Noise Resilient Image Segmentation and Classification Methods with Applications in Biomedical and Semiconductor Images

by

Asaad F. Said

A Dissertation Presented in Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy

ARIZONA STATE UNIVERSITY

December 2010

Noise Resilient Image Segmentation and Classification Methods with Applications in

Biomedical and Semiconductor Images

by

Asaad F. Said

has been approved

October 2010

Graduate Supervisory Committee:

Lina J. Karam, Chair
Chaitali Chakrabarti
Cihan Tepedelenlioglu
Nital Patel

ACCEPTED BY THE GRADUATE COLLEGE

ABSTRACT

Thousands of high-resolution images are generated each day. Segmenting, classifying, and analyzing the contents of these images are the key steps in image understanding. This thesis focuses on image segmentation and classification and its applications in synthetic, texture, natural, biomedical, and industrial images.

A robust level-set-based multi-region and texture image segmentation approach is proposed in this thesis to tackle most of the challenges in the existing multi-region segmentation methods, including computational complexity and sensitivity to initialization.

Medical image analysis helps in understanding biological processes and disease pathologies. In this thesis, two cell evolution analysis schemes are proposed for cell cluster extraction in order to analyze cell migration, cell proliferation, and cell dispersion in different cancer cell images. The proposed schemes accurately segment both the cell cluster area and the individual cells inside and outside the cell cluster area. The method is currently used by different cell biology labs to study the behavior of cancer cells, which helps in drug discovery.

Defects can cause failure to motherboards, processors, and semiconductor units. An automatic defect detection and classification methodology is very desirable in many industrial applications. This helps in producing consistent results, facilitating the processing, speeding up the processing time, and reducing the cost. In this thesis, three defect detection and classification schemes are proposed to automatically detect and classify different defects related to semiconductor unit images. The first proposed defect detection scheme is used to detect and classify the solder balls in the processor sockets as either defective (Non-Wet) or non-defective. The method produces a 96% classification rate and saves 89% of the time used by the operator. The second proposed defect detection scheme is used for detecting and measuring voids inside solder balls of

different boards and products. The third proposed defect detection scheme is used to detect different defects in the die area of semiconductor unit images such as cracks, scratches, foreign materials, fingerprints, and stains. The three proposed defect detection schemes give high accuracy and are inexpensive to implement compared to the existing high cost state-of-the-art machines.

ACKNOWLEDGEMENTS

I owe my gratitude to all those people who have made this dissertation possible and because of whom my graduate experience has been one that I will cherish forever. My deepest gratitude is to my advisor Dr. Lina Karam for her support, patience, stimulating suggestions, and encouragement throughout my graduate studies. Her technical and editorial advice was essential to the completion of this dissertation and has taught me innumerable lessons and insights on the workings of academic research in general.

Bonnie Bennett at Intel Corporations has provided me with necessary assistance and has been always there to listen and give advice. I am deeply grateful to her for giving me the opportunity to join the Pathfinding group at Intel and work on real industrial research problems. I also would like to thank Jeff Pettinato who is the manager of the Pathfinding group at Intel for his continuous encouragement and strong support during the last two years of my PhD research.

Most importantly, none of this would have been possible without the love, care, and patience of my family back in Egypt. My family has been a constant source of love, concern, support and strength all these years. I would like to express my heart-felt gratitude to my family.

Many friends have helped me stay sane through these difficult years. Their support and care helped me overcome setbacks and stay focused on my graduate study. I greatly value their friendship and I deeply appreciate their belief in me.

Finally, I appreciate the financial supporters from the Egyptian Ministry of High Education, Arizona State University, and Intel Corporations who funded part of the research discussed in this dissertation.

TABLE OF CONTENTS

Page

LIST OF TABLES X

LIST OF FIGURES XI

CHAPTER 1: INTRODUCTION 1

1.1. Problem Statement........ 2

1.2. Contributions........ 5

1.3. Thesis organization........ 7

CHAPTER 2: MULTI-REGION LEVEL-SET-BASED SEGMENTATION 8

2.1. Introduction 8

2.2. Existing Multi-region Segmentation Methods........ 10

2.2.1. Multi-region level-set-based hierarchical segmentation........ 10

2.2.2. Multiphase level-set framework for image segmentation 13

2.2.3. Multi-region level-set-based segmentation using constraints........ 17

2.3. Proposed Multi-Region and Texture Segmentation Based on Constrained Level-set Evolution Functions........ 19

2.4. Proposed Multi-region Texture Segmentation........ 22

2.5. Simulation Results........ 23

2.6. Summary........ 29

CHAPTER 3: CELL EVOLUTION ANALYSIS SCHEME BASED ON LEVEL-SET FRAMEWORKS 30

3.1. Introduction........ 30

3.2. Existing Problem and Objectives........ 32

3.3. Review of the Existing Methods........ 33

3.4. Cell Evolution Analysis Scheme........ 35

3.4.1. Cell migration analysis 36

Chapter Page

3.4.2. Cell proliferation analysis ... 37

3.4.3. Cell dispersion analysis ... 37

3.5. Cell Migration Scheme Based on Piecewise ROI Segmentation ... 38

3.5.1. Weighted level-set segmentation method ... 39

3.5.1.1. Simulation results using the weighted level-set ROI segmentation method ... 42

3.5.2. Two-step level-set segmentation method ... 44

3.5.2.1. Simulation results of using the two-step ROI segmentation method ... 44

3.6. Noise-Resilient and Robust Cell Migration using Texture-based ROI Segmentation ... 46

3.6.1. Outer-circle segmentation ... 48

3.6.2. Structure tensor vector ... 49

3.6.3. À Trous Wavelet filtering ... 49

3.6.4. Adaptive statistical level-set segmentation ... 51

3.6.5. Simulation results for the texture-based ROI segmentation ... 52

3.7. Cell Proliferation and Dispersion Analysis ... 55

3.7.1. Proposed individual cell counting and segmentation method ... 56

3.7.2. Cell dispersion analysis ... 60

3.8. Simulation Results ... 63

3.9. Summary ... 65

CHAPTER 4: AUTOMATED DEFECT DETECTION AND CLASSIFICATION IN SEMICONDUCTOR UNIT IMAGES ... 67

4.1. Introduction ... 67

4.2. Automated Non-Wet Defect Detection in Solder Joints ... 68

4.2.1. Problem statement ... 70

Chapter Page

4.2.2. Multi-View x-ray imaging set-up 75

4.2.3. Proposed automatic defect classification scheme for solder joints 77

4.2.3.1. ROI segmentation 79

4.2.3.2. Joint centroid computation and alignment 82

4.2.3.3. Centroid-based classification of good and suspect joints 83

4.2.4. Automatic joint mapping 87

4.2.4.1. Automatic image alignment and mapping 89

4.2.5. Performance results and statistics with ground truth data 92

4.2.6. Summary of non-wet detection 95

4.3. Robust Automated Void Detection in Solder Balls and Joints 96

4.3.1. Problem statement 97

4.3.2. Automated void detection method 102

4.3.2.1. Individual solder ball segmentation 103

4.3.2.2. Candidate void detection 105

4.3.2.3. Feature extraction 106

4.3.2.4. Void classification 107

4.3.3. Simulation results 107

4.3.3.1. Comparison with the data obtained from 2D x-ray embedded algorithm 108

4.3.3.2. Comparison with the data obtained from 2D X-ray images by trained operator 109

4.3.3.3. Comparison with data obtained from 3D CT scan by trained operator 110

4.3.4. Void detection in the presence of vias and other artifacts in solder balls ... 113

4.3.4.1. Via extraction 114

Chapter Page

4.3.4.2. Void detection in the presence of via and via reflection 116

4.3.5. Simulation results for improved void detection method............................ 120

4.3.6. Summary of void detection... 121

4.4. Defect Detection and Classification in the Die Area of Semiconductor Units 123

4.4.1. Preprocessing step 125

4.4.2. ROI segmentation.. 127

4.4.3. Defect detection... 129

4.4.4. Feature extraction 130

4.4.5. Defect classification 132

4.4.5.1. Reference-based classification 133

4.4.5.2. Reference-free classification 133

4.4.6. Simulation results 134

4.5. Summary…………………………………………………………………….137

CHAPTER 5: CONCLUSION 138

5.1. Contributions……………………………………………………………...138

5.2. Future Work……………………………………………………………….140

REFERENCES 142

LIST OF TABLES

Table Page

Table 3.1: Comparison between manual and proposed method………………………..64

Table 4.1: Non-Wet Detection Using the Proposed Algorithm with Automatic Mapping…………………………………………………………………......91

Table 4.2: Non-Wet Joints Obtained from the Dye and Pull Test and the Proposed Algorithm…………………………………………………………………...93

LIST OF FIGURES

Figure Page

2.1 Multi-region image segmentation using level-set-based hierarchical segmentation ………………………………………………………………………. 12

2.2. Representations of the level-set functions for different phases ……………… 14

2.3 Illustration of multiphase level-set framework for image segmentation using two level-set functions to separate between 4 classes ………………………… 15

2.4 Image segmentation using multiphase level-set framework of Chan and Vese at different initialization …………………………………..…………………… 16

2.5 Multi-region segmentation for an image with 6 regions using the proposed method……………………………………………………………………… 24

2.6 Multi-region segmentation for an image with 6 regions using [89]…................. 24

2.7 Texture image segmentation using the proposed multi-region segmentation method ……………………………………………………………………... 25

2.8 Texture segmentation results using the proposed method with different initializations.. ……………………………………………………………….. 26

2.9 Texture image segmentation with 8 regions using the proposed method with different initializations ………………………………………………………. 26

2.10 Brain image segmentation using the proposed method ………………............... 27

2.11 Natural texture image segmentation with 2 regions using the proposed method 28

2.12 Natural image segmentation with 4 different regions using the proposed method 28

2.13 Natural image segmentation with 4 different regions using the proposed method 29

3.1 Sample image of the cell population in the bladder cancer cell image at initial time point 31

Figure | Page

3.2 Sample image of the cell population in the bladder cancer cell image after 24 hours ………. 31

3.3 Block diagram of the proposed CEA system ………. 35

3.4 Block diagram of the proposed CEA scheme based on piecewise ROI segmentation ………. 38

3.5 ROI segmentation using the proposed weighted level-set method for a bladder cancer cell image at T16 and T40 ………. 43

3.6 ROI segmentation using the proposed two-step method for a bladder cancer cell image ………. 45

3.7 An example of artifacts in cancer cell images ………. 46

3.8 Block diagram of the texture-based ROI segmentation ………. 47

3.9 Cell cluster segmentation using the proposed texture-based [78] and piecewise-based methods [79] ………. 53

3.10 Examples of ROI segmentation for different types of bladder cancer cell images using the proposed texture-based ROI segmentation ………. 54

3.11 Examples of ROI segmentation for T98 Glioblastoma cancer cell images using the proposed texture-based ROI segmentation ………. 55

3.12 Cell migration rates for 30 cancer cell images (bladder and Glioblastoma cancer cells) at two different time points ………. 55

