_n}^{th}$ pore candidate in the n^{th} test image in the second step, respectively, where, $0 \leq k_n \leq m_n^{(2)}$. Notably, the local minimum pixel image of Figure 6.1(a) with the applied threshold has now only 3962 pixels (72.1% reduction in the second step).

In the third step, we try to reduce pixel pore candidates once more. Figure 6.5 shows a close-up image of porosity spot for the Figure 6.1(a). As it is seen in this figure, the core of porosity is shown in the center of the image witch has a gray level value of 78. In this figure, we notice that the first periphery layer around the core of porosity (its direct eight neighbors) has greater gray level values than the core of porosity. However, this is not an unexpected observation, since we have mentioned earlier that the core of porosity is a local minimum.

In this tradition, we also notice that the second periphery layer around the core of porosity has greater gray level values (more or less) than the first periphery layer. The same analogy is also true when it comes to the comparison of the third layer values with the second layer values. It is important to note that the periphery layers of the core of porosity can be considered straight or diagonally. In short, it can be experimentally concluded that as we get farther from the core of porosity to the inner periphery layers and from the inner periphery layers to the outer periphery layers, the average gray level value of pixels are getting larger and larger. One may argue what if when we hit another porosity or tool mark in the outer layers ? We have two answers: first, if in a neighborhood of a core of porosity, we hit a second core of porosity or tool mark (hit a second local minimum) and the second core of porosity or tool mark has a value lower than the first one, when it comes to consideration of second porosity then the algorithm still can select the second one as a pore candidate. Second, when we have more than one core with some distances from each other, then logically this is a large spot while, as we mentioned earlier in this section, pore spots are small, therefore, this large spot should not be selected as a pore candidate.

Figure 6.6 shows four matrices $A^{(1)}$, $A^{(2)}$, $A^{(3)}$ and $A^{(4)}$. These matrices have been used in Equation (6.7) to calculate their correlation with the pixel pore candidates in the second step. These matrices help us to ensure that the gray level average value of the core of porosity neighborhood is smaller than the gray level average value of the farther pixels located on the straight (horizontal and vertical) sides of the given spot resulted from the second step. $S^{(q)}_{k_n,n}$ shows the correlation value of the k_n^{th} local minimum pixel with the applied threshold in the n^{th} test image with the q^{th} matrix shown in Figure 6.6.



200	203	194	194	198	196	200	200	212
210	203	194	167	156	154	176	186	198
203	194	164	[illegible]	111	116	140	173	194
194	163	127	104	86	91	125	157	185
174	153	126	104	78	83	113	157	189
175	162	138	119	107	112	129	164	192
186	171	160	146	140	149	156	177	190
191	184	179	171	175	177	175	185	204
191	190	194	191	189	189	184	194	202

Figure 6.5: Porosity spot

$$S^{(q)}_{k_n,n} = \sum_{i=-3}^{3}\sum_{j=-3}^{3} A^{(q)}_{i+4,j+4} \times B^{(7)}_{P^{(5)}_{1,k_n,n}+i,P^{(5)}_{2,k_n,n}+j,n} \tag{6.5}$$

for $q = 1, ..., 4$

Figure 6.7 shows four matrices $A^{(5)}$, $A^{(6)}$, $A^{(7)}$ and $A^{(8)}$. These matrices have been used in Equation (6.6) to calculate their correlation with the pixel pore candidates in the second step. These matrices help us to ensure that the gray level average value of the core of porosity neighborhood is smaller than the gray level average value of the farther pixels located on the diagonal sides of the given spot resulted in the second step. $D^{(q)}_{k_n,n}$ shows the correlation value of the k^{th}_n local minimum pixel with the applied threshold in the n^{th} test image with the q^{th} matrix shown in Figure 6.7.

$$D^{(q)}_{k_n,n} = \sum_{i=-4}^{4}\sum_{j=-4}^{4} A^{(q+4)}_{i+5,j+5} \times B^{(7)}_{P^{(5)}_{1,k_n,n}+i.P^{(5)}_{2,k_n,n}+j.n} \tag{6.6}$$

for $q = 1, ..., 4$

Since the gray level average value of the core of porosity neighborhood should be smaller than the gray level average value of the pixels located on the straight and diagonal sides of the given spot resulted from the second step, the correlation values of the core of porosity neighborhood with all matrices in Figures 6.6 and 6.7 should be positive if the pixel spot belongs to a porosity spot. Therefore, we only consider those pixels which satisfy the Inequality (6.7) as the final pore candidates. We choose $m_n^{(3)}$ as the number of final pore candidates in the n^{th} image. $P^{(6)}_{1,k'_n,n}$, $P^{(6)}_{2,k'_n,n}$ and $P^{(6)}_{3,k'_n,n}$ are also considered as the corresponding row, column, and value of the $k_n'^{th}$ final pore candidate in the n^{th} test image, respectively, where, $0 \leq k'_n \leq m_n^{(3)}$. Notably, the number of the final pore candidates of Figure 6.1(a) is now only 268 pixels (93% reduction in the third step).

$$\{\ k_n \mid Min_{q=1,\dots,4}\{S^{(q)}_{k_n,n}, D^{(q)}_{k_n,n}\} > 0\ \} \qquad (6.7)$$

6.5. Pore Feature Selection

In image processing and machine vision, feature extraction is a special form of dimensionality reduction. Image features at various levels of complexity are extracted from the image data. A feature based approach reduces the size of the data set and improves the efficiency of the algorithm [Gonzales *et al.* 2004]. It involves the simplification of the amount of resources required to describe a large set of data accurately. Unfortunately, analysis with a large number of variables generally requires a large amount of memory and computation power. Therefore, the fewer the number of features is, the more efficient the algorithm is. Best results are achieved when a set of application dependent features is constructed. In this chapter, we propose five features for clustering. These features help us to make a distinction between pore spots and tool mark spots. Since the best way to

$$\begin{bmatrix} \frac{1}{7} & \frac{1}{7} & \frac{1}{7} & \frac{1}{7} & \frac{1}{7} & \frac{1}{7} & \frac{1}{7} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{-1}{8} & \frac{-1}{8} & \frac{-1}{8} & 0 & 0 \\ 0 & 0 & \frac{-1}{8} & 0 & \frac{-1}{8} & 0 & 0 \\ 0 & 0 & \frac{-1}{8} & \frac{-1}{8} & \frac{-1}{8} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

(a) Top straight filter matrix, $A^{(1)}$

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{-1}{8} & \frac{-1}{8} & \frac{-1}{8} & 0 & 0 \\ 0 & 0 & \frac{-1}{8} & 0 & \frac{-1}{8} & 0 & 0 \\ 0 & 0 & \frac{-1}{8} & \frac{-1}{8} & \frac{-1}{8} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \frac{1}{7} & \frac{1}{7} & \frac{1}{7} & \frac{1}{7} & \frac{1}{7} & \frac{1}{7} & \frac{1}{7} \end{bmatrix}$$

(b) Bottom straight filter matrix, $A^{(2)}$

$$\begin{bmatrix} \frac{1}{7} & 0 & 0 & 0 & 0 & 0 & 0 \\ \frac{1}{7} & 0 & 0 & 0 & 0 & 0 & 0 \\ \frac{1}{7} & 0 & \frac{-1}{8} & \frac{-1}{8} & \frac{-1}{8} & 0 & 0 \\ \frac{1}{7} & 0 & \frac{-1}{8} & 0 & \frac{-1}{8} & 0 & 0 \\ \frac{1}{7} & 0 & \frac{-1}{8} & \frac{-1}{8} & \frac{-1}{8} & 0 & 0 \\ \frac{1}{7} & 0 & 0 & 0 & 0 & 0 & 0 \\ \frac{1}{7} & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

(c) Left straight filter matrix, $A^{(3)}$

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{7} \\ 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{7} \\ 0 & 0 & \frac{-1}{8} & \frac{-1}{8} & \frac{-1}{8} & 0 & \frac{1}{7} \\ 0 & 0 & \frac{-1}{8} & 0 & \frac{-1}{8} & 0 & \frac{1}{7} \\ 0 & 0 & \frac{-1}{8} & \frac{-1}{8} & \frac{-1}{8} & 0 & \frac{1}{7} \\ 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{7} \\ 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{7} \end{bmatrix}$$

(d) Right straight filter matrix, $A^{(4)}$

Figure 6.6: Components of straight filter

present the features is a vector, a vector corresponding to each pore candidate is needed. Obviously, each vector has five elements. We define $\mathbf{V}_{\acute{k}_n,n}$ as the feature vector of the $\acute{k}_n$ final pore candidate in the n^{th} image. Equation (6.8) shows the five features of the aforementioned vector. Each feature is calculated as follows:

$$\begin{bmatrix} 0 & 0 & 0 & 0 & \frac{1}{5} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{5} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{5} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{5} & 0 & \frac{-1}{8} & \frac{-1}{8} & \frac{-1}{8} & 0 & 0 & 0 \\ \frac{1}{5} & 0 & 0 & \frac{-1}{8} & 0 & \frac{-1}{8} & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{-1}{8} & \frac{-1}{8} & \frac{-1}{8} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

(a) Top left diagonal filter matrix, $A^{(5)}$

$$\begin{bmatrix} 0 & 0 & 0 & 0 & \frac{1}{5} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1}{5} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{5} & 0 & 0 \\ 0 & 0 & 0 & \frac{-1}{8} & \frac{-1}{8} & \frac{-1}{8} & 0 & \frac{1}{5} & 0 \\ 0 & 0 & 0 & \frac{-1}{8} & 0 & \frac{-1}{8} & 0 & 0 & \frac{1}{5} \\ 0 & 0 & 0 & \frac{-1}{8} & \frac{-1}{8} & \frac{-1}{8} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

(b) Top right diagonal filter matrix, $A^{(6)}$

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{-1}{8} & \frac{-1}{8} & \frac{-1}{8} & 0 & 0 & 0 \\ \frac{1}{5} & 0 & 0 & \frac{-1}{8} & 0 & \frac{-1}{8} & 0 & 0 & 0 \\ 0 & \frac{1}{5} & 0 & \frac{-1}{8} & \frac{-1}{8} & \frac{-1}{8} & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{5} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{5} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{5} & 0 & 0 & 0 & 0 \end{bmatrix}$$

(c) Bottom left diagonal filter matrix, $A^{(7)}$

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{-1}{8} & \frac{-1}{8} & \frac{-1}{8} & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{-1}{8} & 0 & \frac{-1}{8} & 0 & 0 & \frac{1}{5} \\ 0 & 0 & 0 & \frac{-1}{8} & \frac{-1}{8} & \frac{-1}{8} & 0 & \frac{1}{5} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{5} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1}{5} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{5} & 0 & 0 & 0 & 0 \end{bmatrix}$$

(d) Bottom right diagonal filter matrix, $A^{(8)}$

Figure 6.7: Components of diagonal filter

$$\mathbf{V}_{k_n^{\prime},n} = \begin{bmatrix} V_{k_n^{\prime},n}^{(1)} \\ V_{k_n^{\prime},n}^{(2)} \\ V_{k_n^{\prime},n}^{(3)} \\ V_{k_n^{\prime},n}^{(4)} \\ V_{k_n^{\prime},n}^{(5)} \end{bmatrix} \tag{6.8}$$

1. The first feature value is the correlation of pore candidate spot with $F^{(1)}$ matrix shown in Figure 6.8(a). This correlation results from the gray level average value of the second straight layer minus the gray level average value of the first straight

$$\begin{bmatrix} \frac{1}{16} & \frac{1}{16} & \frac{1}{16} & \frac{1}{16} & \frac{1}{16} \\ \frac{1}{16} & \frac{-1}{8} & \frac{-1}{8} & \frac{-1}{8} & \frac{1}{16} \\ \frac{1}{16} & \frac{-1}{8} & 0 & \frac{-1}{8} & \frac{1}{16} \\ \frac{1}{16} & \frac{-1}{8} & \frac{-1}{8} & \frac{-1}{8} & \frac{1}{16} \\ \frac{1}{16} & \frac{1}{16} & \frac{1}{16} & \frac{1}{16} & \frac{1}{16} \end{bmatrix}$$

(a) First feature matrix, $F^{(1)}$

$$\begin{bmatrix} \frac{1}{24} & \frac{1}{24} & \frac{1}{24} & \frac{1}{24} & \frac{1}{24} & \frac{1}{24} & \frac{1}{24} \\ \frac{1}{24} & \frac{-1}{16} & \frac{-1}{16} & \frac{-1}{16} & \frac{-1}{16} & \frac{-1}{16} & \frac{1}{24} \\ \frac{1}{24} & \frac{-1}{16} & 0 & 0 & 0 & \frac{-1}{16} & \frac{1}{24} \\ \frac{1}{24} & \frac{-1}{16} & 0 & 0 & 0 & \frac{-1}{16} & \frac{1}{24} \\ \frac{1}{24} & \frac{-1}{16} & 0 & 0 & 0 & \frac{-1}{16} & \frac{1}{24} \\ \frac{1}{24} & \frac{-1}{16} & \frac{-1}{16} & \frac{-1}{16} & \frac{-1}{16} & \frac{-1}{16} & \frac{1}{24} \\ \frac{1}{24} & \frac{1}{24} & \frac{1}{24} & \frac{1}{24} & \frac{1}{24} & \frac{1}{24} & \frac{1}{24} \end{bmatrix}$$

(b) Second feature matrix, $F^{(2)}$

$$\begin{bmatrix} 0 & 0 & 0 & \frac{1}{12} & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{12} & 0 & \frac{1}{12} & 0 & 0 \\ 0 & \frac{1}{12} & 0 & \frac{-1}{4} & 0 & \frac{1}{12} & 0 \\ \frac{1}{12} & 0 & \frac{-1}{4} & 0 & \frac{-1}{4} & 0 & \frac{1}{12} \\ 0 & \frac{1}{12} & 0 & \frac{-1}{4} & 0 & \frac{1}{12} & 0 \\ 0 & 0 & \frac{1}{12} & 0 & \frac{1}{12} & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{12} & 0 & 0 & 0 \end{bmatrix}$$

(c) Third feature matrix, $F^{(3)}$

$$\begin{bmatrix} 0 & 0 & 0 & 0 & \frac{1}{16} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{16} & 0 & \frac{1}{16} & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{16} & 0 & \frac{-1}{8} & 0 & \frac{1}{16} & 0 & 0 \\ 0 & \frac{1}{16} & 0 & \frac{-1}{8} & 0 & \frac{-1}{8} & 0 & \frac{1}{16} & 0 \\ \frac{1}{16} & 0 & \frac{-1}{8} & 0 & 0 & 0 & \frac{-1}{8} & 0 & \frac{1}{16} \\ 0 & \frac{1}{16} & 0 & \frac{-1}{8} & 0 & \frac{-1}{8} & 0 & \frac{1}{16} & 0 \\ 0 & 0 & \frac{1}{16} & 0 & \frac{-1}{8} & 0 & \frac{1}{16} & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{16} & 0 & \frac{1}{16} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{16} & 0 & 0 & 0 & 0 \end{bmatrix}$$

(d) Fourth feature matrix, $F^{(4)}$

Figure 6.8: Corresponding matrices of features

layer. In other words, this feature measures the average amount of change in the straight layer in vicinity of the core of porosity. Equation (6.9) gives the first feature value of the $\acute{k}_n$ porosity candidate in the n^{th} image.

$$V^{(1)}_{\acute{k}_n,n} = \sum_{i=-2}^{2} \sum_{j=-2}^{2} (F^{(1)}_{i+3,j+3}.B^{(7)}_{p^{(6)}_{1.\acute{k}_n,n}+i,P^{(6)}_{2.\acute{k}_n,n}+j,n}) \quad \forall \acute{k}_n, n \tag{6.9}$$

2. The second feature value is the correlation of pore candidate spot with $F^{(2)}$ matrix shown in Figure 6.8(b). This correlation results from the gray level average value of the third straight layer minus the gray level average value of the second straight

layer. In other words, this feature measures the average amount of change in the straight layer farther from the core of porosity. Equation (6.10) gives the second feature value of the $\acute{k}_n$ porosity candidate in the n^{th} image.

$$V^{(2)}_{\acute{k}_n,n} = \sum_{i=-3}^{3} \sum_{j=-3}^{3} (F^{(2)}_{i+4,j+4}.B^{(7)}_{P^{(6)}_{1,\acute{k}_n,n}+i,P^{(6)}_{2,\acute{k}_n,n}+j,n}) \quad \forall \acute{k}_n,n \tag{6.10}$$

3. The third feature value is the correlation of pore candidate spot with $F^{(3)}$ matrix shown in Figure 6.8(c). This correlation results from the gray level average value of the third diagonal layer minus the gray level average value of the first diagonal layer. In other words, this feature measures the average amount of change in the diagonal layer in vicinity of the core of porosity. Equation (6.11) gives the third feature value of the $\acute{k}_n$ porosity candidate in the n^{th} image.

$$V^{(3)}_{\acute{k}_n,n} = \sum_{i=-3}^{3} \sum_{j=-3}^{3} (F^{(3)}_{i+4,j+4}.B^{(7)}_{P^{(6)}_{1,\acute{k}_n,n}+i,P^{(6)}_{2,\acute{k}_n,n}+j,n}) \quad \forall \acute{k}_n,n \tag{6.11}$$

4. The fourth feature value is the correlation of pore candidate spot with $F^{(4)}$ matrix shown in Figure 6.8(d). This correlation results from the gray level average value of the fourth diagonal layer minus the gray level average value of the second diagonal layer. In other words, this feature measures the average amount of change in the diagonal layer distant from the core of porosity. Equation (6.12) gives the fourth feature value of the $\acute{k}_n$ porosity candidate in the n^{th} image.

$$V^{(4)}_{\acute{k}_n,n} = \sum_{i=-4}^{4} \sum_{j=-4}^{4} (F^{(4)}_{i+5,j+5}.B^{(7)}_{P^{(6)}_{1,\acute{k}_n,n}+i,P^{(6)}_{2,\acute{k}_n,n}+j,n}) \quad \forall \acute{k}_n,n \tag{6.12}$$

5. The fifth feature value is the minimum correlation of pore candidate spot with the 8 matrices shown in Figures 6.6 and 6.7. The value of this feature helps us to distinguish porosity spots from the tool marks: the greater the value of the feature, the slimmer the chance of being a tool mark. The largeness of this value guarantees a significant difference between porosity spot and its vicinity. Equation (6.13) gives the fifth feature value of the $\acute{k}_n$ porosity candidate in the n^{th} image (see Section 6.4).

$$V^{(5)}_{\acute{k}_n,n} = Min_{q=1,\ldots,4}\{S^{(q)}_{\acute{k}_n,n}, D^{(q)}_{\acute{k}_n,n}\} \quad \forall \acute{k}_n, n \tag{6.13}$$

6.6. Decision Making

In this section, we describe fuzzy c-means clustering, the way it has been adopted in our work and its corresponding experimental results.

6.6.1. Fuzzy C-Means Clustering

Since fuzzy C-means clustering was explained in details in Chapter 5, please refer to Section 5.7.1.

6.6.2. Pore Candidate Clustering and Experimental Results

In order to decide whether a pore candidate is a pore or a tool mark, we need to determine the cluster centers. Pore candidates are classified into three different clusters as follows:

- First cluster: strong porosity (defect part)
- Second cluster: weak porosity
- Third cluster: no porosity (non defect part)

Table 6.1: Cluster centers

	First cluster	Second cluster	Third cluster
First feature	0.8156	0.2866	0.0946
Second feature	0.8745	0.3304	0.0883
Third feature	0.8225	0.2953	0.1015
Fourth feature	0.7922	0.2905	0.0542
Fifth feature	0.8135	0.1766	0.0682

The first cluster includes those pixels that can strongly be considered as the core of porosity spot (defect part). The third cluster includes those pixels that can not be considered as the core of porosity (non defect part). We define the second cluster as a cluster which lies between the first and the third cluster. In other words, we may have one or more weak pores in one image. If we are strict about the porosity, this cluster can also be considered as a defect part, the same as the first cluster. In this chapter, we consider the second cluster as a defect cluster to make the model more strict on pore defects.

One thousand points from five different images are used in training to find the aforementioned cluster centers. Using $\mathbf{V}_{\acute{k}_n,n}$ vectors in Section 6.5 and the methodology described in Section 5.7.1, we find the cluster centers after $s = 58$ iterations ($\varepsilon = \frac{1}{100000}$). Figure 6.9 is one of the five images, with 183 pore candidates that has been used to find the cluster centers. The pore candidates have been circled in black.

Table 6.1 shows the cluster center values. The corresponding fuzzy c-means clustering objective function is shown in Figure 6.10. The next task is to measure the Euclidean distance between each point and the cluster centers. Each point (vector) falls into the category (cluster) from which it has the minimum Euclidean distance.

Consequently, 105 images have been tested with 422 pores with the following results: 394 pores have been recognized (93.36%), 28 pores have not been recognized (false

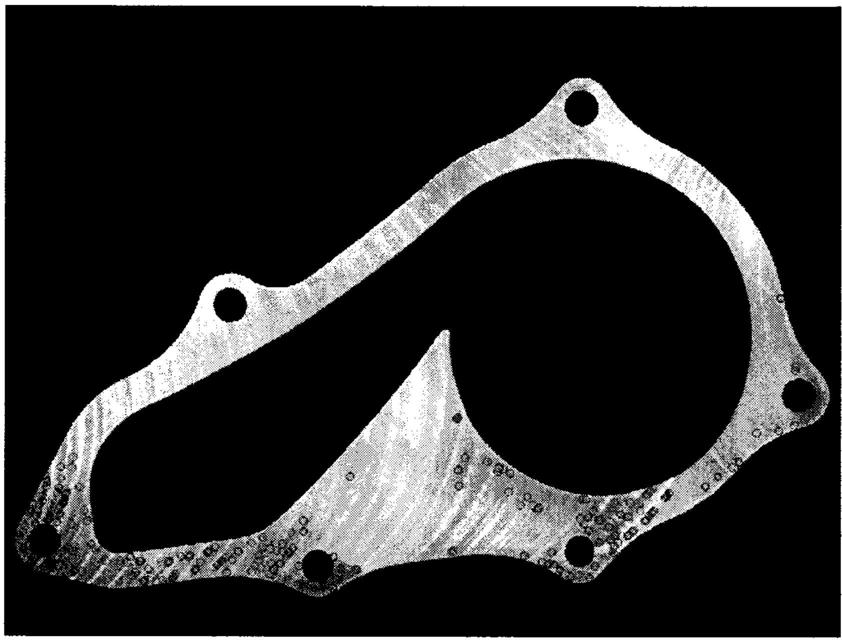


Figure 6.9: Pore candidates

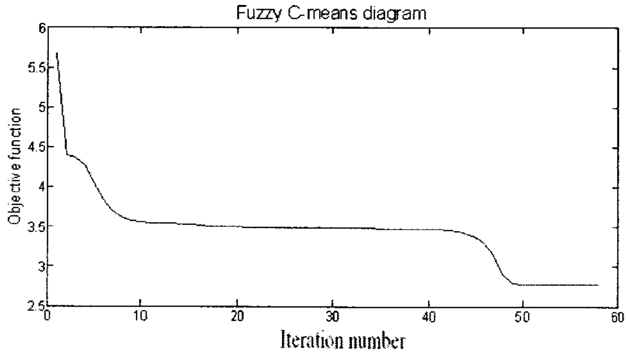


Figure 6.10: Fuzzy c-means clustering objective function

Table 6.2: Fuzzy machine vision based porosity detection model performance

Results	Model truth (T)	Model truth (F)
Expert truth (T)	TP = 93.36 %	FN = 6.64 %
Expert truth (F)	FP = 6.67 %	TN = 93.33 %

negative 6.64%) and 7 tool marks have been mistakenly recognized as pores (false positive 6.67%). Table 6.2 shows pores chart used to score the fuzzy recognizer performance against the truth field defined by the human expert. The TP represents true positive results, FN is false negative results, FP is false positive results and TN is true negative results. The processing time per image was about 200 seconds on a Pentium 4 (3 GHz), 1 GB RAM and a Windows XP platform (90 seconds for pre-processing and 110 seconds for porosity detection). For detecting huge pores (as large as 100 pixels and more), larger size of windows can be easily inserted into the model, however, pores of such size are very few in the entire database. Figures 14 through 19 depict six selected images from our database which present the performance of the proposed method.

6.7. Conclusions

The current research outlines experience with one industrial example which involves the porosity inspection of die cast aluminum automotive water-pump housings. In this research, we have presented an objective fuzzy approach for fast and accurate porosity vision based inspection. The proposed method not only detects any pore defects as long as they break the homogeneity of textures but it also makes it possible to determine the types and classes of porous part defects according to their levels of severity, developed during casting the parts, from aluminum alloy. The proposed method is based on the correlation of the core of pore candidates with twelve developed matrices resulted in five novel features. The fuzzy decision making process on porosity detection adopted and presented in this chapter, adds great value to the whole production system, by increasing the confidence of the inspectors in the machine performing real-time verification. The main complexity of this problem results from the similarity of porosity and tool marks. Although, the relationship between the pore characteristics and the texture attributes of die cast parts is very complex in nature, by using the proposed fuzzy machine vision based method, the pores can be automatically detected. The results demonstrate that 93.36% of the pores in the entire database is correctly identified. It is clear that adopting such system will make automated fuzzy machine vision based porosity detection a more feasible alternative for many companies.