3.13 Intensities of the individual cells and background are very close in values ….. 56

3.14 Segmentation masks and histograms for the ROI and background of a bladder cancer cell image ………. 57

3.15 Process for individual cell segmentation and counting for the bladder cancer cell image shown in Fig. 3.14(a) ………. 59

Figure		Page

3.16 Example 1: Texture-based ROI segmentation followed by individual cell segmentation for a cancer cell image 60

3.17 Example 2: Texture-based ROI segmentation followed by individual cell segmentation for a cancer cell image ……………………………………….. 61

3.18 Cell dispersion analysis …………………………………………..………. 62

3.19 Cell dispersion for 30 bladder cancer cell images after 16 hours.…………….. 63

3.20 Cell migration analysis (*um/hr*) using manual versus proposed method for different cancer cell images with different treatments ………………………… 65

4.1 Examples of defective solder joints (non-wet) ………………………………… 69

4.2 Examples of non-defective solder joints ……………………………………… 69

4.3 Image acquisition using 2D x-ray machine ……………………………………. 74

4.4 Rotation motion of the x-ray stage in the X-Y plane during inspection ……… 74

4.5 Sample images obtained at different oblique angles using the x-ray machine. Dotted circles are used to highlight non-wet joints …………………………… 76

4.6 Block diagram of the proposed automatic defect classification scheme for non-wet joints in processor socket ………………………………………………… 78

4.7 Thresholding using histogram analysis ………………………………………. 79

4.8 ROI segmentation of candidate solder joints ………………………………… 81

4.9 Centroids and directional clustering: classification of centroids into different groups based on alignment along two directions ……………………………… 82

4.10 Detecting suspect joints using the proposed reference-free classification method ……………………………………………………………………. 85

4.11 Examples of non-wet solder joints detection using the proposed algorithm ….. 86

4.12 Extracting joints' centroids in each reference image …………………………... 87

Figure Page

4.13 Automatic mapping scheme ………………………………………………… 87

4.14 Misalignment between test and reference images …………………………….. 90

4.15 Image alignment between test and reference images ………………………….. 90

4.16 Results of the proposed algorithm with automatic mapping showing different-view images in which the same non-wet joint (AU43) was detected more than once. The detected non-wet joint is marked with a spot in its center ………… 92

4.17 Examples showing the detection of the neighbor joints to the missed non-wet joints (A14) in different images using the proposed algorithm ……………….. 94

4.18 Solder ball shape in 3D and the distribution of voids in both 3D and 2D images ……………………………………………………………………... 98

4.19 Example of 2D x-ray images that show voids inside solder balls in different product lines ………………………………………………………………… 99

4.20 Examples of some Challenges in the 2D x-ray images ……………………….. 100

4.21 Results using 2D x-ray machine with an embedded void detection algorithm 101

4.22 Block diagram of the proposed method for void detection inside solder balls 102

4.23 Steps of the proposed void detection method ………………………………… 103

4.24 Comparison between existing void detection algorithm in 2D x-ray machine and proposed void detection algorithm in product line B ……………………. 108

4.25 Manual void percentage calculation ……………………………….……….. 109

4.26 2D x-ray images versus 3D CT scan images ….………………………………. 110

4.27 Cumulative voiding comparison between proposed algorithm and manual 3D CT scan data ………………………………………………………………… 110

Figure Page

4.28 Comparison between the results obtained by the proposed method from 2D x-ray images and the results obtained by trained operator from the 3D CT scan images ………………………………………………………………………. 111

4.29 Limitations of processing the 2D x-ray images ………………………………. 111

4.30 Proposed algorithm (Auto) versus engineer: algorithm performance (Auto) compares well to the experienced engineer (CNTRL) ………………………… 112

4.31 More challenges related to 2D x-ray images…………………………………… 113

4.32 Procedures that show the steps of the proposed method for robust void detection ………………………………………………………………………. 115

4.33 Reconstruction of voids obscured by via …………………………………….. 120

4.34 Matching results between the proposed method's void percentage result and result obtained by experienced operators for two different x-ray Tools ……… 121

4.35 Repeatability results of the proposed method for one of the tested product captured by two different x-ray tools ………………………………………….. 122

4.36 Image acquisition using line scan camera……………………………………. 123

4.37 Unit image with different region of interests ………………..………………… 123

4.38 Samples of defects in the die area of the semiconductor units ………………. 123

4.39 Block diagram of the proposed automatic defect classification scheme in unit images ………………………………………………………………………. 125

4.40 Light enhancement using nonlinear mapping ………………………………… 126

4.41 Image alignment using fiducial marks ………………………………………… 126

4.42 ROI segmentation using histogram analysis ………………………………….. 128

4.43 Edge Detection with closing contours to a portion inside the die area ………. 130

4.44 Defect detection using the proposed method …………………………………. 130

Figure Page

4.45 Shape-based feature extraction ……………………………………………………... 131

4.46 Procedure for crack detections …………………………………………………..... 134

4.47 Examples showing the performance of the proposed reference-free classification method in classifying cracks inside the die area of semiconductor units ……………………..…………………………………….. 136

CHAPTER 1: INTRODUCTION

Analyzing image content has become a vital subject in several applications. Computer vision is one of the most important applications based on image content. Computer vision is divided into many subfields including but not limited to image processing, pattern recognition, graph theory, statistical learning. However, the objectives of these areas are as varied as the detection and recognition of objects in images, registration of different views of the same scene, tracking of objects through image sequences, searching for images by their content and so on. Most of these applications require automatic extraction of meaningful information. Thus, the range of computer vision applications is very large and methods carried out can be very different, inspired from physics, biology, statistics theory, functional analysis, and other disciplines.

Medical imaging has also become an important application of image analysis by helping doctors in the detection and diagnosis of diseases and assisting them during the operating phases. Industrial imaging of different units or components is an essential process in many fabs for automation and helps in improving the accuracy and product quality, in increasing the productivity, and in decreasing the cost. Other applications are coming to light with the fast development of communications and entertainment which need more and more tools to compress, process, and manipulate multimedia content. The large number of images and their very high resolutions make manual analysis obsolete and, thus, automatic algorithms need to be developed.

Among all image processing tools, segmentation is probably one of the most important since it is a necessary step in numerous applications. Loosely speaking, it consists in partitioning an image into regions of interest. There is no general definition of a region of interest since it may depend on the type of image, and on the considered application. The way the human brain defines the regions of interest depends on the way we look at the problem and this may vary from person to person based on different

applications. The regions of interest can be defined as homogeneous regions based on given specifications or as regions that fit some prior knowledge. For example, in natural images, one will try to mimic the human visual system (HVS) to obtain results equivalent to the ones that would be given by a human. Given the numerous varieties of natural images, trying to mimic the HVS is in general quite difficult. Image intensity, shape, color and texture are probably the most prominent features that can be used to mimic the HVS but, when analyzing a scene, in addition to the image content, a human makes use of prior knowledge learned from his/her own experience or inherited from the species.

Another problem is the segmentation of medical images. In this case, the objective is not to copy the visual system but to extract real structures/organs from a given modality of acquisition. Hence, understanding and modeling the acquisition process is a key step to extract information from acquired images. Despite all advances in medical image acquisition, many important structures remain hardly visible. All information that can be obtained from prior knowledge is of great interest and it has to be used to help the segmentation process.

Image analysis is widely used in several areas. Industrial imaging is one of the hot areas in image analysis. Having a reliable defect detection and classification in industrial imaging helps in improving the accuracy to obtain reliable and consistent results in addition to decreasing the cost and the time consumed using inaccurate manual operations.

1.1. Problem Statement

A wide range of methods have been proposed to tackle the problems posed by image segmentation. They include direct thresholding based on image gray-level [1-6], grouping from edge detectors results [7-12], classical clusterings on pixel values (k-means, Gaussian mixture models) [13-18], graph-cuts [19-23], Markov random fields [24-29] and others [30, 31]. In this thesis, we consider applications related to multi-region and

texture image segmentation, cell distribution and tracking for cell migration analysis, and defect detection and classification in semiconductor units images.

One of the important segmentation approaches is a geometric approach that is based on front evolution techniques (level-set). For multi-region, texture, and biomedical image segmentation, the choice of a geometric approach has been made for several reasons. The segmentation problem is by definition a geometric one since geometric structures have to be extracted from the 2D image domain. Optimizing with respect to contours or shapes, we consider an optimization space that permits a straightforward formulation of partitioning problems. Geometric constraints can then be naturally expressed to integrate properties of object borders. Simple spatial regularizations can be introduced but more complex geometric knowledge can be explored. Most of the level-set image segmentation methods in this thesis are derived from Mumford-Shah [32] and Bayesian formulation methods [33]. Segmentation based on partial differential equation (PDE) techniques such as level-sets, which offer powerful mathematical tools, helps to well represent the segmentation problem. Level-set techniques, also known as implicit active contours, have been the subject of active research in the last few years. The use of level-set image segmentation techniques is being explored in a variety of applications including medical image segmentation in two- and three-dimensions [34-42], defect detection in industrial applications [43-54], motion analysis [55-60], computer vision [61-67], image denoising [68-71], and image inpainting [72-76]. Most of the classical level-set-based segmentation methods focus mainly on segmenting two regions. There are few related works on multi-region segmentation using level-set functions. Some of these methods use computationally expensive and complex models to segment multiple regions. In addition, some of these models are sensitive to initialization which may produce inaccurate segmentation results. This problem is discussed and tackled in detail in Chapter 2 of this thesis.

Medical image segmentation is one of the most interesting and challenging applications today in biomedical image analysis. Medical images may contain many details such as intensity variation, texture, noise and other artifacts and this makes the segmentation of medical images challenging. So, segmentation of a specific type of medical images, using a certain method, may not work well for another type of medical image. An application of medical image analysis is the migration and proliferation behavior of different types of cancer cell images. The cell images suffer from a lot of challenges including heavily populated (high cell density), tiny cells, touching and overlapping cells, presence of noise and artifacts in the images, and exhibit very low contrast as the intensity values of cell boundaries and background are very close. Accurate segmentation of the cell clustering region is an essential process to compute the migration and proliferation behavior of the cells. Studying the migration and proliferation behavior of cells contributes to the understanding of biological processes and disease pathologies such as cancer, angiogenesis, vascular stenosis and arthritis. The cell evolution analysis of cancer cell images is discussed in Chapter 3 of this thesis.

Image-based classification is perhaps one of the most important parts of digital image analysis and is used in many real applications. Image-based classification is the process of grouping image pixels into categories or classes to produce a thematic representation. In most of the classification schemes, segmentation is usually used as the first step to extract objects in an image before classifying them. Segmentation is the process of extracting different objects in the image without knowing what each object represent, while classification is the process of distinguishing between these objects and assigning each object to its related class or category. Image classification has a lot of applications including but not limited to face identification for human recognition, defect classification in industrial applications, distinguishing cancerous and non-cancerous cells in biomedical images, remote sensing and satellite image classification, and computer

vision. In Chapter 4, we investigate the challenges in detecting and classifying different types of defects in semiconductor unit images. Three different types of defect detection and classification applications are studied in Chapter 4: (i) non-wet solder joints in processor socket, (ii) voids in solder balls and solder joints, and (iii) several defect detection and classification in the die area. The existence of non-wet solder joints in PCB sockets can cause boards failures and it is necessary to inspect theses sockets to locate any possible defective joints. Also voids in solder balls can cause board failures and the detection and assessment of voids in solder balls can help in reducing board yield issues caused by incorrect scrapping and rework. 2D or advanced x-ray machines are used to image solder joints and make solder joints visible to be examined by the operator who determines manually if each individual joint is defective or not, which is a very time consuming process. Different defect classification challenges are presented in industrial images. Some of these challenges and their solutions are discussed in Chapter 4 of this thesis.