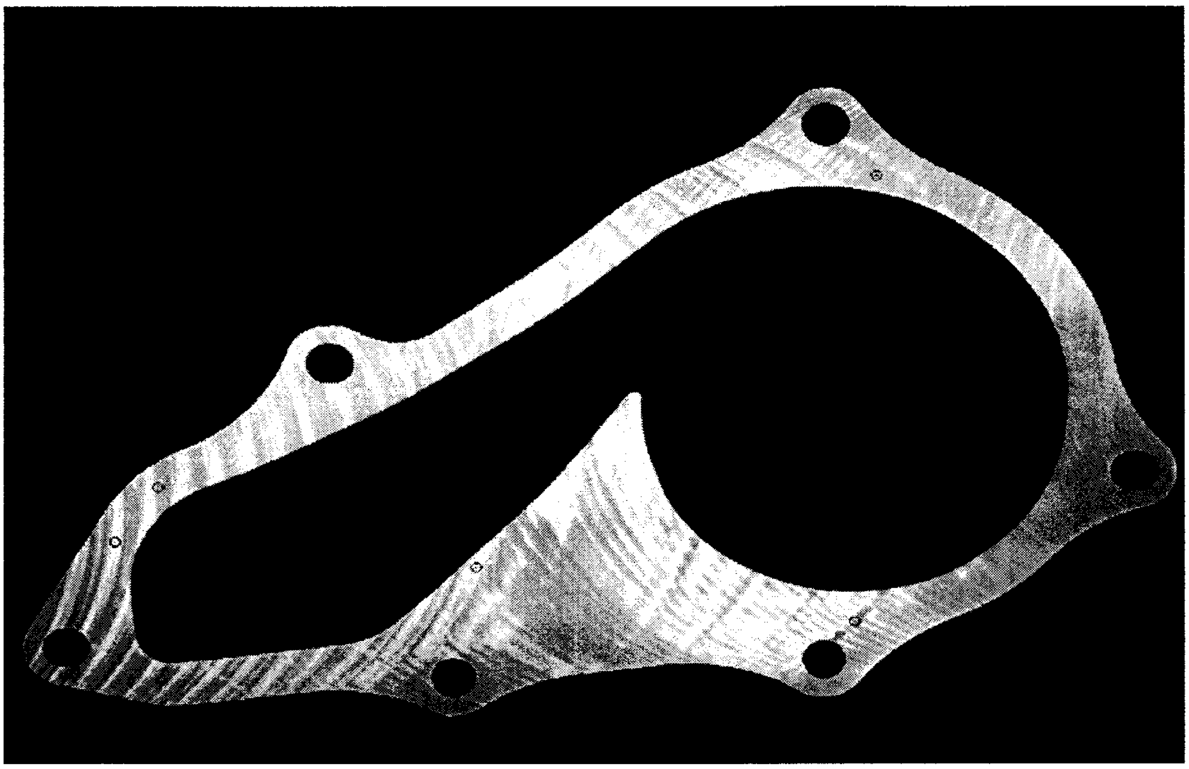


Figure 6.11: An image with several small sized porosity

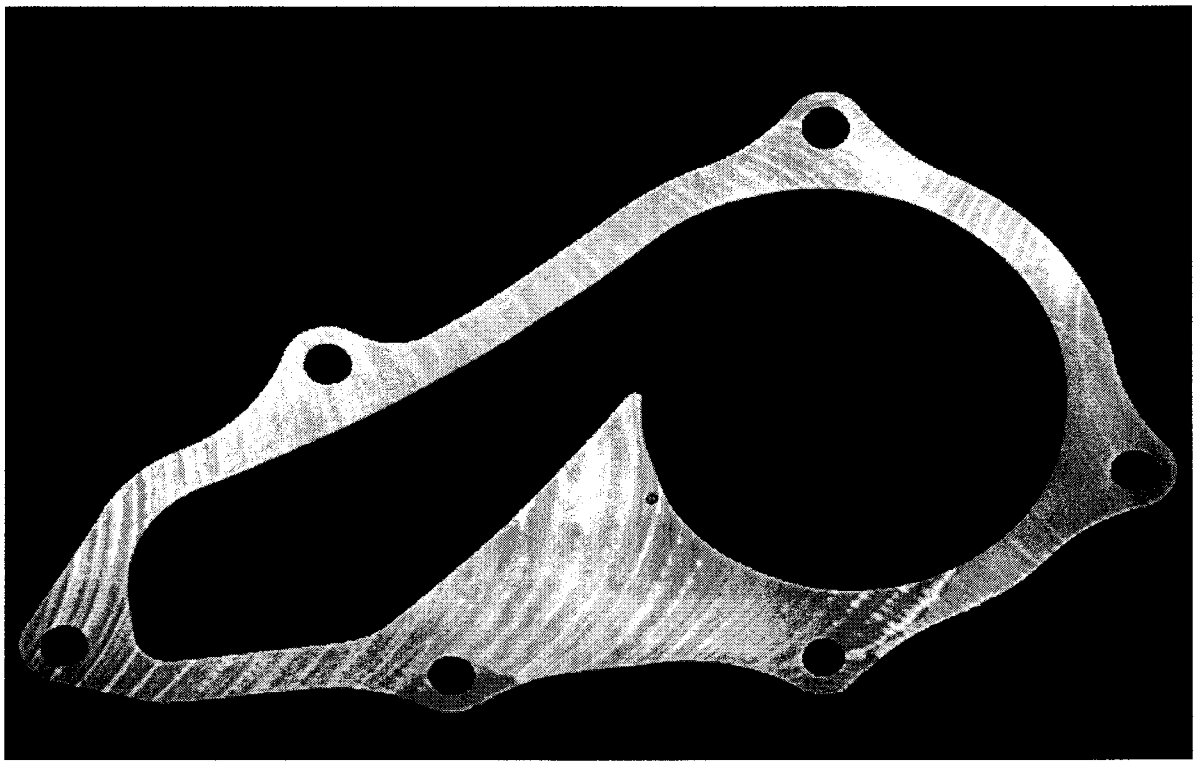


Figure 6.12: An image with two large sized porosity

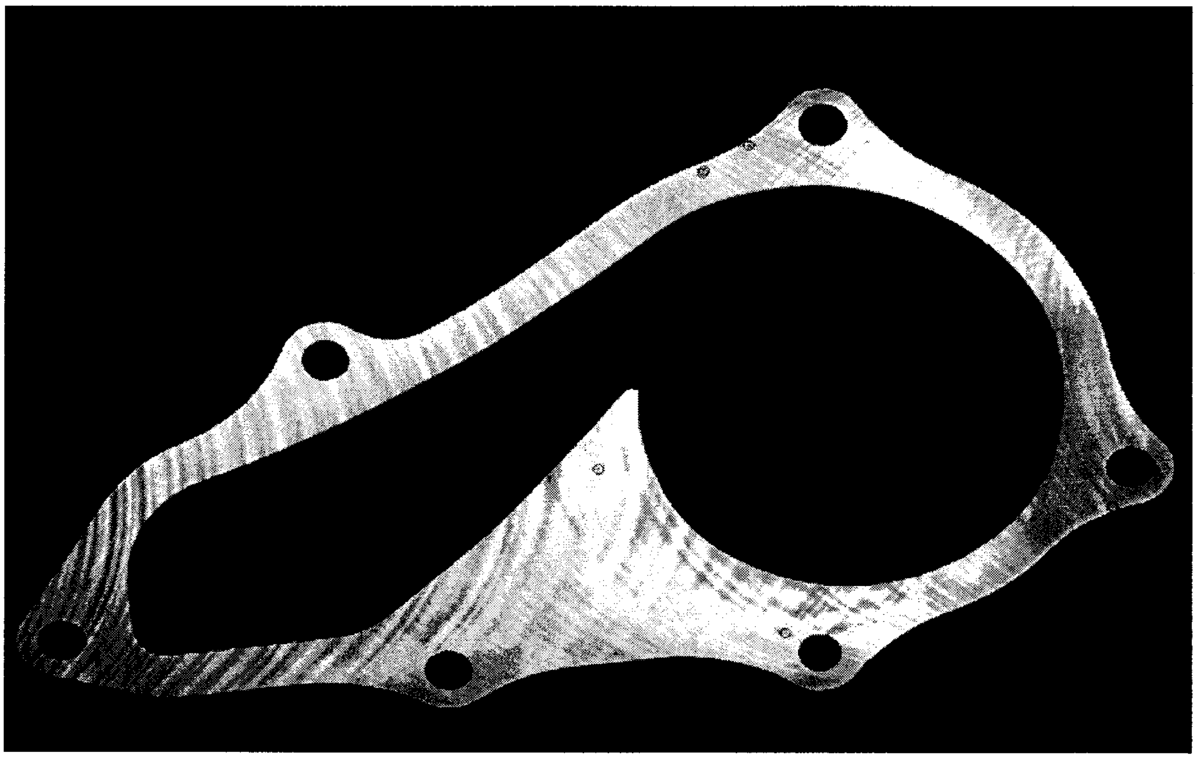


Figure 6.13: Finding small sized porosity among a lot of tool marks

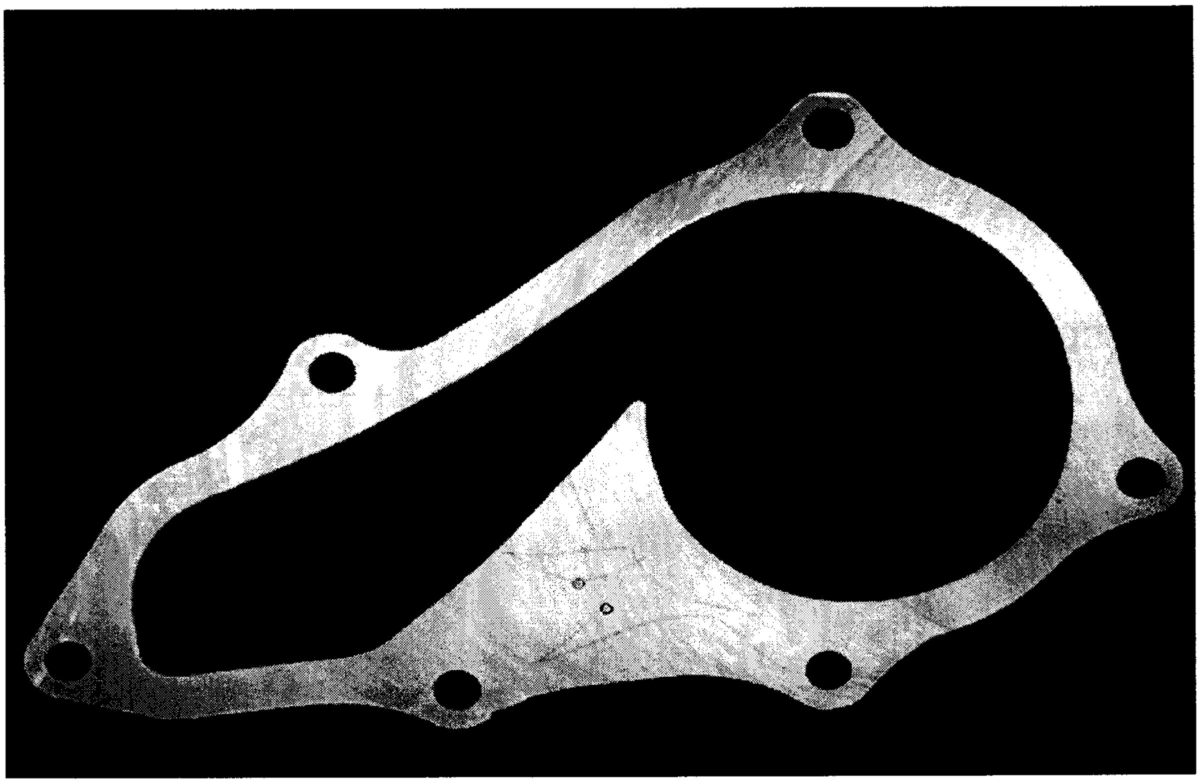


Figure 6.14: While this is a scrathced image, the method finds the pores

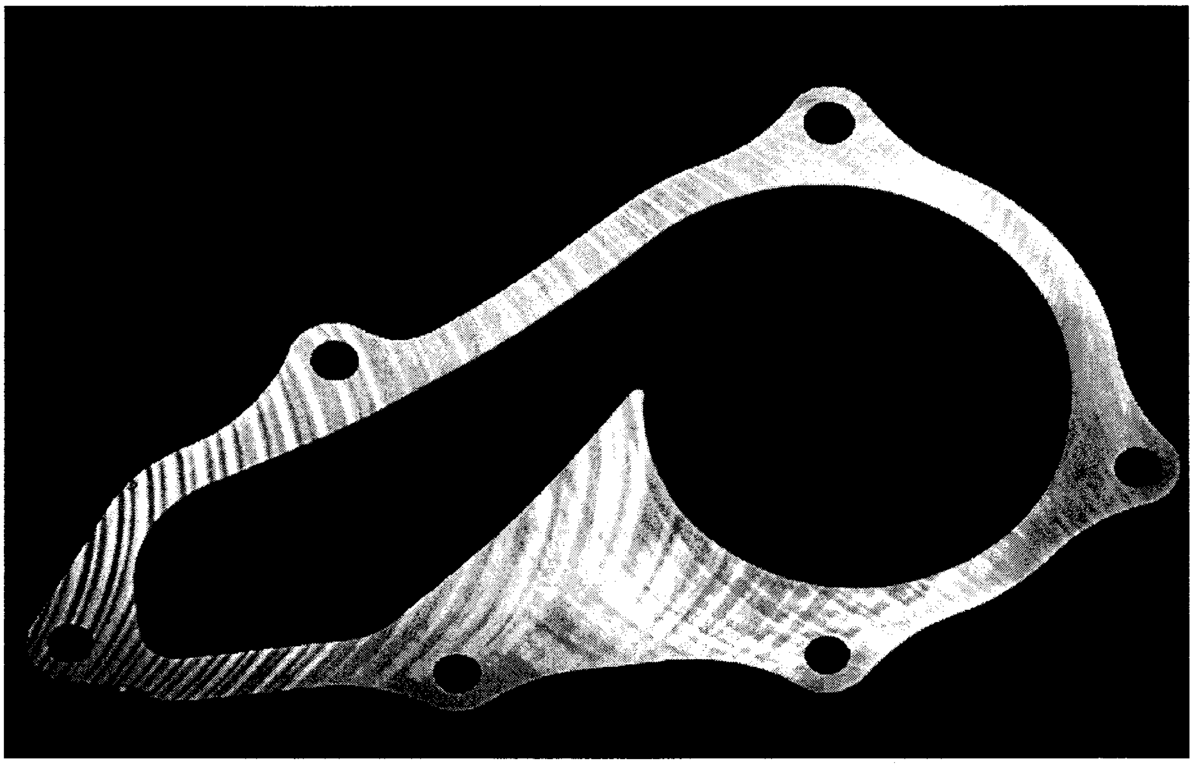


Figure 6.15: It is a false positive image, see top left of the image

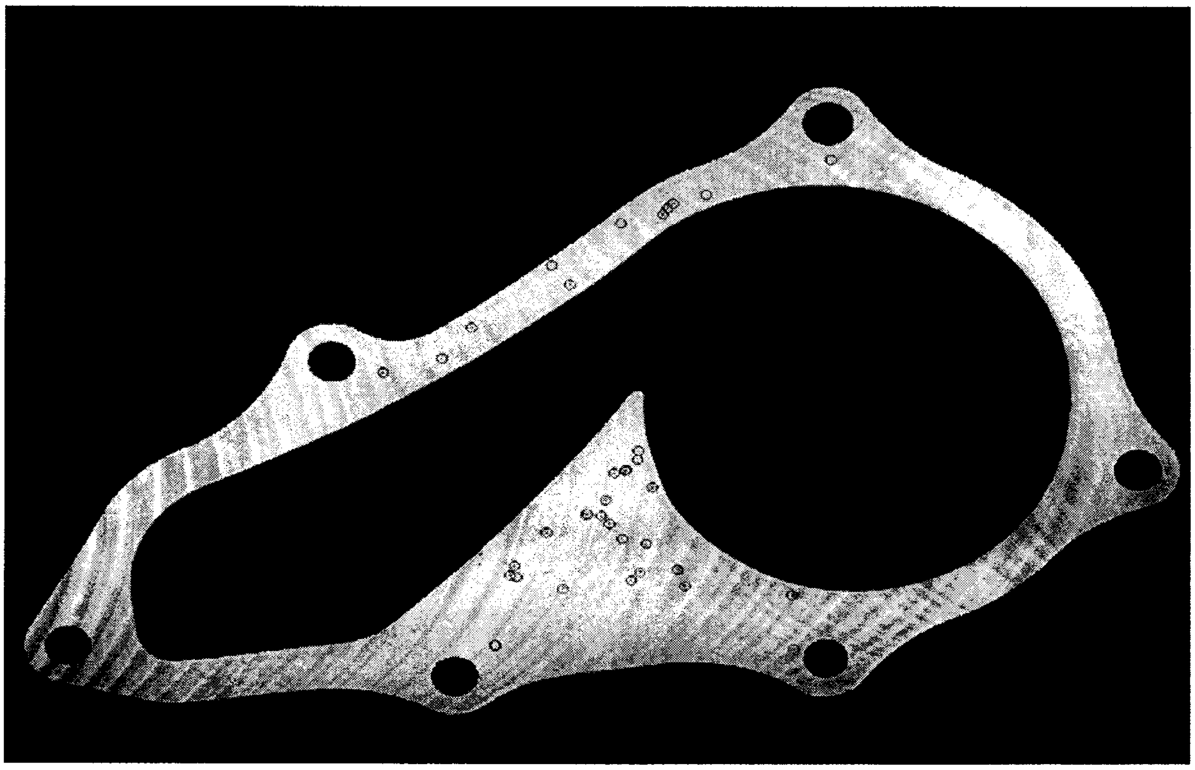


Figure 6.16: An image with many tool marks

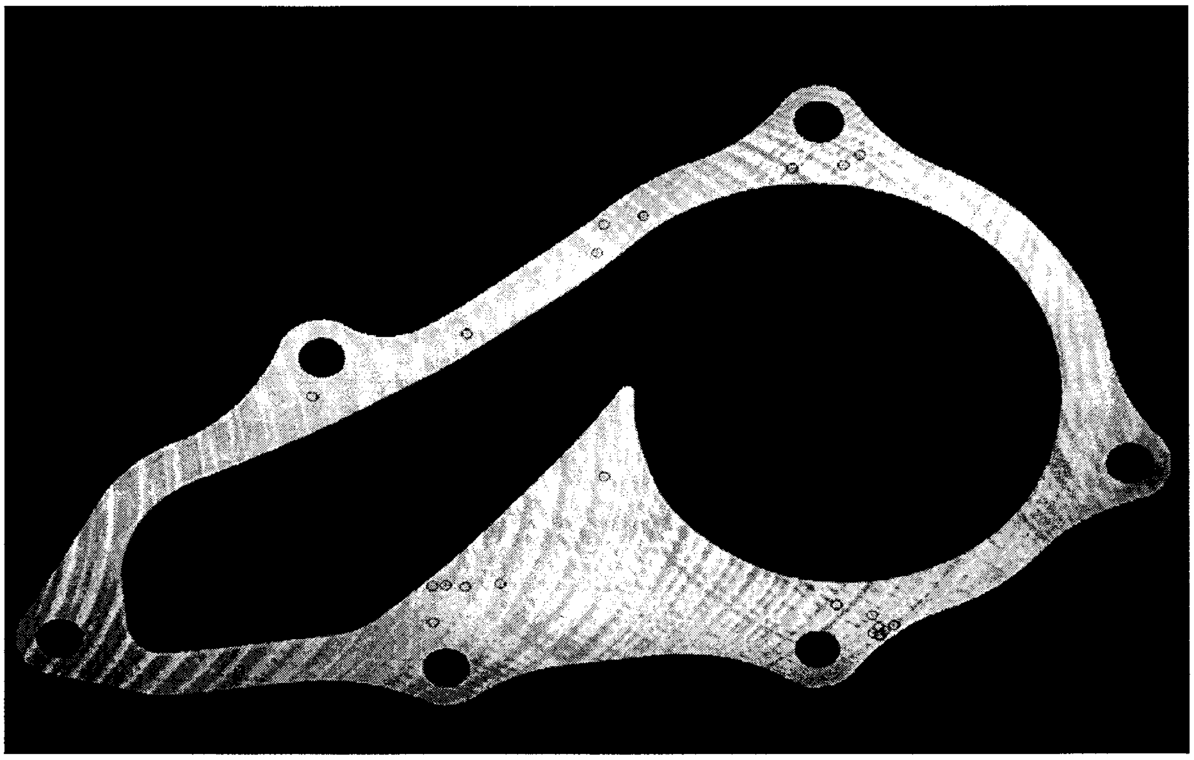


Figure 6.17: The performance of method in a challenging environment

Chapter 7

Fuzzy Wavelet Based Image Classification

7.1. Introduction

Texture normally refers to some characteristics like graininess, coarseness, consistency, and uniformity in the image. While a rigorous definition of texture is not yet available, texture can be defined as a local statistical pattern of primitive elements in observer's domain of interest [Avci and Sengur 2009]. It provides a valuable source of information for interpreting the contents of real world images and hence texture analysis plays an important role in many machine vision and image processing algorithms having applications in robot navigation [Sridharan and Stone 2009], remote sensing [Tamm and Remm 2009], industrial quality control [Maldonado and Grana 2009], content-based image retrieval [He *et al.* 2009], object identification [Li and Meng 2009], shape understanding [Daliri and Torre 2009], document analysis [Abraham *et al.* 2009], image coding [Yang *et al.* 2009], medical image analysis [Iscan *et al.* 2009], individual's biometric authentication [Cetingl *et al.* 2006] and etc. The aim of texture classification is to assign the best matching category to a given texture. Therefore, it is essential to obtain quantitative signatures and descriptors of textures in order to be able to use the mathematical tools.

Statistical methods based on co-occurrence matrix descriptors, gray level-run-length and gray level differences are perhaps the first valuable effort in this area [Haralick *et al.* 1973, Galloway 1975].

Fourier transform and Gabor transform are the two texture analysis techniques used in the frequency domain. Fourier power spectrum was presented in [Weszka *et al.* 1976]. Gabor features, Markov random fields (MRF) based features and fractal features were compared in literature [Chen and Chen 1999]. Features extracted from wavelet decompositions have also been used to develop models of attributes [Pal *et al.* 2008].

In this research, wide variety of tool marks, such as groove marks, prod marks and skip marks are formed during machining operations on the surface of water pumps. Hence, based on the type and the number of tool marks three different classes of images are defined. A texture image classification scheme is proposed which uses wavelet transform and adaptive neural-fuzzy inference system (*ANFIS*) classifier. As a result, this work deals with the tool mark images as a source of different textures for industrial water pump classification.

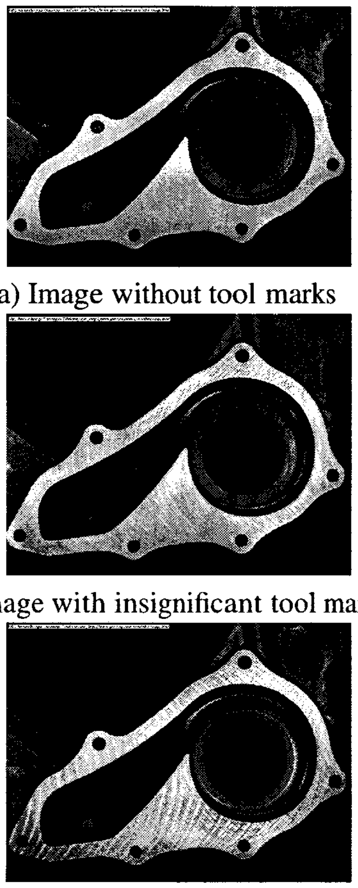
The remaining part of this chapter is organized as follows. Section 2 is devoted to the nature of the problem at hand. In Section 3, we explain the pre-processing process. The methodology and the feature extraction process is given in Section 4. In Section 5, the developed ANFIS classification model endorsed with experimental results has been discussed. Finally, we draw the conclusions in Section 6.

7.2. Problem Definition

This research basically involves the inspection of die cast aluminum automotive pump housing. The water pump has several machined surfaces. Since its front surface is a sealing surface, any visible tool marks on this surface potentially turns it into a defective part, leading to a leakage.

In our database, we have 105 water-pump images where, 35 of them have no tool marks (class 1), 35 of them have insignificant tool marks (class 2) and the rest have significant tool marks (class 3). All the initial images have been captured in standard Tagged Image File Format (Tiff). They are all gray level images with a size of 960 by 1280. Figure 7.1(a) shows a generic defect-free image with no considerable tool marks (class 1) whereas, Figure 7.1(b) illustrates a typical image with insignificant number of tool marks (class 2), meanwhile Figure 7.1(c) shows an image with a lot of tool marks (class 3). For a better comparison among class 1, class 2 and class 3, Figures 7.2, 7.3 and 7.4 display each of these classes with four samples, respectively. However, when it comes to images of classes 2 and 3, a large number of varieties exists, depending on the location, number, shape and severity of tool marks. For example, Figure 7.4 only shows four highly different samples of numerous possibilities of images of class 3.

As it is seen in Figure 7.4, the main complexity of image classification based on tool mark problem results from the numerous varieties of tool marks. It should be noted that texture detection is extremely sensitive to noise. This arises a defying situation as we are dealing with rapid changes in lighting conditions from one image to another as well as swift changes in light conditions from one area of image to another. More importantly, even the texture of a single image varies and can not be considered constant, e.g. see Figure 7.4(d). Therefore, making a distinction among different classes of tool marks could be a very challenging task.



(a) Image without tool marks

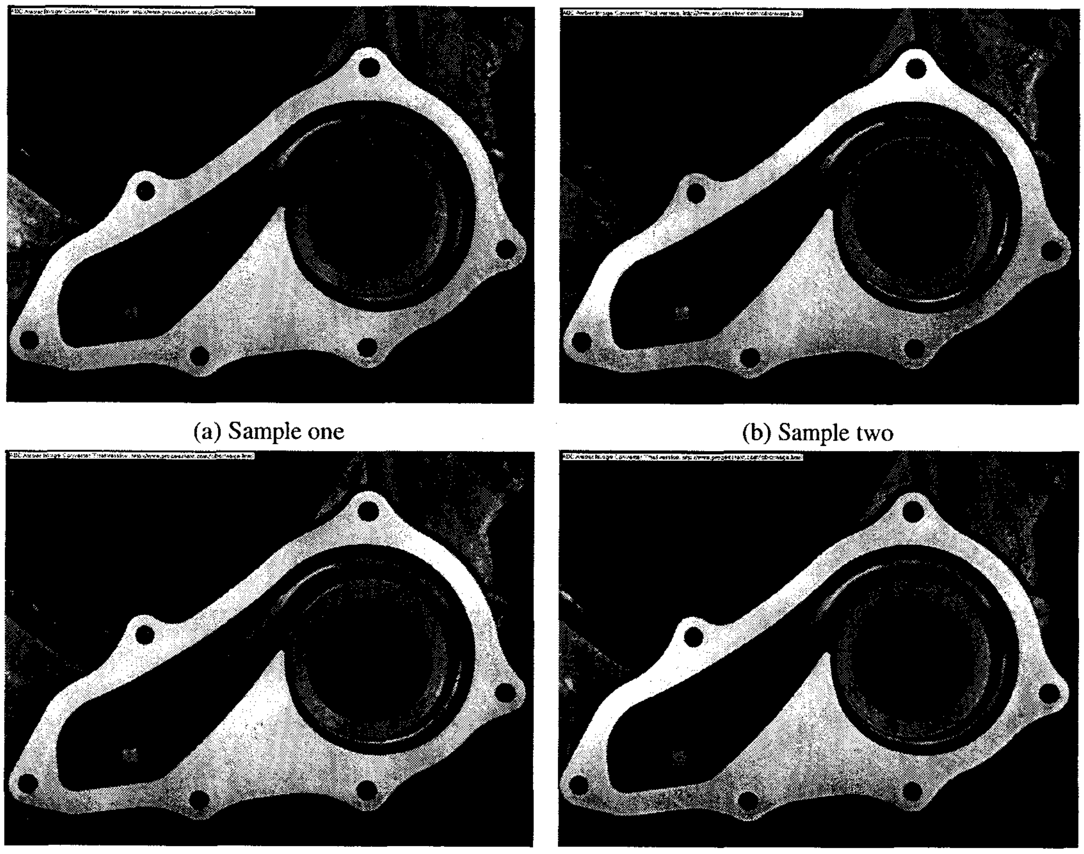
(b) Image with insignificant tool marks

(c) Image with significant tool marks

Figure 7.1: Three classes of images

In addition to tool mark detection problem, reflection is another specific problem of machined metallic surfaces. This, in turn, introduces variations in the appearance of defect-free surfaces as well as in the appearance of the tool mark surfaces. In this work we approach to tool mark classification problem as a clustering problem and refer to neuro-fuzzy method for decision making.

Using required pre-processing steps, the main objective of this work is the development of an efficient fuzzy machine vision based method to correctly identify and classify parts with tool marks. As it is shown in Figure 7.5, we introduce the steps that should be



(a) Sample one

(b) Sample two

(c) Sample three

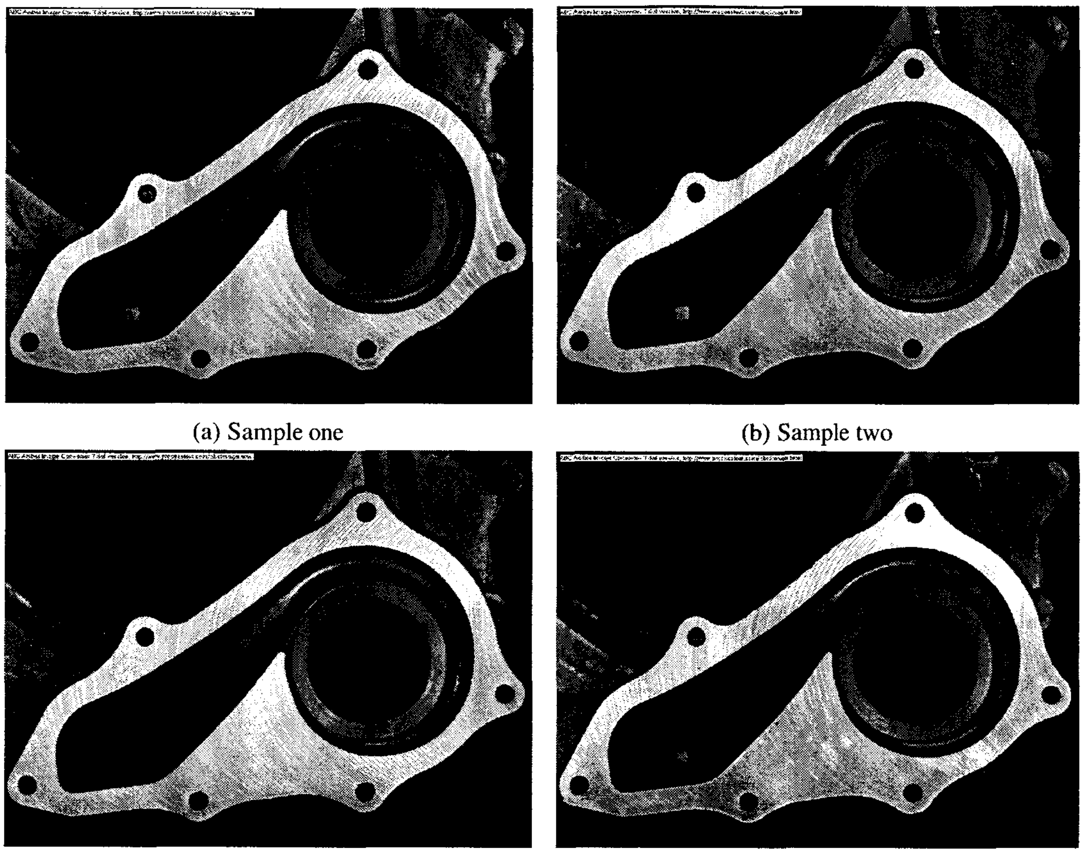
(d) Sample four

Figure 7.2: Four samples of images of class 1

taken in order to reach the desirable results. Our machine vision based system has four main steps as follows:

1. *Image acquisition*: The acquired digital image is quantized in both the spatial coordinates and levels. Initial images in our database are the outputs of this step.

2. *Pre-processing*: Image pre-processing is carried out on the image to align images for getting a consolidated scene, such as noise filtering, image segmentation and the application of horizontal, vertical and orientation corrections. We illustrate this part using industrial water pump images in Section 3.



(a) Sample one

(b) Sample two

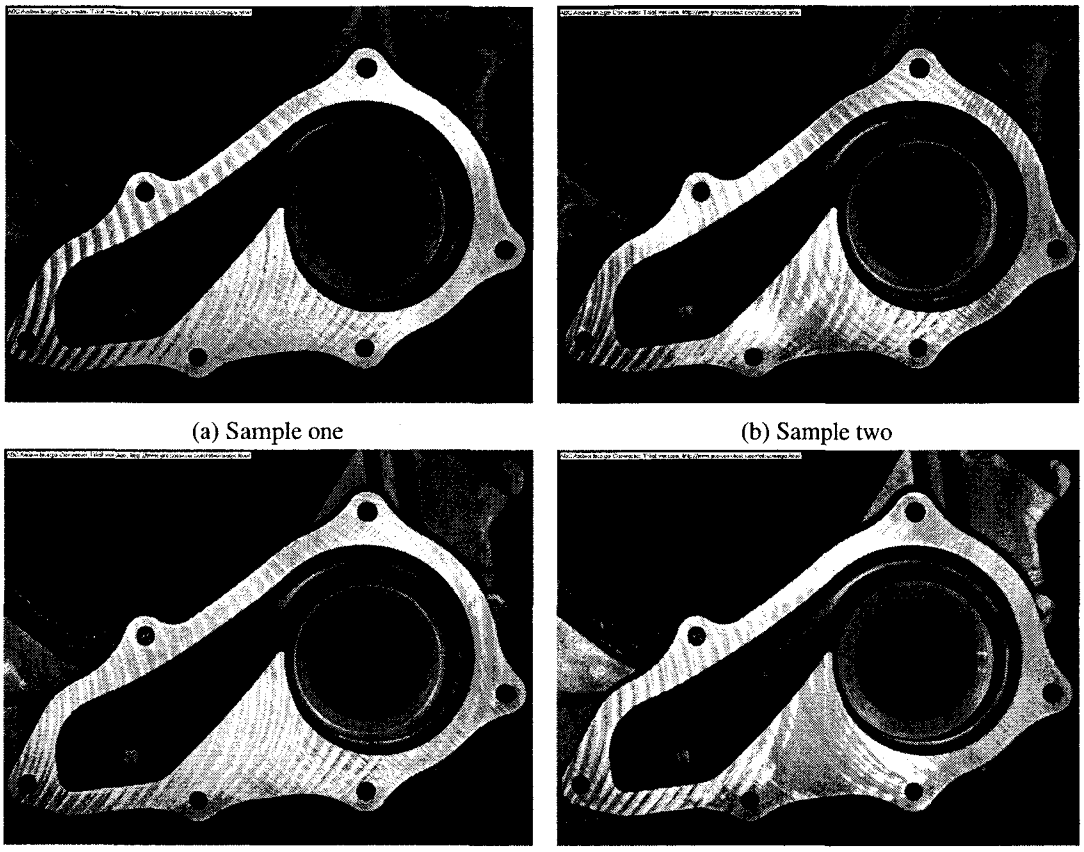
(c) Sample three

(d) Sample four

Figure 7.3: Four samples of images of class 2

3. *Feature extraction*: In this step, the properties of each image are translated to vectors. In this research, they are translated to wavelet and statistical attributes presented by feature vectors. Such representations facilitate processing and statistical analysis. Section 4 describes our proposed features for part classification based on tool mark.

4. *Recognition*: ANFIS classification is made based on feature vectors for decision making. This step is explained in Section 5 with details. As it is earlier mentioned in this section, in the context of this study we classify water pumps into three different



(a) Sample one

(b) Sample two

(c) Sample three

(d) Sample four

Figure 7.4: Four samples of images of class 3

classes.

7.3. Pre-processing

In this section, we introduce the steps that should be taken in order to reach to the desirable image. First, we create a binary image, then clear all the noises and redundant objects in order to detect the main object (water-pump surface). Next, the test image is horizontally and vertically adjusted, based on the reference image, and its location is set (displacement and orientation adjustment). Afterward, the zone of interest for each image

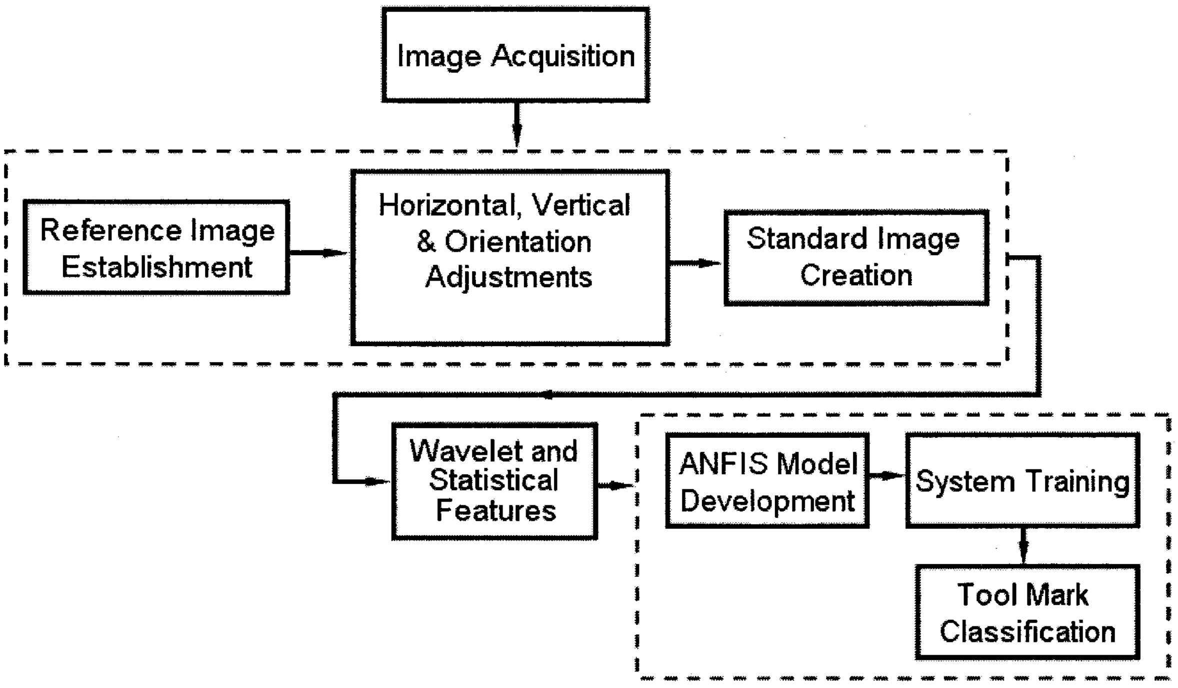
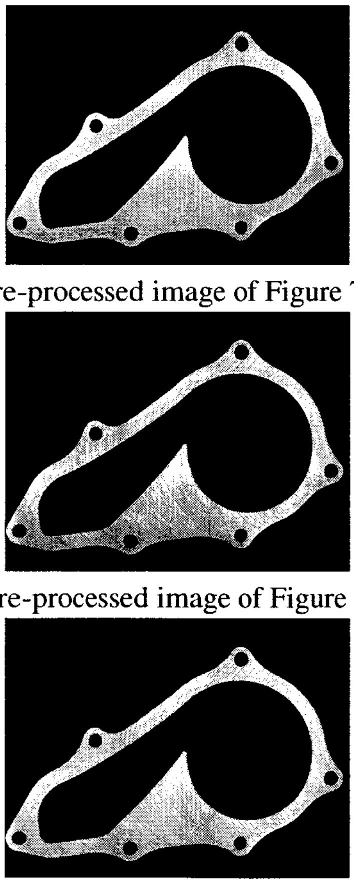


Image Acquisition
Reference Image Establishment
Horizontal, Vertical & Orientation Adjustments
Standard Image Creation
Wavelet and Statistical Features
ANFIS Model Development
System Training
Tool Mark Classification

Figure 7.5: Scheme of the machine visoin based system

is automatically determined. Finally, the zone of interest for each image is transformed to a comparable rectangle to be workable for two dimensional discrete wavelet application. Since similar steps were explained in details in Chapter 5, we don't explain them here. Readers can refer to 5.4.

As a result, Figures 7.6(a), (b) and (c) show the outputs of the pre-processed images of Figure 7.1(a), (b) and (c), respectively. In this step we refer to the pixel in the i^{th} row and in the j^{th} column for the n^{th} pre-processed image as $P^{(n)}(i,j)$. The corresponding matrix is shown with $\mathbf{P}^{(n)}$.



(a) Pre-processed image of Figure 7.1(a)

((b) Pre-processed image of Figure 7.1(b)

(c) Pre-processed image of Figure 7.1(c)

Figure 7.6: Three pre-processed images of three classes

7.4. Methodology

In this section we briefly explain wavelet transform, the proposed statistical and wavelet based features.

7.4.1. Wavelet Transform

The wavelet transform has a huge number of applications in science, engineering, mathematics and image processing. The main advantage of wavelet over the traditional methods such as Fourier transform is that it has a varying window size, being wide for low frequencies and narrow for the high ones. Moreover, if a signal is not stationary, Fourier transform can not specify where in time a certain component of frequency appears but wavelet transform can. The continuous wavelet transform of a 1-D signal $f(t)$ is defined in Equation (7.1).

$$CWT_f^{\psi}(\tau,s) = \Psi_f^{\psi} = \frac{1}{\sqrt{|s|}} \int f(t)\psi^*(\tfrac{t-\tau}{s})dt \tag{7.1}$$

The transformed signal is a function of two variables, τ and s, the translation and scale parameters, respectively. ψ is the transforming function, and it is called the mother wavelet. The extension to the 2-D case can be performed by using a product of 1-D filters. In practice, the transform is computed by applying a separable filter band to the image using Equations (7.2) to (7.5).

$$A_j(\mathbf{t}) = [H_x * [H_y * A_{j-1}]_{\downarrow 2,1}]_{\downarrow 1,2}(\mathbf{t}) \tag{7.2}$$

$$D_j^{(h)}(\mathbf{t}) = [H_x * [G_y * A_{j-1}]_{\downarrow 2,1}]_{\downarrow 1,2}(\mathbf{t}) \tag{7.3}$$

$$D_j^{(v)}(\mathbf{t}) = [G_x * [H_y * A_{j-1}]_{\downarrow 2,1}]_{\downarrow 1,2}(\mathbf{t}) \tag{7.4}$$

$$D_j^{(d)}(\mathbf{t}) = [G_x * [G_y * A_{j-1}]_{\downarrow 2,1}]_{\downarrow 1,2}(\mathbf{t}) \tag{7.5}$$

H and G are low-pass and band-pass filters. $\downarrow 2,1$ and $\downarrow 1,2$ are down-sampling along the rows and columns, respectively. $\mathbf{t} = (t_1, t_2) \in R^2$, $*$ shows the convolution operator and $A_0 = I(\mathbf{x})$ is the original image. In two dimensional wavelet analysis, a signal is split into an approximation A, a horizontal detail $D^{(h)}$, a vertical detail $D^{(v)}$ and a diagonal detail $D^{(d)}$. The approximation is then itself split into a second-level approximation and details, and the process is repeated. A_j is obtained by low-pass filtering at scale j, thus it is called low-resolution image at scale j. Details are obtained by band-pass filtering at scale j. A multiscale representation of original image I of depth p is done using a set of subimages at several scales $\{A_p, D_j^{(h)}, D_j^{(v)}, D_j^{(d)}\}^{j=0,\ldots,p-1}$. In this work we use a depth of $p = 3$ for subimages. Figure 7.7 displays such representation. However, it is not the intent of this research to provide the details of the wavelet theory. To obtain more background on the wavelet techniques, the reader is referred to [Chui 1992].