1.2. Contributions

In Chapter 2, a multi-region and texture image segmentation method based on a level-set framework is proposed [77]. The proposed method is less sensitive to initializations and exhibits faster convergence as compared to existing multi-region level-set segmentation schemes. Simulation results using synthetic, texture, medical, and natural images are presented to show the robustness of the proposed method in segmenting multi-region images with any number of regions and with different initializations.

In Chapter 3, two Cell Evolution Analysis (CEA) schemes are proposed for analyzing cancer cell images [78-81]. The Region of Interest (ROI), which consists of cell clusters, is segmented based on two different segmentation methods: piecewise-based and texture-based image segmentation. Cell proliferation and cell dispersion analysis are also

discussed in Chapter 3. The developed Cell Evolution Analysis system was tested in two different labs from two leading companies in which researchers are working in cell migration analysis and drug discovery. The reports [82] from these two companies have strong recommendations and show that the developed CEA system produces accurate results compared to manual results by expert operators. In addition, it is fully automated, which helps in saving time and cost. The proposed method is included in a filed US patent [80, 83].

In Chapter 4, three defect detection and classification methods are proposed. Two automated defect detection and classification methods are proposed to automatically detect and classify the defects that are related to solder joints or solder balls [50-54]. The third proposed method is related to defect detection and classification on the die area of the semiconductor unit images. The two defects related to solder joints are: (i).“Non-Wet” defects which occur in solder joints in processor sockets and can cause motherboard failures [50, 51, 54], and (ii).“Void” defects which occur in solder joints and solder balls and can cause incorrect scrapping and rework in addition to motherboard failures [50, 52, 53]. The two proposed methods are being used at Intel Corporation labs for testing semiconductor units that are undergoing assembly. The proposed methods are also inexpensive to implement and are based on low cost 2D x-ray machine. The statistics of the proposed methods [50-54] show high accuracy compared to the results obtained by using state-of-the-art x-ray machines which are 5x to 10x the price of the 2D x-ray machine that was used to produce the images for the proposed algorithms.

The automatic and accurate detection and classification of several defects in the die area are important in manufacturing and help in improving the productivity, saving the time and cost, and reducing the number of return units. In Chapter 4, a scheme for detecting several defects in the die area is proposed. The proposed method successively segments the die area with 100% accuracy based on the statistics collected from over than

1000 tested images. Crack defect is one of the toughest defects to detect due to the challenges related to fading, lighting, and thickness. However, our proposed method gives a 99% detection rate for crack defects, which is considered a very high detection rate, and is inexpensive to implement compared to some of the state-of-the-art machines in the market.

1.3. Thesis organization

This thesis is organized as follows. Chapter 2 presents different methods for multi-region and texture segmentation using a level-set based framework. Chapter 3 presents methods for cell evolution analysis for cell cancer images. Chapter 4 presents automated defect detection and classification methods in semiconductor unit images. Finally, Chapter 5 summarizes the contributions of this thesis and discusses possible future research directions.

CHAPTER 2: MULTI-REGION LEVEL-SET-BASED SEGMENTATION

2.1. Introduction

Image segmentation is one of the fundamental steps in computer vision and is widely used in several image processing applications. The main purpose of image segmentation is to partition an input image into a number of meaningful and non-overlapped regions based on some features that help in distinguishing between different image regions. The level-set segmentation framework encompasses powerful mathematical tools for capturing and tracking topological changes in images. It yields a convenient representation of regions and their boundaries without the need for sophisticated data structures. Most of the classical level-set-based segmentation methods are mainly applied to segment two regions (object and background). There are relatively only few works on multi-region segmentation using level-set functions [84-99]. In [84-87], hierarchical segmentation schemes were presented in which the input image is segmented into multiple regions by using only one level-set function. In these schemes [84-87], the input image is initially divided into two subimages by using one level-set function. Then, each resulting subimage is checked for further segmentation into two subimages using another level-set function. The segmentation process is repeated until all regions are segmented. However, there are issues that arise when using hierarchical segmentation schemes such as identifying which regions need to be further segmented and which regions need to merged, difficulties when dealing with non rectangular/square images, and the sensitivity of the segmentation in case of neighboring regions with close intensity values. In [93], a multiphase level-set-based segmentation method is proposed by modifying the two-phase segmentation model of [100] and by using k level-set functions to segment 2^k regions. However, the model of [93] introduces empty regions if the number of regions is not a power of 2. In [94], the multi-phase segmentation method of [93] was modified by adding an inverse scale information term as in [101] to balance

the scale of the features of the images among the phases (regions). The inverse scale information can be viewed as a function of change in image intensity where the scale is inversely proportional to the image intensity. An inverse scale threshold can be used to eliminate some features of the image based on the variation of the image intensity. Although the method of [94] performs well in the case of piecewise images, it introduces additional incorrectly segmented phases in the case of non piecewise-constant images. In addition, this method is sensitive to noise and requires a denoising procedure such as Total Variation (TV) before applying the segmentation model of [94]. In [88-92], multi-region segmentation methods were proposed based on constraints that help in ensuring disjoint regions. The constrained conditions in [90-92] are applied to the energy function. However, the energy function minimization and the estimation of the parameters in [90-92] require a high computational complexity and are considered among the main difficulties of those methods. Brox and Weickert [88, 89] proposed a multi-region segmentation scheme which integrates a constraint in the curve evolution functions to ensure a balanced competition between regions. However, the methods of [88, 89] are computationally intensive and are sensitive to the initialization of the level-set functions.

One goal of this work is to develop a level-set-based multi-region segmentation method that is more efficient and less sensitive to initialization problems than the existing level-set-based multi-region segmentation schemes and that works for any number of regions. In this work, the curve evolution constraints that are used in the multi-region segmentation method of [88, 89] are improved upon using a scheme based on a modified Chan and Vese model [100]. In the proposed multi-region segmentation method, each region in the input image is represented by one level-set function. A competition factor is used in the proposed method to keep a balance between regions and to make the evolving curves move faster without getting stuck at undesired points. A regularization term is introduced to impose smoothness and regularity for each region. The proposed multi-

region method was successfully applied to various images with multiple regions, including synthetic, medical, texture, and natural images.

In the following, we will first give a review of the existing multi-region level-set based segmentation methods, then we will discuss our proposed multi-region and texture-based segmentation methods. Finally, results are presented to illustrate the performance of the proposed methods.

2.2. Existing Multi-region Segmentation Methods

2.2.1. Multi-region level-set-based hierarchical segmentation

In [100], Chan and Vese proposed a two-phase level-set segmentation model which is derived from the Mumford-Shah model [102]. Their model is defined by the following energy functional:

$$F(c_1, c_2, \phi) = \int_{\Omega} \left(|I_0 - c_1|^2 H(\phi) + |I_0 - c_2|^2 \left(1 - H(\phi)\right) + v\, |\nabla H(\phi)| \right) dxdy \quad (2.1)$$

where I_0 is the input image with a domain Ω, ϕ is the level-set function which is used to distinguish between two regions ($\phi > 0$ and $\phi < 0$), $H(\phi)$ represents the regularized Heaviside function which is equal to 1 when $\phi \geq 0$ and 0 otherwise, $\nabla H(\phi)$ is the gradient of the Heaviside function $H(\phi)$, and c_1 and c_2 represent the mean values of the two regions in I_0 corresponding to $\phi > 0$ and $\phi < 0$, respectively. In (2.1), v is a positive parameter that can be adjusted to control the smoothness and the total length of the contours ($\phi(x, y) = 0$). A small value of v is chosen if the interest is to detect many small objects, otherwise a high value is chosen for detecting few large objects. The energy function (2.1) is minimized using the following associated Euler–Lagrange equation [103] to get the curve evolution:

$$\frac{\partial \phi}{\partial t} = -\,\delta(\phi) \left\{ (I_0 - c_1)^2 - (I_0 - c_2)^2 - v\ div\left(\frac{\nabla \phi}{|\nabla \phi|} \right) \right\} \quad (2.2a)$$

$$c_1 = \frac{\int_\Omega I_0(x,y)\, H(\phi_1)\ dxdy}{\int_\Omega H(\phi_1)\ dxdy}, \qquad c_2 = \frac{\int_\Omega I_0(x,y)\, \big(1 - H(\phi_1)\big)\ dxdy}{\int_\Omega \big(1 - H(\phi_1)\big)\ dxdy} \tag{2.2b}$$

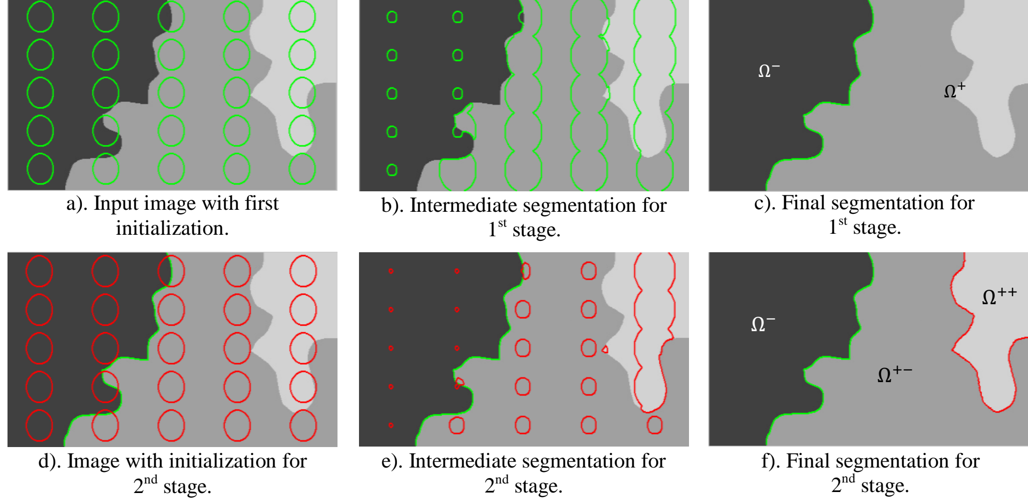
where $\delta(\phi) = \partial H(\phi)/\partial\phi$ represents the Dirac delta function. In (2.2a), div is the divergence operator which is given by $div\ (I) = \nabla . I = (\partial I/\partial x + \partial I/\partial y)$. $div(\nabla\phi/|\nabla\phi|)$ represents the curvature of ϕ. The evolution of ϕ is controlled by (2.2a) to segment the input image I_0 into two regions with two different mean values c_1 and c_2 (2.2b) defined by $(\phi > 0)$ and $(\phi < 0)$, respectively. The evolution function of (2.2a) is used to segment two different regions (object and background). However, the method can not be used to segment an image that has more than two different regions with different intensity values.