7.4.2. Feature Extraction

Feature extraction is carried out by mapping of a multidimensional space into a space of fewer dimensions. If the extracted features are deliberately selected, it is expected that the feature set extracts the relevant information from the input data to perform the desired task using this reduced representation instead of the full size input. In this part we construct five features to make a distinction among variations of tool marks on images. The best way to present the features is a vector. Hence, a vector corresponding to each image is

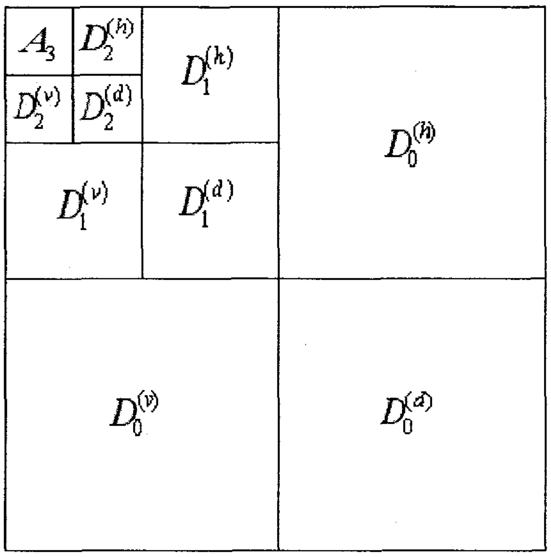


Figure 7.7: A multiscale representation of original image of a depth of $p = 3$

introduced. Equation (7.6) shows the five features of the n^{th} image for the aforementioned vector. The five features that make up the $\mathbf{V}_n$ are explained in detail below.

$$\mathbf{V}_n = \begin{bmatrix} v_n^{(1)} & v_n^{(2)} & v_n^{(3)} & v_n^{(4)} & v_n^{(5)} \end{bmatrix} \tag{7.6}$$

1. As we earlier described the construction of pre-processed image in Section 5.4.3, let $\mathbf{P}^{(n)}$ as the corresponding matrix for the n^{th} pre-processed image. We break $\mathbf{P}^{(n)}$ to smaller 8 by 8 sub-matrices ($P_{k_1,k_2}^{(n)}$), where $k_1 = 1,2...120$ and $k_2 = 1,2...160$, as shown in Equation (7.7).

$$\mathbf{P}^{(n)} = \begin{bmatrix} P_{1,1}^{(n)} & P_{1,2}^{(n)} & \cdots & P_{1,160}^{(n)} \\ P_{2,1}^{(n)} & P_{2,2}^{(n)} & \cdots & P_{2,160}^{(n)} \\ \vdots & \vdots & \vdots & \vdots \\ P_{120,1}^{(n)} & P_{120,2}^{(n)} & \cdots & P_{120,160}^{(n)} \end{bmatrix} \tag{7.7}$$

The first feature measures the summation of range in all the sub-matrices. We define $R^{(n)}_{k_1,k_2}$ as the range in the sub-matrix in the k_1^{th} row and in the k_2^{th} column for the n^{th} pre-processed image. $R^{(n)}_{k_1,k_2}$ is calculated in Equation (7.8).

$$R^{(n)}_{k_1,k_2} = max_i(max_j(P^{(n)}_{k_1,k_2}(i,j))) - min_i(min_j(P^{(n)}_{k_1,k_2}(i,j))) \quad \forall\, k_1,k_2 \qquad (7.8)$$

where, $i = 1,2,...,8$ and $j = 1,2,...,8$. The summation of the ranges which composes the first feature is shown in Equation (7.9). We expect pre-processed images with many tool marks render larger value than the ones with no or few tool marks.

$$v^{(1)}_n = \sum_{k_1=1}^{120} \sum_{k_2=1}^{160} R^{(n)}_{k_1,k_2} \qquad (7.9)$$

2. The second feature measures the variation of the average sub-matrices in each pre-processed image. It is hypothesized that if in a certain pre-processed image the standard variation of the average sub-matrices gets a large value, that image probably belongs to the third category of images (images with many tool marks). If the standard variation of the average sub-matrices has a small value that image probably comes from the first category of images (images with no tool marks). $\bar{P}^{(n)}_{k_1,k_2}$ is defined as the average value of the sub-matrix in the k_1^{th} row and in the k_2^{th} column for the n^{th} pre-processed image. It is calculated in Equation (7.10).

$$\bar{P}^{(n)}_{k_1,k_2} = \frac{\sum_{i=1}^{8} \sum_{j=1}^{8} P^{(n)}_{k_1,k_2}(i,j)}{64} \quad \forall\, k_1,k_2 \qquad (7.10)$$

We represent the corresponding average value of the sub-matrices for the n^{th} pre-processed image by $\bar{\mathbf{P}}^{(n)}$ using the following matrix.

$$\bar{\mathbf{P}}^{(n)} = \begin{bmatrix} \bar{P}^{(n)}_{1,1} & \bar{P}^{(n)}_{1,2} & \cdots & \bar{P}^{(n)}_{1,160} \\ \bar{P}^{(n)}_{2,1} & \bar{P}^{(n)}_{2,2} & \cdots & \bar{P}^{(n)}_{2,160} \\ \vdots & \vdots & \vdots & \vdots \\ \bar{P}^{(n)}_{120,1} & \bar{P}^{(n)}_{120,2} & \cdots & \bar{P}^{(n)}_{120,160} \end{bmatrix} \tag{7.11}$$

The mean value of all the sub-matrices average values for the n^{th} pre-processed image is presented by $\mu^{(n)}$ and is calculated in Equation (7.12).

$$\mu^{(n)} = \sum_{k_1=1}^{120} \sum_{k_2=1}^{160} \frac{\bar{P}^{(n)}_{k_1,k_2}}{120 \times 160} \tag{7.12}$$

Finally, the standard deviation of the $\bar{\mathbf{P}}^{(n)}$ which is the second feature, is calculated using Equation (7.13).

$$v_n^{(2)} = \sqrt{\sum_{k_1=1}^{120} \sum_{k_2=1}^{160} \frac{(\bar{P}^{(n)}_{k_1,k_2} - \mu^{(n)})^2}{120 \times 160}} \tag{7.13}$$

3. The third feature is the summation of the largest values in variance image for all the sub-matrices. A pixel in the variance image becomes the variance value for a small window around the corresponding pixel in the original image [Davies 2005b]. However, the window to generate variance image should have an odd size. In this application, we adopt 7 by 7 window to generate the variance image. Choosing the proper size of the window is entirely application depended and finding the optimum size of the window is beyond the scope of this study. More research can be done to investigate the effect of the window with different sizes on the final result. To make the variance image the same size as the original image, we use zero-padding technique which assumes that the signal is zero outside the original support. Therefore,

we make sure that the variance image is exactly the same size as pre-processed image. The variance image for the n^{th} pre-processed image is denoted by $\mathbf{W}^{(n)}$ and is calculated using Equation (7.14). $\mathbf{Z}^{(n)}$ represents applying zero-padding technique on the n^{th} variance image.

$$W^{(n)}(i,j) = \textstyle\sum_{i=-3}^{i=3} \sum_{j=-3}^{j=3} (P^{(n)}(i,j) - \frac{\sum_{i=-3}^{i=3} \sum_{j=-3}^{j=3} P^{(n)}(i,j)}{49})^2 \tag{7.14}$$

However, to be consistent with sub-matrices introduced earlier for the first two features, we again use the same 8 by 8 window as the size of sub-matrices on the zero-padding variance image. We break $\mathbf{Z}^{(n)}$ into smaller 8 by 8 sub-matrices ($Z^{(n)}_{k_1,k_2}$), where $k_1 = 1,2...120$ and $k_2 = 1,2...160$ as shown in Equation (7.15).

$$\mathbf{Z}^{(n)} = \begin{bmatrix} Z^{(n)}_{1,1} & Z^{(n)}_{1,2} & \cdots & Z^{(n)}_{1,160} \\ Z^{(n)}_{2,1} & Z^{(n)}_{2,2} & \cdots & Z^{(n)}_{2,160} \\ \vdots & \vdots & \vdots & \vdots \\ Z^{(n)}_{120,1} & Z^{(n)}_{120,2} & \cdots & Z^{(n)}_{120,160} \end{bmatrix} \tag{7.15}$$

We define $M^{(n)}_{k_1,k_2}$ as the maximum value in the sub-matrix in the k_1^{th} row and in the k_2^{th} column for the n^{th} pre-processed image. $M^{(n)}_{k_1,k_2}$ is calculated as shown in Equation (7.16). It is highly probable this feature increases where it hits a tool mark. As a result, we anticipate that images with many tool marks render a significant value for this feature.

$$M^{(n)}_{k_1.k_2} = max_i(max_j(Z^{(n)}_{k_1,k_2}(i,j))) \quad \forall\, k_1,k_2 \tag{7.16}$$

where, $i = 1,2,...,8$ and $j = 1,2,...,8$.

The summation of the maximum values in each sub-matrices, which composes the third feature, is shown in Equation (7.17).

$$v_n^{(3)} = \sum_{k_1=1}^{120} \sum_{k_2=1}^{160} M_{k_1,k_2}^{(n)} \tag{7.17}$$

4. The fourth feature is the summation of wavelet energy signatures in all three scales $j = 0,1,2$. Since most relevant texture information has been removed by low-pass filtering, the energy of low resolution image (A_p) is not considered. This feature targets the distribution of energy along the frequency axis over scale and amplitude. Equation (7.18) shows how the fourth feature for the n^{th} image is calculated.

$$v_n^{(4)} = \sum_{j=0}^{2} \sum_{t_1=1}^{640\times 2^{-j}} \sum_{t_2=1}^{480\times 2^{-j}} \left(\frac{[D_{nj}^{(h)}(t_1,t_2)]^2}{(480\times 640\times 2^{-2j})} + \frac{[D_{nj}^{(v)}(t_1,t_2)]^2}{(480\times 640\times 2^{-2j})} + \frac{[D_{nj}^{(d)}(t_1,t_2)]^2}{(480\times 640\times 2^{-2j})}\right) \tag{7.18}$$

where, $D_{nj}^{(h)}(t_1,t_2)$, $D_{nj}^{(v)}(t_1,t_2)$ and $D_{nj}^{(d)}(t_1,t_2)$ are the wavelet coefficients at scale j in the n^{th} image for horizontal, vertical and diagonal details, respectively (see Section 7.4).

5. The fifth feature is the summation of the variances of the wavelet coefficients in the first scale $j = 0$. This feature measures the distribution of wavelet coefficients in each image. Equation (7.19) presents how the fifth feature for the n^{th} image is calculated.

$$v_n^{(5)} = \sum_{t_1=1}^{640} \sum_{t_2=1}^{480} \left(\frac{[D_{n0}^{(h)}(t_1,t_2)-E(D_{n0}^{(h)})]^2}{480\times 640} + \frac{[D_{n0}^{(v)}(t_1,t_2)-E(D_{n0}^{(v)})]^2}{480\times 640} + \frac{[D_{n0}^{(d)}(t_1,t_2)-E(D_{n0}^{(d)})]^2}{480\times 640}\right) \tag{7.19}$$

where, $E(D_{n0}^{(h)})$, $E(D_{n0}^{(v)})$ and $E(D_{n0}^{(d)})$ are the expected wavelet corresponding coefficient values (means) at scale 0 in the n^{th} image for horizontal, vertical and diagonal details, respectively.

7.5. Decision Making and Experimental Results

In this section, we describe adaptive neuro-fuzzy inference systems (ANFIS) classifier, the way it has been adopted in our research and corresponding experimental results.

7.5.1. Architecture of Adaptive Network Based Fuzzy Inference System

Neuro-fuzzy refers to combinations of artificial neural networks and fuzzy logic. Neuro-fuzzy system results in a hybrid intelligent system that takes advantage of these two techniques. This combination of fuzzy reasoning and network calculation results in good reasoning in terms of quality and quantity. The ANFIS is a fuzzy Sugeno model placed in the framework of adaptive systems to facilitate learning and adaptation [Jang 1992a]. To present the ANFIS architecture, a fuzzy inference system with m rules, n inputs and one output is considered.

Rule 1: If (x_1 is A_{11}) and (x_2 is A_{12}) ... (x_n is A_{1n}) then ($f_1 = p_{10} + p_{11}x_1 + p_{12}x_2 ... p_{1n}x_n$)

Rule 2: If (x_1 is A_{21}) and (x_2 is A_{22}) ... (x_n is A_{2n}) then ($f_2 = p_{20} + p_{21}x_1 + p_{22}x_2 ... p_{2n}x_n$)

$\vdots$

Rule m: If (x_1 is A_{m1}) and (x_2 is A_{m2}) ... (x_n is A_{mn}) then ($f_m = p_{m0} + p_{m1}x_1 + p_{m2}x_2 ... p_{mn}x_n$)

where, x_j, $(j = 1, 2, ..., n)$ are the inputs, A_{ij}, $(i = 1, 2, ..., m)$ and $(j = 1, 2, ..., n)$ are the fuzzy sets, f_i are the outputs within the fuzzy domain specified by the fuzzy rule, p_{ij}, $(i = 1, 2, ..., m)$ and $(j = 0, 1, ..., n)$ are the consequent parameters that are determined during the training process. The ANFIS architecture to implement these m rules is shown in Figure 7.8.

1. Layer 1: Every node ij in this layer is an adaptive node (square node) with a node function.

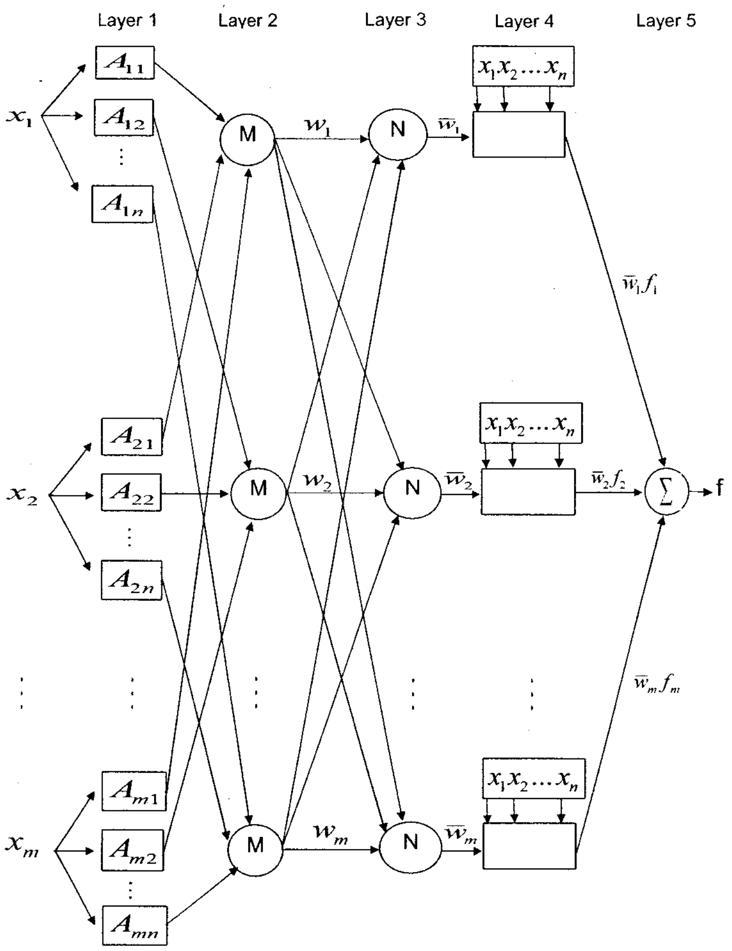


Figure 7.8: ANFIS structure

$$O^1_{ij} = \mu_{A_{ij}}(x_j) \quad i = 1,2,...,m \text{ and } j = 1,2,...,n \tag{7.20}$$

where x_j is the input to node ij and $\mu_{A_{ij}}(x_j)$ is the degree of the fuzzy membership function represented by the node. O^1_{ij} can adopt any fuzzy membership function. Usually a Gaussian function is chosen to model the $\mu_{A_{ij}}(x_j)$.

$$\mu_{A_{ij}}(x_j) = e^{-\frac{x_j - c_i}{b_i}} \tag{7.21}$$

where b_i and c_i are the parameters of the membership function.

2. Layer 2: The nodes in this layer are fixed (circle nodes). They multiply (labeled with M) the incoming signals and send the product out. O^2_i which is the output of the second layer is also called firing strengths of the rules.

$$O^2_i = w_i = \prod_{j=1}^{n} \mu_{A_{ij}}(x_j) \quad i = 1,2,...,m \tag{7.22}$$

3. Layer 3: Every node in this layer is fixed labeled with N indicating normalization operation. The i^{th} node calculates the ratio of the i^{th} rule firing strength to the sum of all rules firing strengths. Output of this layer called normalized firing strengths.

$$O^3_i = \bar{w}_i = \frac{w_i}{\sum_{i=1}^{m} w_i} \tag{7.23}$$

4. Layer 4: Every node i in this layer is an adaptive with a node function. The output of each node is the product of the normalized firing strength and a first order polynomial. Parameters in this layer will be referred to as consequent parameters.

$$O_i^4 = \bar{w}_i f_i = \bar{w}_i(f_1 = p_{10} + p_{11}x_1 + p_{12}x_2 ... p_{1n}x_n) \quad i = 1,2,...,m \tag{7.24}$$

5. Layer 5: The last node is a fixed node labeled with $\sum$. This node computes the overall output as the summation of all the incoming signals.

$$O^5 = \sum_{i=1}^{m} \bar{w}_i f_i = \frac{\sum_{i=1}^{m} w_i f_i}{\sum_{i=1}^{m} w_i} \tag{7.25}$$

In ANFIS architecture, $\{b_i, c_i\}$ are called premise parameters whereas $\{a_{m0}, a_{m1}, ..., a_{mn}\}$ are called consequent parameters. These parameters are set in the first and fourth layers, respectively (adaptive layers). An algorithm combining the least squares method and the gradient descent method is used to train the parameters. It has two stages, a forward pass stage and a backward pass stage. Given that the premise parameters are fixed, the least squares method is adopted to optimize the consequent parameters in the forward pass stage [Guler and Ubeyli 2005]. Once the optimal consequent parameters are found, the backward pass starts immediately. In the backward pass stage, the gradient descent method is used to optimize the fuzzy premise parameters. After finding the consequent parameters in the forward pass, the output of the ANFIS is calculated. Finally, the error of output and a standard back propagation algorithm is used to re-optimize the premise parameters. It has been proven that this hybrid algorithm is highly efficient in training the ANFIS [Jang 1992b, 1993].

7.5.2. ANFIS Classification Model and Corresponding Experimental Results

The fuzzy logic algorithm is much closer in spirit to human thinking than traditional logical systems. The main problem with fuzzy logic classifier is related to the choice of the suitable parameters [Castillo and Melin 2002]. For this reason, we apply the ANFIS methodology on extracted features in Section 7.4.2 to construct the fuzzy model. As we previously mentioned in Section 7.2, images are classified into three different clusters as follows:

- Images without tool marks (class 1)
- Images with few tool marks (class 2)
- Images with many tool marks (class 3)

The experiment is conducted on the earlier developed features. Therefore, training data are generated from 30 randomly selected sample images (10 images of class 1, 10 images of class 2 and 10 images of class 3) using Equations (7.9), (7.13), (7.17), (7.18) and (7.19). The linguistic Sugeno rules are established considering the dynamic behavior of the different tool marks and analyzing the error and its variation. In this application, these Sugeno rules are expressed as follows. The consequent of a rule can be expressed as a polynomial function of the inputs, and the order of the polynomial also determines the order of the rule. The 5-input first-order Sugeno model with m rules has the following form:

$$\text{If } v_n^{(1)} \text{ is } A_{11} \text{ and } v_n^{(2)} \text{ is } A_{12} \ldots v_n^{(5)} \text{ is } A_{15} \text{ then} \tag{7.26}$$

$$Z_1 = p_{10} + p_{11}u_1 + p_{12}u_2 + \ldots + p_{15}u_5$$

$$\text{If } v_n^{(1)} \text{ is } A_{21} \text{ and } v_n^{(2)} \text{ is } A_{22} \ldots v_n^{(5)} \text{ is } A_{25} \text{ then} \tag{7.27}$$

$$Z_2 = p_{20} + p_{21}u_1 + p_{22}u_2 + \ldots + p_{25}u_5$$

$$\vdots$$

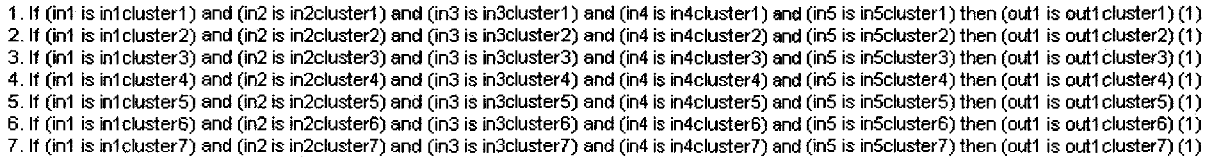
$$\text{If } v_n^{(1)} \text{ is } A_{m1} \text{ and } v_n^{(2)} \text{ is } A_{m2} \ldots v_n^{(5)} \text{ is } A_{m5} \text{ then} \tag{7.28}$$

$$Z_m = p_{m0} + p_{m1}u_1 + p_{m2}u_2 + \ldots + p_{m5}u_5$$

We define the output of the training model for images without tool marks (class 1) as one, for images with few tool marks (class 2) as two and finally for images with many tool marks (class 3) as three. Therefore, $O^{(n)}$ is defined as the value for the output of the n^{th} test image in the model. The corresponding equation is as follows:

$$O^{(n)} = \begin{cases} 1; & \text{image with no tool marks} \\ 2; & \text{image with few tool marks} \\ 3; & \text{image with many tool marks} \end{cases} \tag{7.29}$$

We used the ANFIS methodology to estimate the parameters of the membership functions and the consequent functions. To generates a fuzzy inference system with a suitable number of rules and to distinguish the fuzzy qualities associated with each of the clusters, fuzzy subtractive clustering [Chiu 1994] is adopted. Fuzzy subtractive clustering is capable of identifying the structure of the fuzzy rule base for ANFIS. The subtractive clustering method partitions the data into groups called clusters. After an enumerative search on the subtractive clustering parameters, the following values for those parameters are chosen (range of influence $r_a = .5$, squash factor $\eta = 1.25$, accept ratio $\bar{\varepsilon} = .5$ and reject ratio $\underline{\varepsilon} = .15$). As a result, we identified a fuzzy model composed of seven rules. However, it should be noted that the processing and training time for finding the optimal subtractive clustering parameters can be quite long. Hence, an exhaustive search of all possible subtractive clustering parameter combinations was not feasible. In addition, it is beyond the scope of this thesis to find the optimum set of initializing subtractive clustering parameters. The fuzzy rules automatically generated by the ANFIS method are shown in Figure 7.9. We also show in Table 7.1 and 7.2 the centers and standard deviations of Gaussian membership functions generated by ANFIS. We present in Figure 7.10 the fuzzy rule viewer of MATLAB, which shows the use of the fuzzy system for calculating the output of the model for specific input values. Finally, the consequent parameters are given in Table 7.3.



```
1. If (in1 is in1cluster1) and (in2 is in2cluster1) and (in3 is in3cluster1) and (in4 is in4cluster1) and (in5 is in5cluster1) then (out1 is out1cluster1) (1)
2. If (in1 is in1cluster2) and (in2 is in2cluster2) and (in3 is in3cluster2) and (in4 is in4cluster2) and (in5 is in5cluster2) then (out1 is out1cluster2) (1)
3. If (in1 is in1cluster3) and (in2 is in2cluster3) and (in3 is in3cluster3) and (in4 is in4cluster3) and (in5 is in5cluster3) then (out1 is out1cluster3) (1)
4. If (in1 is in1cluster4) and (in2 is in2cluster4) and (in3 is in3cluster4) and (in4 is in4cluster4) and (in5 is in5cluster4) then (out1 is out1cluster4) (1)
5. If (in1 is in1cluster5) and (in2 is in2cluster5) and (in3 is in3cluster5) and (in4 is in4cluster5) and (in5 is in5cluster5) then (out1 is out1cluster5) (1)
6. If (in1 is in1cluster6) and (in2 is in2cluster6) and (in3 is in3cluster6) and (in4 is in4cluster6) and (in5 is in5cluster6) then (out1 is out1cluster6) (1)
7. If (in1 is in1cluster7) and (in2 is in2cluster7) and (in3 is in3cluster7) and (in4 is in4cluster7) and (in5 is in5cluster7) then (out1 is out1cluster7) (1)
```

Figure 7.9: Fuzzy rules generated by the ANFIS method

Table 7.1
Centers of Gaussian membership functions (μ)

Cluster (No)	First input (center)	Second input (center)	Third input (center)	Fourth input (center)	Fifth input (center)
Rule 1	0.065	0.036	0.083	0.105	0.153
Rule 2	0.306	0.110	0.318	0.730	0.675
Rule 3	0.658	0.623	0.619	0.546	0.561
Rule 4	0.245	0.044	0.244	0.975	0.998
Rule 5	0.011	0.011	0.013	0.605	0.547
Rule 6	1	0.778	0.999	0.323	0.460
Rule 7	0.807	0.997	0.802	0.687	0.519

By applying the ANFIS model to the whole database, composed of 105 images, we get 98.1% correct identification rate of all the images. Only two images (images 20 and 30) are misidentified. They belong to first class images while the model mark them as second class images. Figure 7.11 demonstrates the ANFIS error for the whole database. The first 35 test images come from the first class, the second 35 test images come from the second class and the rest belong to the third class. As seen in Figure 7.11, images 20 and

30 have the highest error, respectively. While the largest absolute error in 105 images is 0.862, the average absolute error in 105 images is obtained as 0.058. The processing time per test image is about 270 seconds on a Pentium 4 (3 GHz), 1 GB RAM and a Windows XP platform (90 seconds for pre-processing and 180 seconds for feature extraction and decision making).

Table 7.2
Standard deviations of Gaussian membership functions (σ)

Cluster	First input	Second input	Third input	Fourth input	Fifth input
(No)	Deviation	Deviation	Deviation	Deviation	Deviation
Rule 1	0.177	0.176	0.177	0.178	0.177
Rule 2	0.169	0.172	0.170	0.164	0.164
Rule 3	0.174	0.171	0.175	0.174	0.176
Rule 4	0.180	0.176	0.179	0.177	0.178
Rule 5	0.156	0.173	0.148	0.187	0.173
Rule 6	0.176	0.177	0.176	0.176	0.177
Rule 7	0.178	0.182	0.177	0.176	0.177

7.6. Conclusions

Products are usually graded on the basis of humanely defined characteristics for purchasing, manufacturing, marketing or selling purposes. They are usually categorized based on some recognizable characteristics that are of value to consumers. However, performing this process humanly is costly and monotonous. In this chapter, a fuzzy-neural approach for classification of the gray level tool marks using developed wavelet based and statistical features was proposed. Using adaptive network based fuzzy inference system (ANFIS), the developed model successfully classified 98.1% of gray level water pump images of a database with a size of 105 to the appropriate category. The presented neuro-fuzzy model

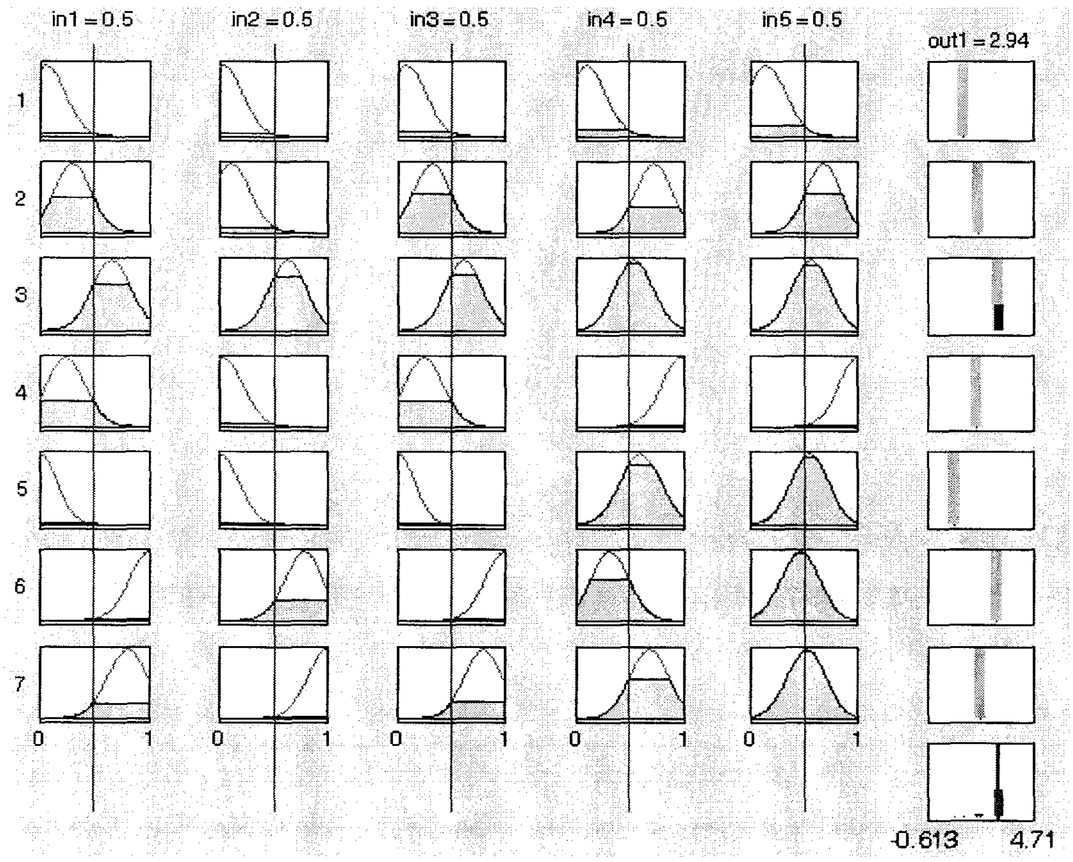


Figure 7.10: Fuzzy rule viewer for calculating the output of the fuzzy system for specific values

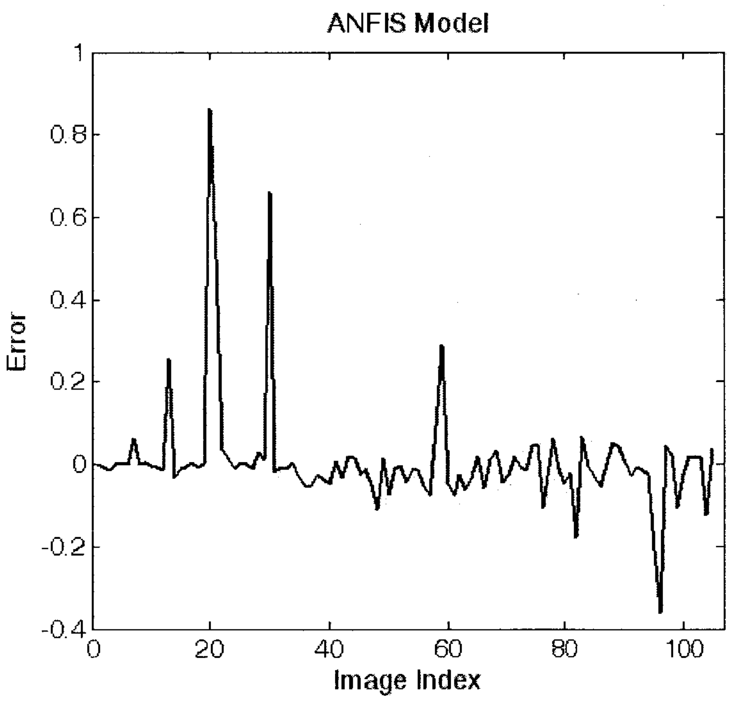


Figure 7.11: ANFIS error for the whole database

Table 7.3
ANFIS Sugeno parameters

	p_{i0}	p_{i1}	p_{i2}	p_{i3}	p_{i4}	p_{i5}
Rule 1	0.984	0.050	0.548	-0.104	-0.111	0.081
Rule 2	2.125	-0.823	-0.117	0.556	-0.136	0.114
Rule 3	2.647	0.366	0.153	-0.218	0.422	-0.107
Rule 4	2.055	0.256	-0.502	-0.185	0.132	-0.181
Rule 5	1.236	-0.547	0.246	-0.277	0.113	-0.580
Rule 6	1.467	0.060	0.867	0.142	1.059	0.668
Rule 7	1.046	0.869	0.779	1.014	-0.231	-0.383

demonstrated very interesting characteristics which was able to respond to different dynamic conditions and mimic human expectation when it comes to proper classification of parts. The developed system is adaptable to other machine vision based automated inspection applications.

Chapter 8

Conclusions and Future Research Directions

The role of machine vision based inspection in achieving enhanced quality and productivity is of paramount importance. There is a trend towards more reliable methods of inspection such as machine vision based inspection, automated vision inspection, computer aided inspection, etc. But, for these modern inspection systems to be attractive, they need to be cost-effective. Moreover, these vision systems should be well suited and tuned to their applications specifically given the fact that the range and scope of machine vision based inspection applications are obviously very wide and diversified. The present dissertation was based on some machine vision based inspection projects which is a collection of developed efficient fuzzy vision techniques endorsed with the experimental results. As a result, the main objective of this work was to develop new supervised and unsupervised fuzzy machine vision based inspection models. Following are the summary, contributions and some recommendations.

1. In the third chapter, an inspection problem faced by a Canadian automotive parts manufacturer was used as a case study. The problem was related to a vision system

which was operating to confirm the placement of metal fastening clips on a structural member that supports a truck dash panel. The objective was to identify the presence or absence of metal clips inserted by a robot arm. It took the manufacturer over 8 months to tune its commercial machine vision system to detect missing clips and yet the accuracy and efficiency of the system were not at desired level (allowable percentage defective is less than 0.1%). To this end, we developed a novel fuzzy machine vision based inspection method for clip detection. At first, we investigated different statistical based features including mean, variance, Euclidean distance, correlation, absolute error and least square error metrics. We demonstrated that none of these approaches lead to 100% correct identification of missing clips. After determination of the interest zone, which is almost a must for any automated vision system, the rest of the machine vision steps are done automatically. No special setting or direct threshold has been applied in our fuzzy model. Our clip detection model relies on fuzzy color processing using composite images and developed XOR feature extractors. The proposed model completely detects present and missing clips among 1910 industrial images. The robustness of the fuzzy model is confirmed by its strong performance in the entire database. We believe that our proposed model is a reliable alternative for machine vision based inspection in adverse industrial situations where it is not possible to sufficiently control the environment.

2. In the fourth chapter, we proposed an objective fuzzy clip orientation identification approach for fast vision based inspection. A two-stage fuzzy model was developed to inspect the orientation of the placement of the above mentioned metal fastening clips. In the first stage, a fuzzy clip detection model based on HSL (hue, saturation and luminance) features was developed to investigate the existence of the clip.

In the second stage, a fuzzy orientation detection model using subtractive clustering was devised to examine if the orientation of the clip is properly placed. The proposed model is based on grouping and counting points (pixels) in pre-selective regions. After training the model, different numbers of pixels in different regions ultimately determine whether the orientation of the clip is acceptable or not. The model significantly reduces the probability of false orientation inspection. It also eliminates the disturbance of artificial lights since it tracks the orientation of objects in separated RGB channels. More importantly, this method can reduce the execution time due to the fact that only small portions of pixel edges in the image are examined. The developed method can be easily tuned and parameterized based on the shape and geometric positions of the test components in the assembly line. It properly identifies the orientation of all the clips in the entire database with a size of 708 images. It is clear that adopting such system will make automated fuzzy machine vision based orientation detection a more feasible alternative in many applications.

3. In the fifth chapter, the experimental research outlines the experience with one industrial example which involves the broken edge inspection of die cast aluminum automotive water pump housings. The chapter proposes a collection of novel steps to arrive at a model to inspect images taken in a noisy environment to identify broken edges. It illustrates the process of extracting the amount of comparability of the edges in a challenging context based on a predefined template to achieve a pairwise comparison approach. By developing three broken edge verification features and using fuzzy C-means clustering, we provide a fuzzy broken edge inspection model that classifies water pumps into three classes; pumps with significant broken edges, pumps with insignificant broken edges and pumps with no broken edges.

The first proposed feature, pair-wise edge based feature, aims to compare the width of broken edge of test and reference images. The second proposed feature, derivative based feature, compares the derivative of test and reference function images and finally the third proposed feature, band edge based feature, measures the depth of the broken edge of test and reference images. The devised machine vision based inspection method is efficient in terms of time processing and accuracy to recognize the parts with broken edges and to make decisions consistent with human knowledge. The proposed method not only detects broken edge defects as long as they are breaking the homogeneity of textures but also it can determine the types of broken edge defects according to their levels of severity. The fuzzy broken edge detection was carried out on a database of 150 gray level images. The developed method properly identifies 94.6% of the broken edges in the entire database. It is worth mentioning that the proposed model can easily be tuned and implemented for inspection purpose in many assembly lines.