Another methodology that uses the level-set framework to segment more than two regions consists of applying hierarchical segmentation using (2.2). In [84-87], hierarchical segmentation schemes were presented in which the input image is segmented into multiple regions by using only one level-set function. In these schemes [84-87], the input image is initially divided into two regions by using one level-set function. Then, each resulting region is checked for further segmentation into two new regions using another level-set function. The segmentation process is repeated until all regions are segmented. An example for segmenting 4 different regions using a hierarchical segmentation can be summarized as follows:

Step 1 – Use one level-set function ϕ to segment the input image I_0 into two regions using the following curve evolution function:

$$\frac{\partial\phi}{\partial t} = \delta(\phi)\left\{ v_1\, div\left(\frac{\nabla\phi}{|\nabla\phi|}\right) - \lambda_1(I_0 - c_1)^2 + \lambda_2(I_0 - c_2)^2 \right\}, \tag{2.3}$$

where c_1 and c_2 are the averages of I_0 inside ϕ $(\phi > 0)$ and outside ϕ $(\phi < 0)$, respectively. The segmentation results of this step will be represented by two regions, Ω^+ and Ω^-, which represent the regions inside and outside ϕ, respectively.



a). Input image with first initialization. b). Intermediate segmentation for 1st stage. c). Final segmentation for 1st stage.

d). Image with initialization for 2nd stage. e). Intermediate segmentation for 2nd stage. f). Final segmentation for 2nd stage.

Fig. 2.1. Multi-region image segmentation using level-set-based hierarchical segmentation.

Step 2 – Check each segmented region Ω^+ and Ω^- in the previous step for further segmentation. Some authors compare the variance inside each region to check which region has different subregions and requires further segmentation, others use different methodologies based on mean, variance, or both, to differentiate between neighboring regions.

Step 3 –If there are more regions inside Ω^+ and Ω^-, apply the same procedure as in Step 1 for both regions Ω^+ and Ω^- of the input image I_0 to get the following regions: Ω^{++}, Ω^{+-}, Ω^{-+} and Ω^{--}. For this purpose, level-set functions ϕ_1 and ϕ_2 are assigned to regions Ω^+ and Ω^-, respectively. The four obtained segmented regions are described as follows:

$$\begin{aligned}
\Omega^{++} &= \{(x,y), \quad \forall\ H(\phi\,(x,y))\, H(\phi_1(x,y)) > 0\}, \\
\Omega^{+-} &= \{(x,y), \quad \forall\ H(\phi\,(x,y))\, (1 - H(\phi_1(x,y))) > 0\}, \\
\Omega^{-+} &= \{(x,y), \quad \forall\ (1 - H(\phi\,(x,y))\, H(\phi_2(x,y)) > 0\}, \\
\Omega^{--} &= \{(x,y), \quad \forall\ (1 - H(\phi\,(x,y)))\, (1 - H(\phi_2(x,y))) > 0\}
\end{aligned} \tag{2.4}$$

Step 4 – Follow the same procedure described in Steps 2 and 3 to find all subregions.

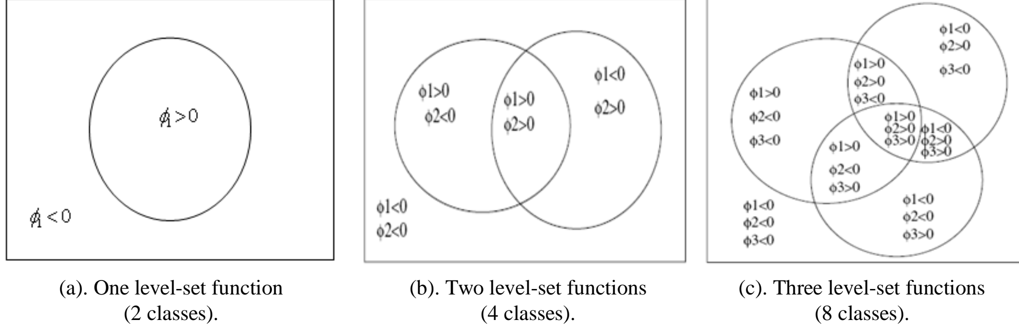
Step 5 – Check the segmented regions for merging to create the final segmentation results.

An example of multi-region level-set-based hierarchical segmentation is given in Fig. 2.1 where Fig. 2.1(a) represents the input image with an initialization for the level-set function ϕ. Figs. 2.1(b) and 2.1(c) represent, respectively, the intermediate and final segmentation results using (2.3) for two regions. Fig. 2.1(d) represents the initialization of the new level-set function ϕ_1. Equation (2.3) is used to segment region Ω^+ $(\phi > 0)$ into two new regions Ω^{++} and Ω^{+-} as shown in Fig. 2.1(f).

The hierarchical multi-region level-set-based segmentation method sounds simple and easy to follow. However, there are a lot of issues that arise when using hierarchical segmentation schemes such as identifying which regions need to be further segmented and which regions need to merged, difficulties when dealing with non rectangular/square images, sensitivity to the initialization, and the sensitivity of the segmentation in case of neighboring regions with close intensity values or similar texture.

2.2.2. Multiphase level-set framework for image segmentation

As indicated in Section 2.2.1, Chan and Vese [100] proposed a two-phase segmentation model using a level-set representation. In the model of [100], one level-set function ϕ is used to separate between two areas as shown in Fig. 2.2(a), where the zero level-set function $(\phi(x, y) = 0)$ is used to distinguish between two different regions $(\phi > 0 \text{ and } \phi < 0)$ with two different mean values. For segmenting more than two regions, Chan and Vese updated the two-phase segmentation model [100] to work based on a multiphase level-set framework [93]. The multiphase level-set-based model [93] requires only k level-set functions to segment 2^k regions (phases). For example, 2 level-set functions are used to segment 4 regions as shown in Fig. 2.2(b), and 3 level-set functions are used to segment 8 regions as shown in Fig. 2.2(c).



(a). One level-set function (2 classes).   (b). Two level-set functions (4 classes).   (c). Three level-set functions (8 classes).

Fig. 2.2. Representations of the level-set functions for different phases.

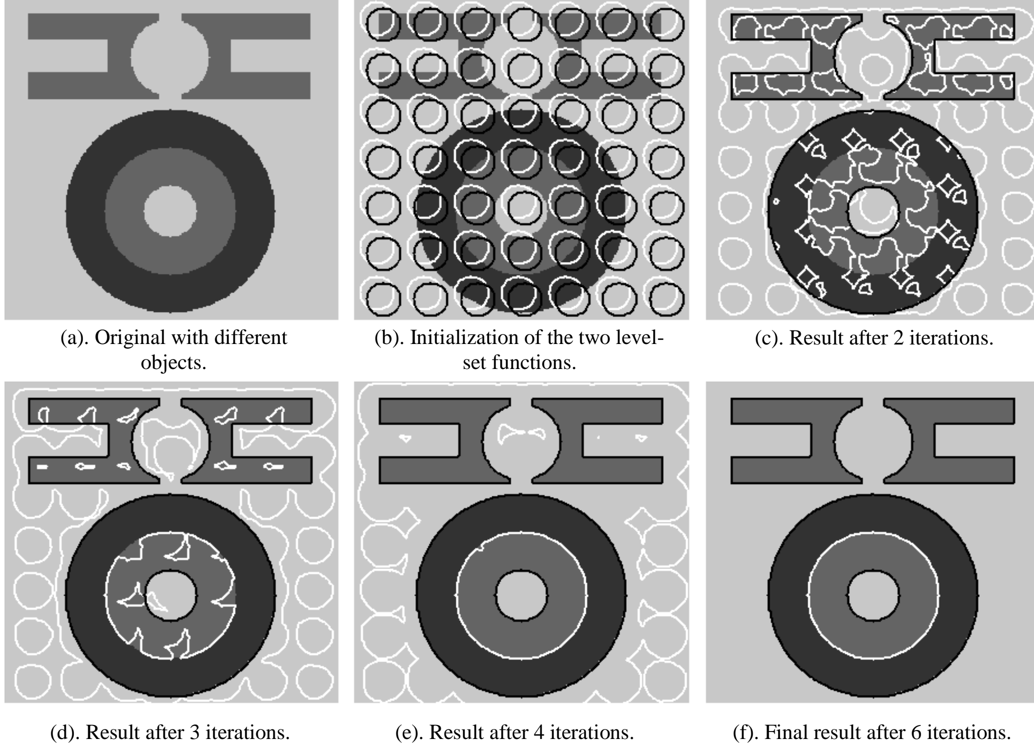
For the purpose of simplification and illustration, let us represent the multiphase level-set model for 4 phases by using 2 level-set functions. The energy function representing the two level-set functions ϕ_1 and ϕ_2 is given by [93]:

$$
\begin{aligned}
F_4(C,\Phi) = & \int_{\Omega} |I_0 - c_{11}|^2 H(\phi_1) H(\phi_2)\ dxdy \\
& + \int_{\Omega} |I_0 - c_{10}|^2 H(\phi_1)\big(1 - H(\phi_2)\big)\ dxdy \\
& + \int_{\Omega} |I_0 - c_{01}|^2 \big(1 - H(\phi_1)\big) H(\phi_2)\ dxdy \\
& + \int_{\Omega} |I_0 - c_{00}|^2 (1 - H(\phi_1))(1 - H(\phi_2))\ dxdy \\
& + v \int_{\Omega} |\nabla H(\phi_1)|\ dxdy + v \int_{\Omega} |\nabla H(\phi_2)|\ dxdy
\end{aligned}
\tag{2.5}
$$

where v is a constant speed, I_0 is the considered image, $\Phi = [\phi_1, \phi_2]$ represents the two level-set functions which are used to define four regions. In (2.5), $C = [c_{11}, c_{10}, c_{01}, c_{00}]$ where $c_{11}, c_{10}, c_{01},$ and c_{00} are constants corresponding to the mean of the image I_0 in the regions $\{\phi_1 > 0, \phi_2 > 0\}, \{\phi_1 > 0, \phi_2 < 0\}$, $\{\phi_1 < 0, \phi_2 > 0\}$, and $\{\phi_1 < 0, \phi_2 < 0\}$, respectively, and given by:

$$
\begin{aligned}
c_{11} &= \frac{\int_{\Omega} I_0(x,y)\ H(\phi_1) H(\phi_2)\ dxdy}{\int_{\Omega}\ H(\phi_1) H(\phi_2)\ dxdy}, \qquad
c_{10} = \frac{\int_{\Omega} I_0(x,y)\ H(\phi_1)\ \big(1 - H(\phi_2)\big)\ dxdy}{\int_{\Omega}\ H(\phi_1)\big(1 - H(\phi_2)\big)\ dxdy}, \\
c_{01} &= \frac{\int_{\Omega} I_0(x,y)\big(1 - H(\phi_1)\big) H(\phi_2) dxdy}{\int_{\Omega}\ \big(1 - H(\phi_1)\big) H(\phi_2)\ dxdy}, \quad
c_{00} = \frac{\int_{\Omega} I_0(x,y)\big(1 - H(\phi_1)\big)\big(1 - H(\phi_2)\big) dxdy}{\int_{\Omega}\ \big(1 - H(\phi_1)\big)\big(1 - H(\phi_2)\big)\ dxdy}
\end{aligned}
\tag{2.6}
$$

The curve evolution functions for the two level-set functions ϕ_1 and ϕ_2 are obtained by



(a). Original with different objects.