4. In the sixth chapter, we presented an objective fuzzy approach for fast and accurate porosity machine vision based inspection. A computer aided methodology of detection of pores which are formed in aluminum alloys during production of water-pumps for car engines with die casting method is presented. The main complexity of this problem arises from the similarity between pores and tool marks. The proposed method is based on the correlations of the core of pore candidates with twelve developed matrices resulted in five novel features. The fuzzy porosity detection was carried out on a database of 105 gray level images. The proposed method correctly identifies 93.4% of the pores in the entire database. Although, the relationship between the pore characteristics and the texture attributes of die cast parts is very complex in nature, by using the proposed fuzzy machine vision based method the

pores can be automatically detected. The model demonstrated that the developed system can provide satisfactory results by using developed perceptual features in a computationally efficient way. Obviously, adopting such systems adds great value to the whole production system by increasing the confidence of the inspectors in the machined vision based performance of the real-time porosity detection.

5. Products are usually graded on the basis of humanly defined characteristics for purchasing, manufacturing, marketing or selling purposes. They are usually categorized based on some recognizable characteristics that are of value to consumers. But doing this process humanly is costly and monotonous. In the seventh chapter, a machine vision system for image classification of the earlier mentioned aluminum alloy water pumps is presented. This particular application was considered to be challenging due to variations of part position, light changes and the non-homogenous water pump textures. The developed neural-fuzzy approach model is able to interpret gray level tool marks in a fuzzy framework. The proposed vision categorizes images to accepted, rework or rejected. The neuro-fuzzy classification model was carried out on a database of 105 gray level images. The proposed adaptive neuro-fuzzy inference method properly classified 98.1% of the images to the right classes in the entire database. The presented neuro-fuzzy model demonstrated very interesting characteristics which are able to respond to different dynamic conditions and mimic human judgements to assign parts to proper classes. The developed model is also adaptable to many other machine vision based part classification for different applications.

The technical contributions in this research are as follows:

1. The first contribution is the development of the concept of composite matrices in conjunction with XOR feature extractor using fuzzy subtractive clustering for clip

detection.

2. The second contribution is about a proposed model based on grouping and counting pixels in pre-selective areas which tracks pixel colors in separated RGB channels to determine whether the orientation of the clip is acceptable or not.

3. The construction of three novel edge based features embedded in fuzzy C-means clustering for broken edge detection marks the third contribution.

4. The fourth contribution presents the core of porosity candidates concept and its correlation with twelve developed matrices. This, in turn, results in the development of five different features used in our fuzzy machine vision based porosity detection approach.

The scientific contributions in this research are as follows:

1. Designing intelligent color based algorithm to find the existence of a non predefined object in the image.

2. Designing intelligent geometric based method to find the orientation of an object in the image.

3. The broken edge detection framework has provided concrete and reasonable approach for class-based systems.

4. The porosity framework provides a new approach to distinguish pores and tool marks from each other.

5. Proposing a new method to classify tool marks based on their severity.

6. The developed methods have proven to be successful in assisting Canadian automotive industry.

Although the work presented in this thesis accounts for only a limited attempt on achieving the goal of fully automated vision inspection, it is my hope that this thesis motivates more researchers to further examine this promising field in the future. To this end, the following recommendations are suggested to launch further research in this field:

1. Future work may address processing of the multiple component detection and orientation identification in one image.

2. Using more than one camera for 3D imaging, in which case, the combination of information obtained from all cameras to get a consolidated picture would pose as a promising avenue for future research.

3. Comparing the strength of each developed feature, in either broken edge or porosity detection problems, seems interesting.

4. Measuring the effects of the correlation of different proposed matrices on pores and tool marks could be a breakthrough in developing more sophisticated methods to distinguish challenging pores from tool marks.

5. A benchmark between different fuzzy clusterings and classifications on specific applications may eventually lead to the design of more advanced vision based classifiers.

Bibliography

Abraham, E., Younus, A., Fatimy, A., Delagnes, J., Nguma, E., and Mounaix, P., 2009. Broadband terahertz imaging of documents written with lead pencils. *Optics Communications*, **282 (15)**, 3104–3107.

Abramovich, G., 2004. Part Inspection by Developmental Vision. 1386. PhD thesis, Biblioteques Universitaries de Catalunya, p. 87.

Albovik, R., 1998. Handbook of image and video processing. 5. Academic Press, New York, p. 50.

Albovik, R. and Daly, P., 1998. Handbook of image and video processing. Academic Press, New York,

Alhichri, H. and Kamel, M., 2003. Virtual circles: a new of features for fast image registration. *Pattern Recognition Letters*, **24 (9-10)**, 1181–1190.

Allada, V. and Anand, S., 1995. Feature-based modeling approaches for integrated manufacturing: state-of-the-art survey and feature research directions. *International Journal of Computer Integrated Manufacturing*, **8 (6)**, 411–440.

Arivazhagan, S., Ganesan, L., and Padam Priyal, S., 2006. Texture classification using gabor wavelets based rotation invariant features. *Pattern Recognition Letters*, **27 (16)**, 1976–1982.

Avci, E. and Sengur, H. D., A., 2009. An optimum feature extraction method for texture classification. *Expert Systems with Applications*, **36 (3)**, 6036–6043.

Awcock, G. and Thomas, R., 1996. Applied image processing, 1st Edition. 1. McGraw-Hill, NewYork, pp. 228–30.

Barth, M. and Barrows, C., 2003. Rapid omnidirectional vision acquisition using an intelligent linear scanning technique. *Machine vision and applications*, **14**, 8593.

Batchelor, B. and Waltz, F., 1990. Machine Vision Systems Integration in Industry. 1386. Bellingham, Wash, Boston, Massachusetts, p. 57.

Bellman, R., R., K., and Zadeh, L., 1966. Abstraction and pattern classification. *Math. Anal*, **13 (1)**, 1–7.

Bezdek, C. R., J.C. and Attikiouzel, Y., 1998. A geometric approach to edge detection. *IEEE Transactions on Fuzzy Systems*, **6 (1)**, 52–57.

Bezdek, J., 1981. Pattern recognition with fuzzy objective function algoritms. Plenum Press, New York,

Bezdek, J. and Pal, S., 1992a. Fuzzy models for pattern recognition: Methods that search for Structures in Data. IEEE Press, New York, p. 45.

Bezdek, J. and Pal, S., 1992b. Fuzzy models for pattern recognition: methods that search for structures in data. IEEE Press, NewYork,

Binaghi, E., 2007. A fuzzy logic inference model for a rule based system in medical diagnosis. *Expert Systems*, **7 (3)**, 134–141.

Bloch, I., 2005. Fuzzy spatial relationships for image processing and interpretation: a review. *Image and Vision Computing*, **23 (2)**, 89–110.

Boutell, M. and Luo, J., 2005. Automatic image orientation detection via confidence-based integration of low-level and semantic cues. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **27 (5)**, 715–726.

Byun, H. and Lee, K., 2004. A decision support system for the selection of a rapid prototyping process using the modified TOPSIS method. *The International Journal of Advanced Manufacturing Technology*, **26 (11-12)**, 1338–1347.

Cai, L. and Kwan, H., 1998. Fuzzy classifications using fuzzy inference networks. *IEEE Transactions on Systems, Man and Cybernetic*, **28 (1)**, 334–347.

Canny, J., 1986. A computational approach to edge detection. *IEEE Transaction. Pattern Analysis and Machine Intelligence*, **8**, 679–714.

Caprari, R., 2000. Algorithm for text page up/down orientation determination. *Pattern Recognition Letter*, **4**, 311–317.

Castillo, O. and Melin, P., 2002. A hybrid fuzzy-fractal approach for time series analysis and plant monitoring. *International Journal of Intelligent Systems*, **17 (8)**, 751–765.

Caulfield, F. J., H.J. and Yoo, S., 2004. Artificial color image logic. *Information Sciences*, **167 (1-4)**, 1–7.

Cetingl, H., Erzin, E., Yemez, Y., and Tekalp, A., 2006. Multimodal speaker/speech recognition using lip motion, lip texture and audio. *Signal Processing*, **86 (12)**, 3549–3558.

Chandraratne, M., Kulasiri, D., and Samarasinghe, S., 2007. Classification of lamb carcass using machine vision: Comparison of statistical and neural network analyses. *Journal of Food Engineering*, **82 (1)**, 26–34.

Chang, H. and Xu, H., 2002. A novel fuzzy logic approach to mammogram contrast enhancement. *Information Sciences*, **148 (1)**, 1–4.

Chaudhuri, B. and Pal, U., 1997. Skew angle detection of digitized indian script documents. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **19**, 182–186.

Chen, B. and Hoberock, L., 1996. Machine vision fuzzy object recognition and inspection using a new fuzzy neural network. International Symposium on Intelligent Control. Dearborn, MI, pp. 206–211.

Chen, C. and Chen, C., 1999. Filtering methods for texture discrimination. *Pattern Recognition Letters*, **20 (8)**, 783–790.

Chen, S. and Chen, S., 2005. A prioritized information fusion method for handling fuzzy decision making problems. *Applied Intelligence*, **22 (3)**, 219–232.

Chen, Y., 1995a. Computer vision for general purpose visual inspection: a fuzzy logic approach. *Optics and Lasers in Engineering*, **22 (3)**, 181–192.

Chen, Y., 1995b. Computer vision for general purpose visual inspection: a fuzzy logic approach. *Optics and Lasers in Engineering*, **22**, 181–192.

Cheng, C. C. C. H., H.D. and Xu, H., 1998. Fuzzy homogeneity approach to multilevel thresjholding. *IEEE Trans. on Image Processing*, **7 (7)**, 1084–1088.

Cheng, C. Y., H.D. and Jiang, X., 2000. Thresholding using twp-dimensional histogram and fuzzy entropy principle. *IEEE Trans.s on Image Processing*, **9 (4)**, 732–735.

Cheng, H. and Glazier, C., 2006. Vehicle detection and classification using model-based and fuzzy logic approaches. *In transportation Research Board*, **1935 (1)**, 154–162.

Cheng, H. and Sun, Y., 2000. Hierarchial approach to color image segmentation using homogeneity. *IEEE Trans. On Image Processing*, **9 (12)**, 2071–2082.

Chien, C., 1995. A computer vision system for extravehicular activity helper/retriever. *Applied Intelligence*, **5 (3)**, 251–268.

Chiu, S., 1994. Fuzzy model identification based on cluster estimation. *Journal of Intelligent and Fuzzy Systems*, **2 (3)**, 267–278.

Choi, Y. and Krishnapuram, R., 1997. A robust approach to image enhancement based on fuzzy logic. *IEEE Trans Image Processing*, **6 (6)**, 808–825.

Chui, C., 1992. An introduction to wavelets. San Diego, Academic Press,

Cinque, F. G., L. and Lombardi, L., 2004. A clustering fuzzy approach for image segmentation. *The Journal of Pattern Recognition Society*, **37 (9)**, 17971807.

Cornelius, T., 1998. Neural network systems, techniques and applications, 5th Edition. Image processing and pattern recognition. Academic Press, NewYork,NY, p. 70.

Cowan, C., Modayur, B., and DeCurtins, J., 1992. Automatic light-source placement for detecting object features. *Journal of SPIE*, **182 (6)**, 397–408.

Cucchiara, R., Prati, A., and Vezzani, R., 2007. A multi camera vision system for fall detection and alarm generation. *Expert Systems*, **24 (5)**, 334–345.

Dai, X. and Khorram, S., 1997. Development of a feature based approach to automated image registration for multi temporal and multi sensor remotely sensed imagery. International Geoscience and Remote Sensing Symposium (IGARSS). Singapore, pp. 243–245.

Daliri, M. and Torre, V., 2009. Shape and texture clustering: Best estimate for the clusters number. *Image and Vision Computing*, **27 (10)**, 1603–1614.

Davies, E., 2005a. Machine vision: theory, algorithms, practicalities, 3rd Edition. Morgan Kaufmann, NewYork,NY,

Davies, E., 2005b. Machine vision: theory, algorithms, practicalities, 3rd Edition. Morgan Kaufmann, New York, NY,

Demirli, K., Cheng, S., and Muthukumaran, P., 2003. Subtractive clustering based modeling of job sequencing with parametric search. *Fuzzy Sets and Systems*, **137 (2)**, 235–270.

Demirli, K. and Muthukumaran, P., 2000. Higher order fuzzy system identification using subtractive clustering. *Journal of Intelligent and Fuzzy Systems*, **9 (4)**, 129–158.

Dereli, T. and Filiz, H., 2002. A note on the use of step standard for interfacing design to process planning. *Computer Aided Design*, **34 (14)**, 1075–1085.

Deriche, R., 1987. Using canny's criteria to derive an optimal edge detector recursively implemented. *International Journal of Computer Vision*, **2**, 15–20.

Dobrzanski, L., Krupinski, M., and Sokolowski, J., 2005. Computer aided classification of flaws occurred during casting of aluminium. *Journal of materials processing technology*, **167**, 456–462.

Du, C. and Sun, D., 2005. Automatic measurement of pores and porosity in pork ham and their correlations with processing time, water content and texture. *Meat science*, **72**, 294–302.

Duan, T., Hong Du, T., Phuoc, T., and Hoang, N., 2005. Building an automatic vehicle license plate recognition system. In: Proc. Int. Conf. Comput. Sci. RIVF. pp. 59–63.

Dunn, J., 1973. A fuzzy relative of the isodata process and its use in detecting compact well-separated clusters. *Journal of Cybernetics*, **3 (1)**, 32–57.

Edinbarough, I., Balderas, R., and Bose, S., 2005. A vision and robot based on-line inspection monitoring system for electronic manufacturing. *Computers in Industry*, **56**, 986–996.

Feigenbaum, A., 1961. Total Quality Control. McGraw-Hill, New York, p. 56.

Fletcher, M., 1996. Fully automatic inspection of welds now a reality. *NDT international*, **29 (3)**, 191–199.

Flusser, J. and Suk, T., 1998. Degraded image analysis: an invariant approach. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **20 (6)**, 590–603.

Francoa, A. and Nanni, L., 2009. Fusion of classifiers for illumination robust face recognition. *Expert Systems with Applications*, **36 (5)**, 8946–8954.

Frisch, A., 2006. Unsupervised construction of fuzzy measures through self-organizing feature maps and its application in color image segmentation. *International Journal of Approximate Reasoning*, **41 (1)**, 23–42.

Galloway, M., 1975. Texture analysis using gray level run lengths. *Computer Graphics and Image Processing*, **4**, 172–179.

Gao, J., Zheng, D., and Gindy, N., 2004. Extraction of machining features for cad/cam integration. *International Journal of Advanced Manufacturing Technology*, , 573–581.

Gonzales, R., Woods, R., and Eddins, S., 2004. Digital image processing using MATLAB. Prentice Hall, Saddle Riber, NJ, Gonzalez, R. and Wintz, P., 1977. Digital Image Processing. Addison-Wesley,Reading,MA,

Grimm, F. and Bunke, H., 2007. An expert system for the selection and application of image processing subroutines. *Expert Systems*, **10 (2)**, 61–74.

Guler, I. and Ubeyli, D., 2005. Adaptive neuro-fuzzy inference system for classification of eeg signals using wavelet coefficients. *Journal of Neuroscience Methods*, **148**, 113–121.

Guo, H., Zhang, Q., and Nandi, A., 2008. Feature extraction and dimensionality reduction by genetic programming based on the fisher criterion. *Expert Systems*, **25 (5)**, 444–459.

Guo, L. C., S.M. and Hsu, C., 2005. An intelligent image agent based on soft-computing techniques for color image processing. *Expert Systems with Applications*, **28 (3)**, 483–494.

Hangai, Y. and Kitahara, S., 2009. Quantitative evaluation of porosity in aluminum alloy die castings by fractal analysis of spatial distribution of area. *Materials and design*, **30 (4)**, 1169–1173.

Haralick, R., Shanmugam, K., and Dinstein, I., 1973. Texture features for image classification. *IEEE Transactions on System, Man, Cybernetics*, **8 (6)**, 610–621.

Hasanzadeh, R. A. S. S. M. M., P.R.R. and Ahmadi, M., 2007. A density-based fuzzy clustering technique for non-destructive detection of defects in materials. *NDT and E International*, **40 (4)**, 337–346.

He, Z., You, X., and Yuan, Y., 2009. Texture image retrieval based on non-tensor product wavelet filter banks. *Signal Processing*, **89 (8)**, 1501–1510.

Ho, K., 1995. Fuzzy edge detection by fuzzy categorization of edges. FEDGE, Montreal, Canada, pp. 182–196.

Hocenski, Z., Vasilic, S., and Hocenski, V., 2006. Improved canny edge detector in ceramic tiles defect detection. 32nd Annual Conference on IEEE Industrial Electronics, IECON 2006. Paris, France, pp. 3328–3331.

Hsieh, M. D., Y.C. and Perlant, F., 1992. Performance evaluation of scence registration and stereo matching for cartographic feature extraction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **14 (2)**, 214–238.

Hu, P., Johnson, W., and Griffin, P., 1986. A computer vision system for the inspection of light emitting diodes. *The International Journal of Advanced Manufacturing Technology*, **1 (2)**, 49–55.

Huang, C. and Clayton, J., 1990. Relationships between mechanical properties and microstructure of porous foods: part 1. a review. *Engineering and food. physical properties and process control*, **1**, 352–360.

Iravani Tabrizipour, M. and Toyserkani, E., 2007. An image-based feature tracking algorithm for real-time measurement of clad height. *Machine vision and applications*, **18**, 343–354.

Iscan, Z., Yksel, A., Dokur, Z., and Korrek, l. T., M., 2009. Medical image segmentation with transform and moment based features and incremental supervised neural network. *Digital Signal Processing*, **19 (5)**, 890–901.

Jang, J., 1992a. Self-learning fuzzy contrlllers based on temporal back propagation. *IEEE Transaction on Neural Networks*, **3 (5)**, 714–723.