(b). Initialization of the two level-set functions.

(c). Result after 2 iterations.

(d). Result after 3 iterations.

(e). Result after 4 iterations.

(f). Final result after 6 iterations.

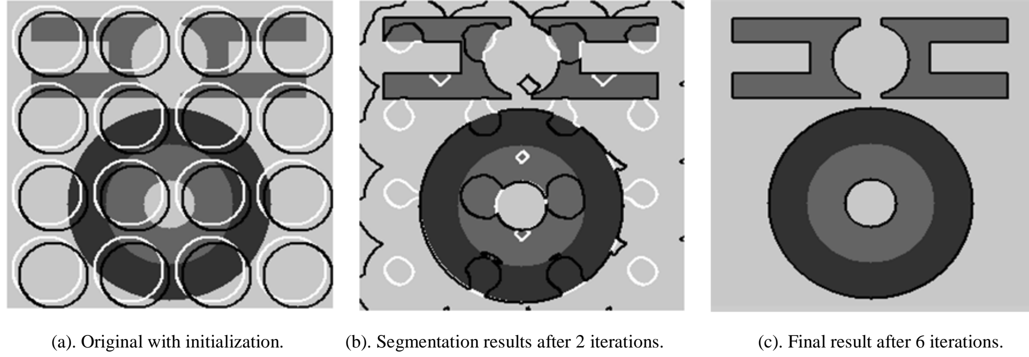
Fig. 2.3. Illustration of the multiphase level-set framework for image segmentation using two level-set functions to separate between 4 classes.

minimizing the energy function of (2.5) with respect to ϕ_1 and ϕ_2 and are given by:

$$\frac{\partial \phi_1}{\partial t} = \delta(\phi_1)\left\{v\, div\left(\frac{\nabla \phi_1}{|\nabla \phi_1|}\right) - ((I_0 - c_{11})^2 - (I_0 - c_{01})^2)H(\phi_2) - ((I_0 - c_{10})^2 - (I_0 - c_{00})^2)\,(1 - H(\phi_2))\right\} \tag{2.7a}$$

$$\frac{\partial \phi_2}{\partial t} = \delta(\phi_2)\left\{v\, div\left(\frac{\nabla \phi_2}{|\nabla \phi_2|}\right) - ((I_0 - c_{11})^2 - (I_0 - c_{10})^2)H(\phi_1) - ((I_0 - c_{01})^2 - (I_0 - c_{00})^2)(1 - H(\phi_1))\right\} \tag{2.7b}$$

Equations (2.7a) and (2.7b) represent the curve evolution for level-set function ϕ_1 and ϕ_2, respectively. Fig. 2.3 gives an example for multiphase level-set segmentation [93] for 4 regions using two level-set functions (2.5). Fig. 2.3(a) displays the original synthetic image, which has different objects with different intensities. By following the same procedures for multiphase image segmentation as explained above, it will be straightforward to segment the given image. Fig. 2.3(b) shows the original image with the



(a). Original with initialization. (b). Segmentation results after 2 iterations. (c). Final result after 6 iterations.

Fig. 2.4. Image segmentation using the multiphase level-set framework of Chan and Vese using different initialization.

initialization for the two level-set functions ϕ_1 and ϕ_2. Figs. 2.3(c)-(e) illustrate the segmentation results after two, three, and four iterations, respectively. Fig. 2.3(f) shows the final segmentation result after 6 iterations. The segmentation results of the multiphase model of Chan and Vese [93] are sensitive to the initialization and the fact that the number of regions is limited to be a power of 2. An example for the initialization problem is shown in Fig. 2.4 where it can be seen that the final result is sensitive to the given initialization. The initialization problem has been discussed by many authors [84, 104] in order to improve the segmentation result; however, the multiphase method of [93] can be trapped in local minima, resulting in incorrect segmentation.

There are several methods that were proposed for multi-region segmentations [93, 95, 96, 105] which are based mainly on the idea of the multiphase model of Chan and Vese [93]. The multi-phase segmentation methods of [93, 95, 96, 105] were used to segment 2^{k} regions by using k level-set functions. However, these methods are sensitive to the selected initialization and are restricted to segment a power of 2 number of regions. If the number of regions is not a power-of-two, empty regions are produced in the segmented image [98]. For example, a 6-phase image requires 3 level-set functions, which produce an 8-phase image even though only a 6-phase image is needed. Another multi-phase segmentation method was proposed in [94]. The method of [94] uses an inverse scale term as in [101] to balance the features of the image among the phases (regions). In [94],

a pixel-wise decision algorithm is used without using the Euler-Lagrange equation of a nonlinear functional. Although the method of [94] performs well in the case of piecewise images, it introduces additional incorrectly segmented phases in the case of non piecewise-constant images. In addition, this method is sensitive to noisy images and requires denoising before segmentation.

2.2.3. Multi-region level-set-based segmentation using constraints

For multi-region segmentation, there are different level-set-based approaches [90, 91, 99, 106] that assign one level-set function to each region. Representing each region by one level-set function is an easy way for direct access to the segmented regions. The energy function of an image with *N* regions can be represented by:

$$F(p_1, \dots, p_N, \phi_1, \dots, \phi_N) = \int_{\Omega} \sum_{i=1}^{N} (-H(\phi_i) \log_e \, p_i + v \, |\nabla H(\phi_i)|) \; dxdy \qquad (2.8)$$

where ϕ_i represents the level-set function which is assigned to region i and p_i represents the probability distribution function inside region i (region corresponding to $\phi_i > 0$). The curve evolution functions of (2.8) are given by:

$$\frac{\partial \phi_i}{\partial t} = \left\{ \log_e p_i + v \;\; div\left(\frac{\nabla \phi_i}{|\nabla \phi_i|}\right) \right\} \delta(\phi_i), \quad \forall \, i = 1,2, \dots, N \qquad (2.9)$$

The $\log_e p_i$ term in (2.9) is always negative and may cause the evolution to shrink without segmenting the input image. Neither (2.8) nor (2.9) has a term that creates a balanced competition between regions. To address this problem, different constraints can be added to either (2.8) or (2.9) to ensure a balance between competing regions in order to keep the regions disjoint, non-overlapping, and to guarantee energy minimization. In [90, 92], a constraint was added to the energy function of (2.8) by using a Lagrangian multiplier as follows:

$$F(p_1, \dots, p_N, \phi_1, \dots, \phi_N) = \int_{\Omega} \sum_{i=1}^{N} (v|\nabla H(\phi_i)| - H(\phi_i) \log_e\, p_i)\, dxdy$$
$$+\beta \int_{\Omega} \left(\sum_{j=1}^{N} H(\phi_j) - 1 \right)^2 dxdy \qquad (2.10)$$

where β is a Lagrange multiplier. However, the minimization of (2.10) is computationally complex. In [91], an artificial coupling force was integrated into (2.10). However, in this latter case, the energy formulation of (2.10) depends on several parameters that need to be selected and tuned. In particular, for a fixed-parameter setting, disjoint regions cannot be ensured for arbitrary images. Brox and Weickert [88, 89] proposed a multi-region segmentation model using a three-step split-and-merge approach. In the first step, they use one level-set function to split the regions of an image in a hierarchical way as explained in Section 2.2.1. These segmented regions are used to initialize a multi-region segmentation stage which is based on the level-set-based minimization scheme of Zhu and Yuille [107]. Brox and Weickert [88, 89] add a constraint to the curve evolution functions in (2.9) as follows:

$$\frac{\partial \phi_i}{\partial t} = \left\{ e_i - \max_{\delta(\phi_j)>0,\ j \neq i} \left(e_j, e_i - 1 \right) \right\} \delta(\phi_i) \qquad (2.11)$$

where

$$e_i = \log_e p_i + v\, div \left(\frac{\nabla \phi_i}{|\nabla \phi_i|} \right), \quad i = 1, 2, \dots, N \qquad (2.12)$$

The *max* operation in (2.11) is used to add a competition between regions in the curve evolution functions. The term $e_i - 1$ in (2.11) is used instead of $e_{j, j \neq i}$ if there are no competing regions in the neighborhood (vacuum regions). In this case, the regions evolve with a small constant speed toward the vacuum. In such a situation, the method of [88, 89] requires a lot of iterations to obtain satisfactory results and it can easily get stuck at some points. In addition, this method is sensitive to the selected initialization of the level-set functions. As an attempt to reduce the sensitivity to initializations, Brox and Weickert

apply, as a preprocessing step, a coarse hierarchical segmentation step. Then, the obtained coarse segmentation masks were used as initialization masks for ϕ_i to get the final segmentation results using (2.11). However, the coarse segmentation step introduces additional computational cost and still leads to slow evolution when the coarse segmentation mask results in non-overlapping regions.

2.3. Proposed Multi-Region and Texture Segmentation Based on Constrained Level-set Evolution Functions

The proposed multi-region level-set segmentation scheme is not limited to a single image component but can be applied to vector-valued images that consist of several components. In particular, the interest here is to provide a level-set-based method that can segment both texture and non-texture regions.

Consider a vector-valued image $\hat{I}$ consisting of L components $I_j(j = 1,2,\ldots,L)$. The proposed energy functional of a vector-valued image $\hat{I}$ containing N regions is given by:

$$F(\hat{C},\widehat{\Phi}) = \sum_{i=1}^{N}\left(\int_{\Omega} w(x,y)\,|\nabla H(\phi_i)|\,dxdy + \lambda\int_{\Omega}\sum_{j=1}^{L}\left|I_j - c_{ij}\right|^2 H(\phi_i)dxdy\right) \quad (2.13)$$

where $\widehat{\Phi}$ is a vector whose elements consist of the level-set functions $(\phi_i, i = 1,\ldots,N)$, c_{ij} represents the mean value inside image I_j when the level-set function ϕ_i is positive, and $\hat{C}$ is a matrix whose entries consist of the mean values $c_{ij}(i = 1,\ldots,N \;\&\; j = 1,\ldots,L)$. The means c_{ij} can be easily obtained by minimizing the energy function of (2.13) using the Euler-Lagrange method [103] and are given by:

$$c_{ij} = \frac{\int_{\Omega} I_j(x,y)\,H(\phi_i(x,y))dxdy}{\int_{\Omega} H(\phi_i(x,y))dxdy}, \qquad \forall\, i = 1,2,\ldots,N, \qquad j = 1,2,\ldots,L. \quad (2.14)$$

In (2.13), $w(x,y)$ is a weighting function and $\int_{\Omega} w(x,y)\,|\nabla H(\phi_i)|\,dxdy$ is used for regularization and represents the weighted arc-length of the level-set function ϕ_i. It was

shown by Kimmel [108] that the weighted arc-length gives better regularization where $w(x,y)$ represents an edge indicator function. A popular choice for $w(x,y)$ is to set $w(x,y) = 1/(1+|\nabla I|^2)$ [109], where ∇I represents the gradient of the input image I. In the proposed method, the following weighting function $w(x,y)$, which is suitable for multidimensional spaces, is adopted [110]:

$$w(x,y) = \frac{1}{|det(\Psi)|} \tag{2.15}$$

where $det(\Psi)$ is the determinant of a 2D matrix Ψ. The matrix Ψ can be represented in terms of the regularized vector-valued image components $\rho[I_j]$ as follows:

$$\Psi = \begin{pmatrix} 1+\sum_{j=1}^{L} \gamma_j \; \rho\left[\left(I_{jx}\right)^2\right] & \sum_{j=1}^{L} \gamma_j \; \rho\left[I_{jx}\, I_{jy}\right] \\ \sum_{j=1}^{L} \gamma_j \, \rho\left[I_{jx}\, I_{jy}\right] & 1+\sum_{j=1}^{L} \gamma_j \, \rho\left[\left(I_{jy}\right)^2\right] \end{pmatrix} \tag{2.16}$$

In (2.16), I_{jx} and I_{jy} are the partial derivatives of I_j in the x and y direction, respectively. The weight parameters γ_j $(j=1,\dots,L)$ are used to ensure the same dynamic range for all the entries of Ψ in (2.16). In (2.16), $\rho[I_j]$ is a regularized version of the jth $(j = 1,\dots,L)$ image component I_j, which can be obtained by using the non-linear diffusion filter of [111] as follows:

$$\frac{\partial I_j}{\partial t} = div\left(g\left(\sum_{k=1}^{L} |\nabla I_k|^2\right) \nabla I_j\right), \qquad \forall j = 1,2,\dots,L \tag{2.17}$$

where $g\left(\sum_{k=1}^{L} |\nabla I_k|^2\right)$ represents the diffusivity function which can be obtained as follows [112]:

$$g\left(\sum_{k=1}^{L} |\nabla I_k|^2\right) = \frac{1}{\left(\sum_{k=1}^{L} |\nabla I_k|^2\right)^p + \epsilon}, \tag{2.18}$$

where $p \in \mathbb{R}$ and $p \geq 1$ and ϵ is a small positive constant added to the denominator to avoid any numerical problems when the gradient gets close to zero. Note that, in (2.17)

and (2.18), all channels are coupled by a joint diffusivity; so, an edge in one channel also inhibits smoothing in the others.

The proposed energy functional (2.13) can be seen as an extension to the multi-region case of the Chan and Vese functional [100], which is used for two-region segmentation. In this work, the curve arc-length term $\int_{\Omega} |\nabla H(\phi_i)| \, dxdy$ in [100] is replaced by a weighted arc-length $\int_{\Omega} w(x,y) \, |\nabla H(\phi_i)| \, dxdy$ for better regularization. It was shown in [110] that using a weighted arc-length [108] for two-region segmentation gives accurate and promising segmentation results compared to the segmentation method of [100]. In addition, the proposed multi-region segmentation method makes use of one level-set function ϕ_i per region, which allows direct access to the segmented regions, instead of using color theory as in [93, 95, 96, 105]. The color theory based methods do not allow easy access and can fail if the number of regions is not a power of 2. The minimization of (2.13) with respect to ϕ_i can be obtained by using the Euler–Lagrange equation [103] as follows:

$$\frac{\partial \phi_i}{\partial t} = \left(\nabla . \left(w(x,y) \frac{\nabla \phi_i}{|\nabla \phi_i|} \right) - \lambda \sum_{j=1}^{L} \left| I_j - c_{ij} \right|^2 \right) \delta(\phi_i) = R_i \, \delta\,(\phi_i), \qquad \forall i = 1,2,\dots,N \tag{2.19}$$

where R_i is given by

$$R_i = \left(w(x,y) \, div \left(\frac{\nabla \phi_i}{|\nabla \phi_i|} \right) + \langle \nabla w(x,y), \frac{\nabla \phi_i}{|\nabla \phi_i|} \rangle - \lambda \sum_{j=1}^{L} \left| I_j - c_{ij} \right|^2 \right), \qquad \forall i = 1,2,\dots,N. \tag{2.20}$$

In (2.20), $div(\nabla \phi_i / |\nabla \phi_i|)$ represents the curvature of the level-set function ϕ_i and is multiplied by the weighting function $w(x,y)$. The second term on the right side of (2.20) represents the dot product between $\nabla w(x,y)$ and $\nabla \phi_i / |\nabla \phi_i|$, which enhances the edges. $\sum_{j=1}^{L} \left| I_j - c_{ij} \right|^2$ represents the data fidelity term with respect to the input vector-valued

image $\hat{I}$. The energy formulation (2.13) and curve evolution (2.19) do not allow a balance between competing regions. To ensure a balanced competition, a constraint should be applied to the curve evolution process. The proposed constrained curve evolution functions are as follows:

$$\frac{\partial \phi_i}{\partial t} = R_i' \ \delta\,(\phi_i)\,, \qquad \forall\, i = 1,2,\dots,N\,, \tag{2.21}$$

where

$$R_i' = \left\{ \alpha\, R_i - (1-\alpha) \max_{\substack{\delta(\phi_j)>0,\ j=1,2,\dots,N \\ j\neq i}} \left(R_i\ , R_j\right) \right\}, \qquad \forall\, i = 1,2,\dots,N. \tag{2.22}$$

In (2.22), R_i is given by (2.20) and α is referred to as a competition factor. The value of α can be set to speed up the evolution process and to prevent the evolving curves from getting stuck at undesired points. If $\alpha = 1$ or 0, the curve evolution functions (2.21) will evolve without segmenting the desired objects because there is no balance between regions. If $\alpha = 0.5$, the curve evolution functions of (2.22) can easily get stuck at some points especially when there is no overlap between evolving curves ϕ_i. Any value of α between 0.3 and 0.5 was found to result in good segmentation results with good computational performance as compared to existing multi-region segmentation results. For values of $\alpha > 0.5$, the first term of (2.22) will be dominant in most of the cases and will create unbalance between competing regions. In order to avoid the generation of small areas (islands), any isolated island with area less than 0.05% of the total area inside each level-set function, is removed. This can be easily performed using simple morphological operators.

2.4. Proposed Multi-region Texture Segmentation

In the case of texture image segmentation, the key is to extract some features which can be used to distinguish between different textures in the input image. In [113-117], the structure tensor vector was used to represent texture features for a wide range of texture

images. In this work, for multi-region texture segmentation, a vector-valued texture image $\hat{I}$, consisting of four components that represent an augmented structure tensor vector, is first computed as follows [113, 115, 117]:

$$\hat{I} = (I, I_x^2, I_y^2, I_x I_y) = (I_1, I_2, I_3, I_4) \qquad (2.23)$$

where I represents the input grayscale image, and I_x and I_y are the first derivatives of I in the x and y directions, respectively. The four components $I_j, (j = 1, \ldots, L,$ where $L = 4)$, of the feature vector $\hat{I}$ in (2.23) are regularized (smoothed) to obtain $\rho[I_j], (j = 1,2,3,4)$, by using a nonlinear diffusion filter for a vector-valued data [111] as given in (2.17). The regularized version $\rho[I_j]$ is obtained from I_j after using (2.17). The curve evolution functions (2.21) and (2.22) are then used to segment the texture regions. In (2.22), R_i $(i = 1, \ldots, N)$ are computed as in (2.20), except that I_j is replaced by the regularized components $\rho[I_j]$ as follows:

$$R_i = \left(w(x,y)\, div\left(\frac{\nabla\phi_i}{|\nabla\phi_i|}\right) + \langle \nabla w(x,y), \frac{\nabla\phi_i}{|\nabla\phi_i|} \rangle - \lambda \sum_{j=1}^{L} \left|\rho[I_j] - c_{ij}\right|^2 \right), \qquad (2.23)$$

$$\forall i = 1,2, \ldots, N$$

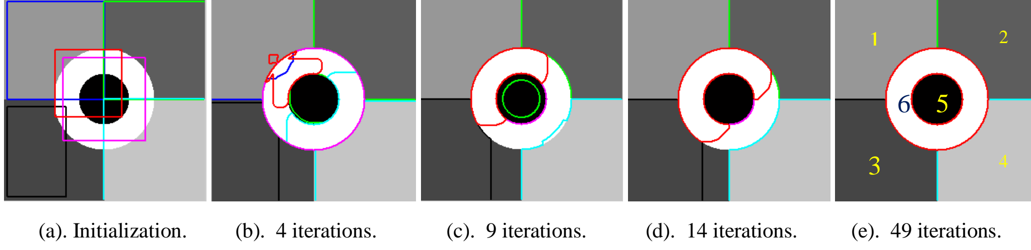
where

$$c_{ij} = \frac{\int_\Omega \rho[I_j(x,y)]\, H\big(\phi_i(x,y)\big) dxdy}{\int_\Omega H\big(\phi_i(x,y)\big) dxdy}, \qquad \forall\ i = 1,2, \ldots, N, \text{and } j = 1,2, \ldots, L. \quad (2.24)$$

2.5. Simulation Results

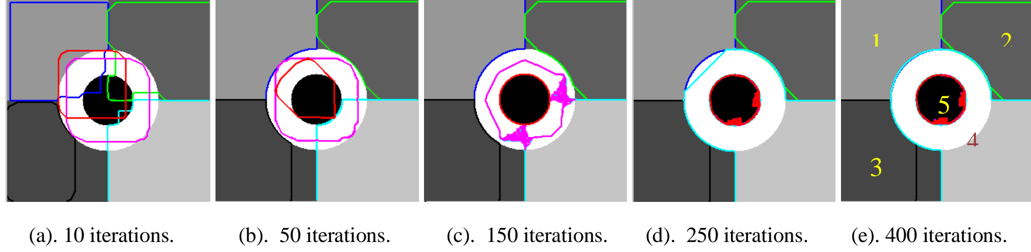
The proposed multi-region segmentation method was successfully applied to segment a variety of images such as synthetic, natural, medical, and texture images. Results are presented in this section to illustrate the performance of the proposed multi-region and texture segmentation methods. In the following examples, the competition factor α is chosen to be 0.45, unless it is mentioned otherwise.

Fig. 2.5 shows segmentation results for a synthetic image using the proposed multi-



(a). Initialization. (b). 4 iterations. (c). 9 iterations. (d). 14 iterations. (e). 49 iterations.

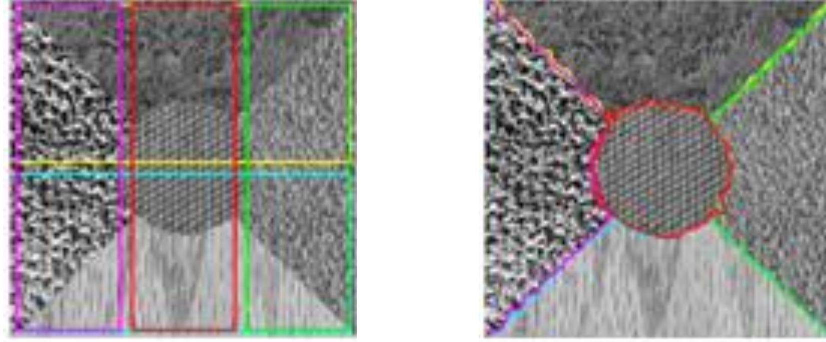
Fig. 2.5. Multi-region segmentation for an image with 6 regions using the proposed method.