Jang, J., 1992b. Self-learning fuzzy controllers based on temporal backpropagation. *IEEE Trans. Neural Network*, **3 (5)**, 714–723.

Jang, J., 1993. Adaptive-network-based fuzzy inference system. *IEEE Trans. Syst. Man Cybern.*, **23 (3)**, 665–685.

Jia, W., Zhanga, H., and Hea, X., 2007. Region-based license plate detection. *Journal of Network and Computer Applications*, **30 (4)**, 1324–1333.

Kannappady, D. K. M. S., S. and Shiri, N., 2006. Application of fuzzy edge detection for fast object-based image retrieval. The North American Fuzzy Information Processing Society (NAFIPS),. Montreal, Quebec, pp. 261–266.

Karmakar, D. L., G.C. and Rahman, S., 2001. Review on fuzzy image segmentation techniques. In Design and Management of Multimedia Information systems: Opportuniries and Challenges. Idea Group Publishing, New York, pp. 282–313.

Keller, J., 1997a. Fuzzy set theory in computer vision. *Fussy Sets and Systems*, **90 (2)**, 177–182.

Keller, J., 1997b. Fuzzy set theory in computer vision: A prospectus. *Fuzzy sets and systems*, **90**, 177–182.

Killing, J., Surgenor, B., and Mechefske, C., 2009. A machine vision system for the detection of missing fasteners on steel stampings. *The International Journal of Advanced Manufacturing Technology*, **41 (7-8)**, 808–819.

Killing, S. B., J. and Mechefske, C., 2006. A neuro-fuzzy approach to machine vision based part inspection. The North American Fuzzy Information Processing Society (NAFIPS). Montreal, Quebec, pp. 696–701.

Killing, S. B., J., Yang, K., and Bone, G., 2007. On the need for robust automated inspection: two machine vision examples. The 2nd International Conference on Changeable, Agile, Reconfigurable and Virtual Production (CARV). Toronto, Ontario, Canada.

Kim, L. W., D.S. and Kweon, I., 2004. Automatic edge detection using 3 × 3 ideal binary pixel patterns and fuzzy-based edge thresholding. *Pattern Recognition Letters*, **25 (1)**, 101–106.

Kim, T. and Han, J., 1998. Edge representation with fuzzy sets in blurred images. *Fuzzy sets and systems*, **100 (1)**, 77–87.

Konstantinidis, G. A., K. and Andreadis, I., 2005. Image retrieval based on fuzzy color histogram processing. *Optic Communications*, **248 (4-6)**, 375–386.

Kouatli, I. and Jones, B., 1990. *Journal of Intelligent Manufacturing*, **1 (4)**, 231–244.

Krishnapuram, R. and Keller, J., 1992. Fuzzy set theoretic approach to computer vision: an overview. IEEE International Conference on Fuzzy Systems. San Diego, CA, USA, pp. 135–142.

Kulkarni, A. and Cavanaugh, C., 2000. Fuzzy neural network models for classification. *Applied Intelligence*, **12 (3)**, 207–215.

Lashkia, V., 2001. Defect detection in x-ray images using fuzzy reasoning. *Image and Vision Computing*, **19**, 261–269.

Le, D., Thoma, G., and Wechsler, H., 1994. Automated page orientation and skew angle detection for binary document images. *Pattern Recognition*, **27 (10)**, 1325–1338.

Lee, E. and Kim, K. H., P.K., 1994. Automatic recognition of a car license plate using color image processing. In: IEEE International Conference on Image Processing. Austin, Texas, pp. 301–305.

Lee, K. and Bien, Z., 1997. A model based machine vision system using fuzzy logic. *International Journal of Approximate Reasoning*, **16 (1)**, 119–135.

Levkowitzh, G. and Herman, T., 1993. A generalized lightness, hue, and saturation color model. *Graphical Models and Image Processing*, **55 (4)**, 271–285.

Li, B. and Meng, M., 2009. Texture analysis for ulcer detection in capsule endoscopy images. *Image and Vision Computing*, **27 (9)**, 1336–1342.

Li, J. X., W. and Wang, Y., 1998. Road recognition for vision navigation of an autonomous vehicle by fuzzy-reasoning. *Fuzzy Sets and Systems*, **93 (3)**, 275–280.

Li, L. G., W. and Wang, Y., 1997. Recognizing white line markings for vision-guieded vehicle navigation by fuzzy reasoning. *Pattern Recognition Latters*, **18 (8)**, 771–780.

Li, Y. and Lau, C., 1989. Development of fuzzy algorithm for servo systems. *IEEE Control Systems Magazine*, **9 (3)**, 65–72.

Liang, L. and Looney, C., 2003. Competitive fuzzy edge detection. *International Journal of Applied Soft Computing*, **3**, 123–137.

Lim, D., 2006. Robust edge detection in noisy images. *Computational statistics and data analysis*, **50 (3)**, 803–812.

Lu, Y. X., J. and Yahagi, T., 2006. A method of face recognition based on fuzzy clustering and parallel neural networks. *Signal Processing*, **86 (8)**, 2026–2039.

Lua, S., Wanga, Z., and Shenb, J., 2003. Neuro-fuzzy synergism to the intelligent system for edge detection and enhancement. *Pattern Recognition*, **36**, 2395–2409.

Maldonado, J. and Grana, M., 2009. Recycled paper visual indexing for quality control. *Expert Systems with Applications*, **36 (5)**, 8807–8815.

Mehran, P., Demirli, K., and Surgenor, B., 2006. Fuzzy clip detection using sugeno model. North American Fuzzy Information Processing Society (NAFIPS). Montreal,Quebec,Canada.

Mitra, S. and Pal, S., 2005a. Fuzzy sets in pattern recognition and machine. *Fuzzy Sets and Systems*, **156 (1)**, 381–386.

Mitra, S. and Pal, S., 2005b. Fuzzy sets in pattern recognition and machine intelligence. *Fuzzy sets and systems*, **156 (1)**, 381–386.

Mitra, S. and Sankar, K., 2005. Fuzzy sets in pattern recognition and machine. *Fuzzy Sets and Systems*, **156**, 381–386.

Muneeswaran, K., Ganesan, L., Arumugam, S., and Ruba Soundar, K., 2005. Texture classification with combined rotation and scale invariant wavelet features. *Pattern Recognition*, **38 (10)**, 1495–1506.

Nath, A. and Lee, T., 1983. On the design of a classifier with linguistic variables as inputs. *Fuzzy Set Systems*, **11 (1)**, 265–286.

Newman, T. and Jain, A., 1995. A survey of automated visual inspection. *Computer Vision and Image Understanding*, **61**, 231–262.

Ong, K. and Wang, Q., 1995. A generalised fuzzy reasoning algorithm for an object oriented expert system tool. *Expert Systems*, **12 (3)**, 199–206.

Otsu, N., 1979. A threshold selection method from gray level histograms. *IEEE Transaction on Systems, Man, and Cybernetics*, **9 (1)**, 62–66.

Ozbay, C. R., Y. and Karlik, B., 2006. A fuzzy clustering neural network architecture for classification of ecg. *Computers in Biology and Medicine*, **36 (4)**, 376–388.

Ozbay, Y., Ceylan, R., and Karlik, B., 2006. A fuzzy clustering neural network architecture for classification of ecg arrhythmias. *Computers in biology and medicine*, **36**, 376–388.

Ozols, J. and Borisov, A., 2001. Fuzzy classification based on pattern projections analysis. *Journal of patternrecognition society*, **34 (4)**, 763–781.

Pal, P. and Kumar, A., 2002. A hybrid approach for identification of 3d features from cad database for manufacturing support. *International Journal of Machine Tools and Manufacture*, **42**, 221–228.

Pal, S., Ghatak, S., De, S., and DasGupta, S., 2008. Evaluation of surface roughness of a plasma treated polymeric membrane by wavelet analysis and quantification of its enhanced performance. *Applied Surface Science*, **255 (5)**, 2504–2511.

Pal, S. and Majumder, D., 1986. Fuzzy mathecal approach to pattern recognition. Halsted Press, New York, p. 34.

Palit, A. and Popovic, D., 1999. Fuzzy logic automatic rule generation and forcasting of time series. In proceeding of IEEE international fuzzy systems conference. Seoul, South Korea, pp. 360–365.

Park, K. and Kim, J., 2005. A real-time focusing algorithm for iris recognition camera. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, **35 (3)**, 441–444.

Pastorius, W., 2001. Flexible robotic vision systems for programmable inspection. *SAE Transactions*, **110 (5)**, 1007–1012.

Pedrycz, W. and Gomide, F., 1998. An Introduction to Fuzzy Sets: Analysis and Design, 2nd Edition. MIT Press, Boston, Massachusetts, p. 86.

Peters, L., Leja, C., and Malaviya, A., 2002. A fuzzy statistical rule generation method for handwriting recognition. *Expert Systems*, **15 (1)**, 48–56.

Pham, D. and Prince, J., 1999. Adaptive fuzzy segmentation of magnetic resonance images. *IEEE Trans Medical Imaging*, **18 (9)**, 737–752.

Plataniotis, A. D., K.N. and Venetsanopoulos, A., 1996. Fuzzy adaptive filters for multi channel image processing. *Signal processing*, **55 (1)**, 93–106.

Puig, D. and Garca, M., 2003. Pixel classification through divergence-based integration of texture methods with conflict resolution. International Conference on Image Processing (ICIP). Barcelona, Spain, pp. 11–24.

Puig, D. and Garca, M., 2004. Pixel classification through divergence-based integration of texture methods with conflict resolution. *IEEE ICIP*, **2 (1)**, 1037–40.

Rahman, M., 2001. Toward prediction of porosity in foods during drying: a brief review. *Drying technology*, **19**, 1–13.

Randen, T. and Husoy, J., 1999a. Filtering for texture classification. *IEEE Trans. Pattern Anal. Mach. Intell*, **21 (4)**, 291–310.

Randen, T. and Husoy, J., 1999b. Filtering for texture classification: a comparative study. *IEEE Trans. Pattern Anal. Mach. Intell*, **21 (4)**, 291–310.

Rao, B., 2001. Visual techniques in non-destructive testing. *Encyclopedia of Materials: Science and Technology*, **1 (1)**, 60436.

Risinger, L. and Kaikhah, K., 2008. Motion detection and object tracking with discrete leaky integrate and fire neurons. *Applied Intelligence*, **29 (3)**, 248–262.

Rueda, L. and Zhang, Y., 2006. Geometric visualization of clusters obtained from fuzzy clustering algorithm. *The Journal of the Pattern Recognition Society*, **39** **(8)**, 1415–1429.

Ruspini, E., 1969. A new approach to classification. *Information control*, **15**, 22–32.

Russo, F., 1999. Fire operators for image processing. *Fuzzy Sets and Systems*, **103** **(2)**, 265–275.

Russo, F. and Ramponi, G., 1994. Edge extraction by fire operators. 3rd IEEE International Conference on Fuzzy Systems. Orlando, Florida, pp. 249–253.

Sainarayanan, N. R., G. and Yaacob, S., 2007. Fuzzy image processing scheme for autonomous navigation of human blind. *Applied Soft Computing*, **7** **(1)**, 257–264.

Saracoglu, T. K., R. and Allahverdi, N., 2007. A fuzzy clustering approach for finding similar documents using a novel similarity measure. *Expert Systems with Applications*, **33** **(3)**, 600–605.

Sengur, A., 2007. Wavelet transform and adaptive neuro-fuzzy inference system for color texture classification. *Expert Systems with Applications*, **34** **(3)**, 2120–2128.

Shapiro, L. and Stockman, G., 2000. Computer vision. Prentice-Hall, Englewood Cliffs, NJ,

Sklansly, J., 1978. Image segmentation and feature extraction. *IEEE Transaction on Systems Man and Cybernetic*, **8** **(4)**, 237–247.

Smith, L. and Smith, M., 2005. Automatic machine vision calibration using statistical and neural network methods. *Image and Vision Computing*, **23** **(10)**, 887–899.

Smolka, B., 2003. On the new robust algorithm of noise reduction in color images. *Computers and Graphics*, **27 (4)**, 503–513.

Solaiman, F. R., B. and Cavayas, F., 1999. Automatic road extraction using fuzzy mask concepts. In International Geosciences and Remote Sensing Symposium. Anchorage, Alaska, pp. 894–896.

Solomon, R., 2002. As if you were there, matching machine vision to human vision. *The Hybrid Vigor Journal*, **1 (3-4)**.

Sridharan, M. and Stone, P., 2009. Color learning and illumination invariance on mobile robots: A survey. *Robotics and Autonomous Systems*, **57 (6-7)**, 629–644.

Stanley, M. R. S. W., R.J. and Aggarwal, C., 2003. A fuzzy based histogram analysis technique for skin lesion discrimination in dermatology clinical images. *Computerized Medical Imaging and Graphics*, **27 (5)**, 387–396.

Stockman, K. S., G. and Benett, S., 1982. Matching images to model for registration and object detection via clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **4 (3)**, 229–241.

Sun, D. and Brosnan, T., 2003. Pizza quality evaluation using computer vision. *Journal of Food Engineering*, **57 (1)**, 81–89.

Tamm, T. and Remm, K., 2009. Estimating the parameters of forest inventory using machine learning and the reduction of remote sensing features. *International Journal of Applied Earth Observation and Geoinformation*, **11 (4)**, 290–297.

Tao, T. W., C.W. and Taur, J., 1993. Fuzzy if-then approach to edge detection. Second IEEE International Conference on Fuzzy Systems. San Francisco, California, pp. 35–45.

Tunstel, E., Howard, A., and Seraji, H., 2002. Rulebased reasoning and neural network perception for safe offroad robot mobility. *Expert Systems*, **19 (4)**, 191–200.

Tzanakou, M., 2001. Supervised and unsupervised pattern recognition. *IEEE Transaction on Neural Networks*, **12 (3)**, 371.

Umbaugh, S., 1998. Computer vision and image processing. Prentice-Hall International Inc., Englewood Cliffs, NJ, p. 76.

Urena, R. F., R. and Berenguel, M., 2001. A machine vision system for seeds germination quality evaluation using fuzzy logic. *Computers and Electronics in Agriculture*, **32 (1)**, 1–20.

Van de Wouwer, G., Scheunders, P., Livens, S., and Van Dyck, D., 1999. signatures for color texture characterization. *Pattern Recognition*, **32 (3)**, 443–451.

Vasileisky, Z. B., A.S. and Berger, M., 1998. Automated image co-registration based on linear feature recognition. Second Conference Fusion of Earth Data. Sophia Antipolis, France, pp. 59–66.

Venkata Rao, R., 2006. A decision-making framework model for evaluating flexible manufacturing systems using digraph and matrix methods. *The International Journal of Advanced Manufacturing Technology*, **30 (11-12)**, 1101–1110.

Wang, J., 1997. Color image processing for the evaluation of stream lines in plane extrusion by fuzzy set theory. *Journal of Materials Processing Technology*, **71 (2)**, 322–328.

Wang, J., 1999. Fuzzy c-means approach to color image processing for evaluation the grid lines in plane extraction. *Journal of materials processing technology*, **91 (1-3)**, 12–17.

Wang, K. and Liu, H., 2006. A fuzzy aggregation approach to group decision making based on centroid measurement. *Expert Systems*, **23 (5)**, 313–322.

Wanga, J., Linb, C., and Chena, S., 2010. Applying fuzzy method to vision-based lane detection and departure warning system. *Expert Systems with Applications*, **37 (1)**, 113–126.

Weszka, J., Dyer, C., and Rosenfeld, A., 1976. A comparative study of texture measures for terrain classification. *IEEE Transactions on System, Man, Cybernetics*, **6**, 269–285.

Yager, R., 1992. A general approach to rule aggregation in fuzzy logic control. *Applied Intelligence*, **2 (4)**, 333–351.

Yanamura, Y., Goto, M., Nishiyama, D., Soga, M., Nakatani, H., and Saji, H., 2003. Extraction and tracking of the license plate using hough transform and voted block matching. In: Proceedings of IEEE intelligent vehicles symposium. pp. 243–6.

Yang, S., Lu, Y., and Wang, J. L., M., 2009. Low bit rate sar image coding based on adaptive multiscale bandelets and cooperative decision. *Signal Processing*, **89 (10)**, 1910–1920.

Yibo, L., 2004. Generic edge feature extraction based on perceptual curve partitioning. MCSc Thesis Defence. Computer Science, Dalhousie University, Halifax, NS,

Yin, T. Y. S. F., Z. and Sun, Z., 2006. Fuzzy clustering with novel separable criterion. *Tsinghua Science And Technology*, **11 (1)**, 50–53.

Yu, B. and Jain, A., 1996. A robust and fast skew detection algorithm for generic documents. *Pattern Recognition*, **29 (10)**, 1599–1630.

Yuksel, M., 2006. A hybrid neuro fuzzy filter for edge preserving restoration of images corrupted by impulse noise. *IEEE Trans Image Processing*, **15 (4)**, 928–936.

Yuksel, M., 2007. Edge detection in noisy images by neuro-fuzzy processing. *International journal of Electronics and Communication*, **61 (2)**, 82–89.

Yuksel, M. and Yildrim, M., 2004. A simple neuro-fuzzy edge detector for digital images corrupted by impulse noise. *International Journal of Electronics and Communications*, **58 (1)**, 72–75.

Zadeh, L., 1965. Fuzzy sets. *Information and Control*, **8**, 338–353.

Zadeh, L., 1977. Fuzzy sets and their applications to classification and clustering. Academic Press, New York, pp. 251–299.

Zadeh, L., 2008. Is there a need for fuzzy logic? *Information Sciences*, **178 (13)**, 2751–2779.

Zeng, K. and An, Z., 2007. Multi-face detection based on down sampling and modified subtractive clustering for color images. *Journal of Zhejiang University Science*, **8 (1)**, 72–78.

Zitova, B. and Flusser, J., 2003. Image registration methods. *Image Vision Computing*, **21 (11)**, 997–1000.