(a). 10 iterations. (b). 50 iterations. (c). 150 iterations. (d). 250 iterations. (e). 400 iterations.

Fig. 2.6. Multi-region segmentation for an image with 6 regions using [89].

region segmentation method. Fig. 2.5(a) represents the input synthetic image with an initialization for 6 non-texture regions. The intermediate segmentation results after 4, 9, and 14 iterations are shown in Figs. 2.5(b), (c), and (d), respectively. The final segmentation result is shown in Fig. 2.5(e) after 49 iterations. It can be seen that the proposed method succeeded in segmenting the 6 regions accurately. The competition factor α is chosen to be 0.3 in the previous example. Fig. 2.6 shows the segmentation results for the same image in Fig. 2.5(a) using the multi-region segmentation method of Brox and Weickert [89]. The same initialization was used for comparison between [89] and our proposed method. Figs. 2.6 (a)-(e) show the segmentation results using [89] after 10, 50, 150, 250, and 400 iterations, respectively. It can be clearly seen that the method of [89] failed to segment the 6 regions. Moreover, the method of [89] takes a lot of iterations and it can easily get stuck at some points. It can be seen from Fig. 2.6 that, in regions 2 and 3, the curve evolution functions (ϕ_2 and ϕ_3) evolve very slowly and can easily get stuck at some points. The curves get stuck because there is no overlap between the evolving functions where there is no competition between regions and, therefore, the curves (ϕ_2 and ϕ_3) evolve very slowly due to the small step size value in (2.11).



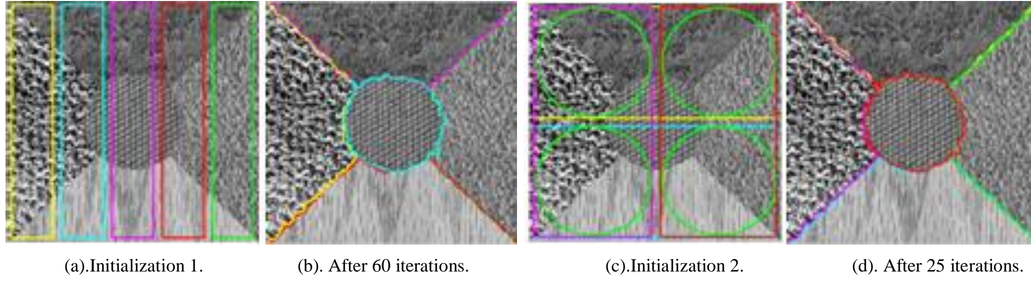
(a). Input image with 5 texture regions and an initialization using 5 level-set functions.

(b). Final segmentation result using the proposed method after 33 iterations.

Fig. 2.7. Texture image segmentation using the proposed multi-region segmentation method.

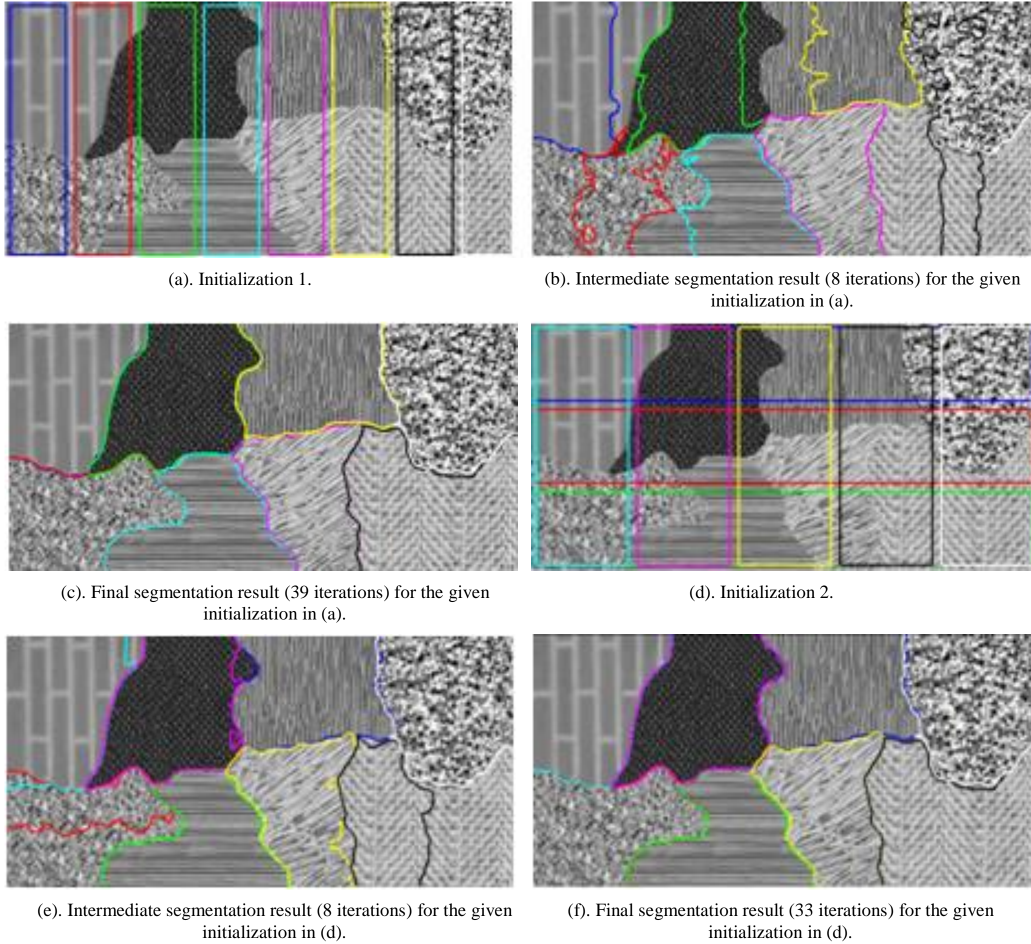
Fig. 2.7 shows segmentation results for a 5-region texture image. Fig. 2.7(a) shows the original image with an initialization using 5 level-set functions. Fig. 2.7(b) shows the final segmentation result (after 22 iterations) using the proposed method. For comparison, the method of [89] takes 2000 iterations to segment the desired regions in the same input image. The method of [89] and the proposed method succeeded in segmenting accurately the five different regions of the given texture image. However, the proposed method is able to obtain the desired segmented regions in significantly less number of iterations (33 versus 2000) as compared to the scheme of [89]. Also, it should be noted that each iteration of the proposed scheme is less computationally complex than an iteration of the scheme of [89]. For each iteration, the method of [89] requires more additions and multiplications than the proposed scheme due to the need in [89] to calculate the variances of the image regions in addition to the curve evolution functions.

The following examples are given to show that the proposed method is resilient to different initializations of the level-set functions. Figs. 2.8(a) and (c) represent two different initializations for a texture image with 5 regions. The final segmentation results using the proposed method for the previous initializations are given in Figs. 2.8(b) and (d), respectively. The proposed multi-region segmentation method succeeded in segmenting the 5 regions of the input image accurately for the different initializations



(a).Initialization 1. (b). After 60 iterations. (c).Initialization 2. (d). After 25 iterations.

Fig. 2.8. Texture segmentation results using the proposed method with different initializations.



(a). Initialization 1.

(b). Intermediate segmentation result (8 iterations) for the given initialization in (a).

(c). Final segmentation result (39 iterations) for the given initialization in (a).

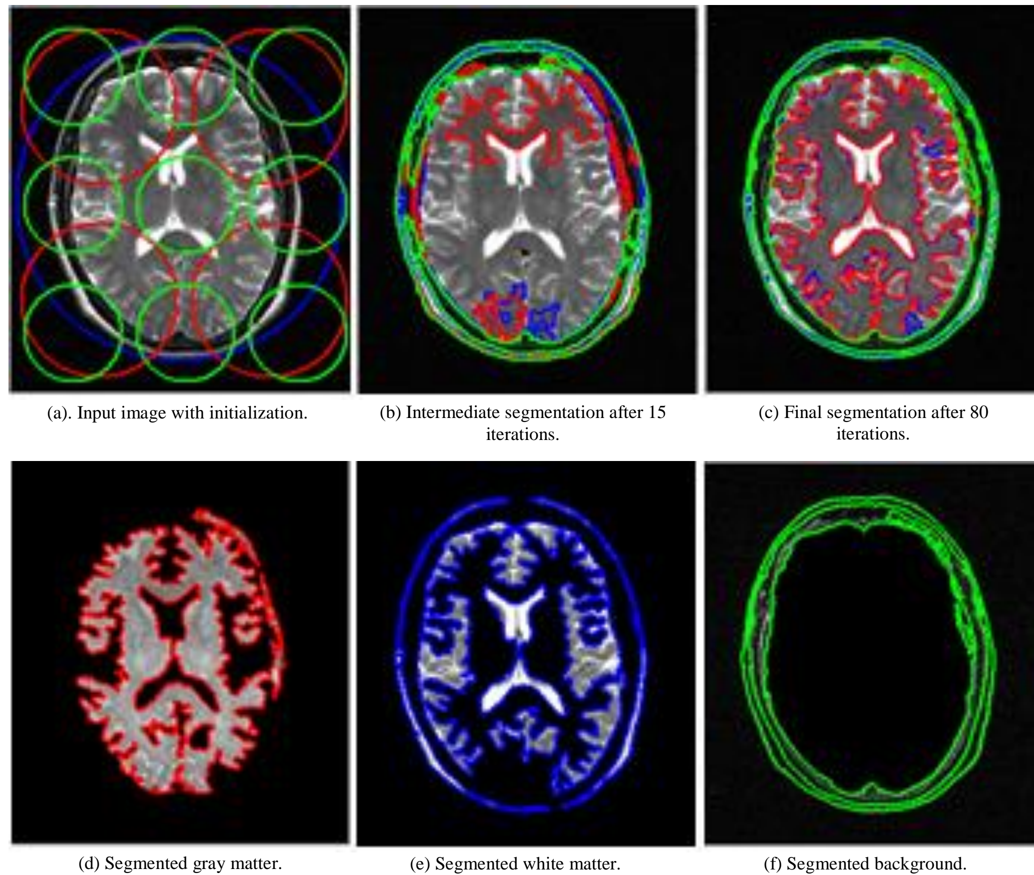
(d). Initialization 2.

(e). Intermediate segmentation result (8 iterations) for the given initialization in (d).

(f). Final segmentation result (33 iterations) for the given initialization in (d).

Fig. 2.9. Texture segmentation with 8 regions using the proposed method with different initializations.

with significantly less number of iterations as compared to [89]. Segmentation results for a texture image with 8 regions using different initializations are also shown in Fig. 2.9. The final segmentation results of the proposed method for initializations given in Figs. 2.9(a) and (d) are shown in Figs. 2.9(c) and (f), respectively. Although many of the



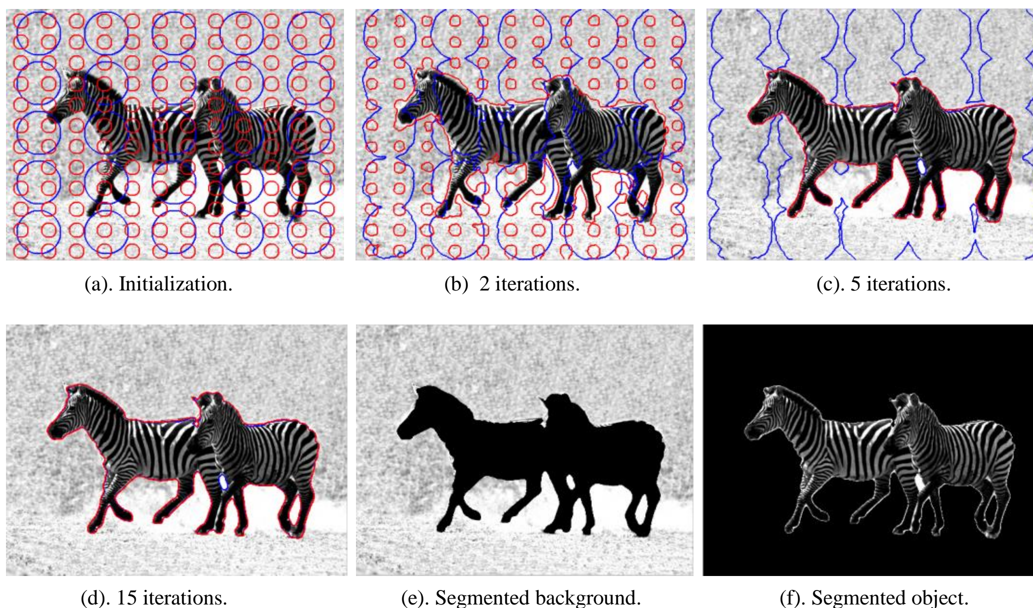
(a). Input image with initialization. (b) Intermediate segmentation after 15 iterations. (c) Final segmentation after 80 iterations.

(d) Segmented gray matter. (e) Segmented white matter. (f) Segmented background.

Fig. 2.10. Brain image segmentation using the proposed method.

regions in the image shown in Fig. 2.9 are similar and are thus hard to segment, the proposed method was able to successfully segment the desired 8 regions.

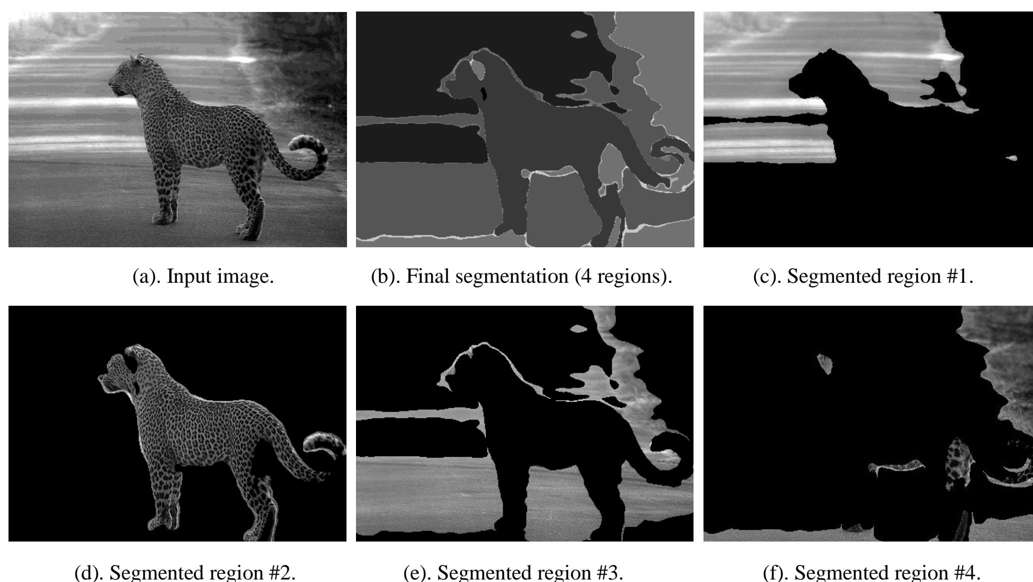
Fig. 2.10 illustrates the performance of the proposed method in the case of gray-level medical (non-texture) images. The test image in this example is a brain image with 3 regions: white matter, gray matter and background. Fig. 2.10(a) shows the brain image with an initialization that is difficult for convergence due to the location and the number of initial curves of each evolving function. Fig. 2.10(b) shows the intermediate segmentation result using the proposed method after 15 iterations. The final segmentation result using the proposed scheme after 80 iterations is shown in Fig. 2.10(c). The segmented gray matter, white matter, and the background are shown in Figs. 2.10(d), (e), and (f), respectively.



(a). Initialization. (b) 2 iterations. (c). 5 iterations.

(d). 15 iterations. (e). Segmented background. (f). Segmented object.

Fig. 2.11. Natural texture image segmentation with 2 regions using the proposed method.

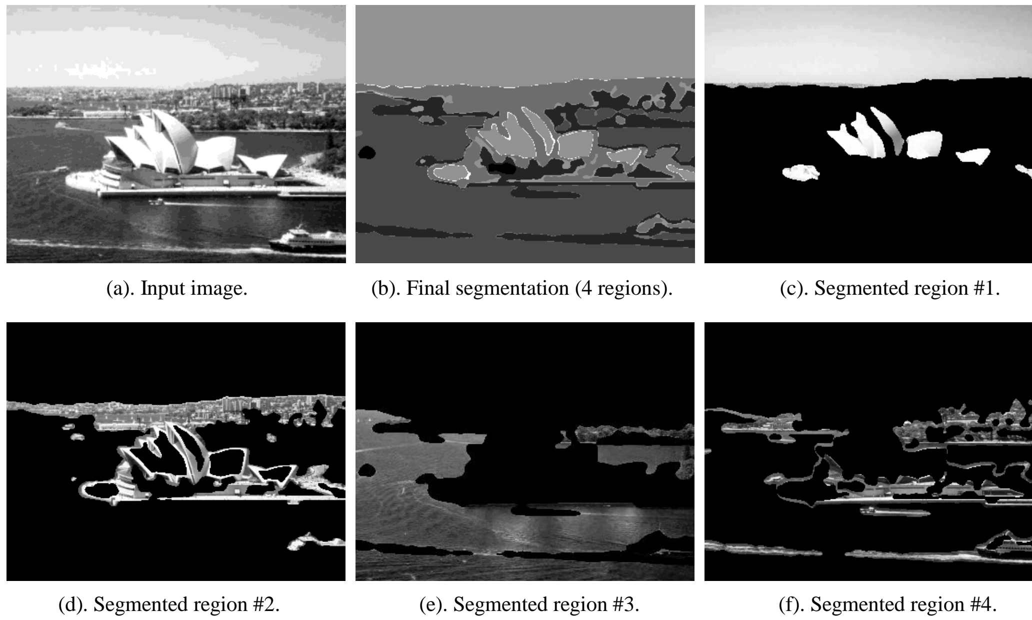


(a). Input image. (b). Final segmentation (4 regions). (c). Segmented region #1.

(d). Segmented region #2. (e). Segmented region #3. (f). Segmented region #4.

Fig. 2.12. Natural image segmentation with 4 different regions using the proposed method.

Fig. 2.11 gives an example of texture segmentation using the proposed method for a natural image with 2 regions. Figs. 2.11(b), (c), and (d) show the segmentation results at 2, 5, and 15 iterations, respectively. The segmented background and object are given in Figs. 2.11(e) and (f), respectively. Figs. 2.12 and 2.13 show the performance of the proposed multi-region segmentation method for two different natural images with 4 different regions.



(a). Input image. (b). Final segmentation (4 regions). (c). Segmented region #1.

(d). Segmented region #2. (e). Segmented region #3. (f). Segmented region #4.

Fig. 2.13. Natural image segmentation with 4 regions using the proposed method.

2.6. Summary

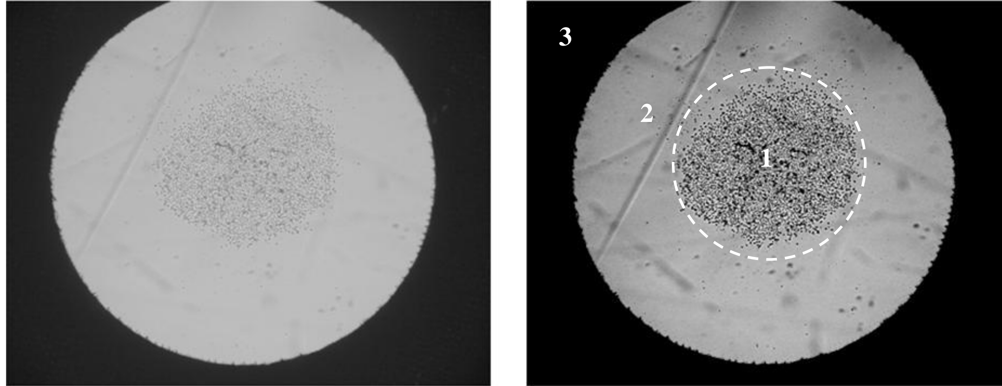
A multi-region segmentation based on constrained level-set evolution functions was proposed in this chapter. The proposed method is less sensitive to initializations and exhibits faster convergence as compared to existing multi-region level-set segmentation schemes. Simulation results using synthetic, texture, medical, and natural images were presented to show the robustness of the proposed method in segmenting multi-region images with any number of regions and with different types of initializations.

CHAPTER 3: CELL EVOLUTION ANALYSIS SCHEME BASED ON LEVEL-SET FRAMEWORKS

3.1. Introduction

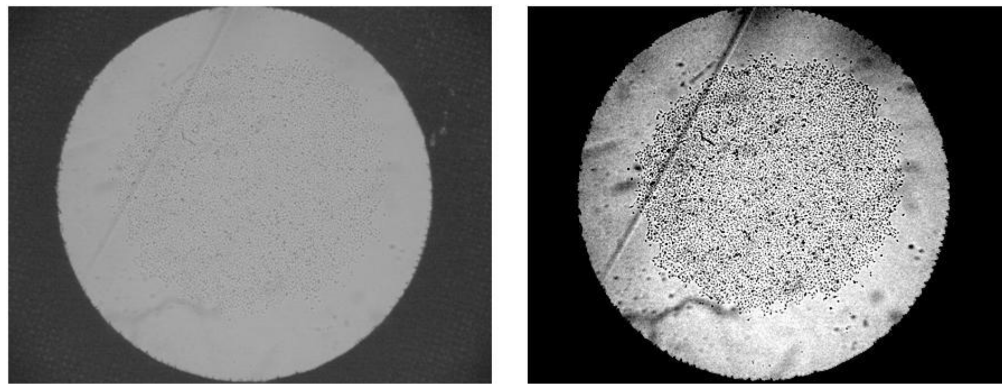
Studying the migration, proliferation, and dispersion behavior of cells contributes to the understanding of biological processes and disease pathologies such as cancer, angiogenesis, vascular stenosis and arthritis. Cell migration analysis underlies fundamental features of embryonic development, wound healing, immune cell trafficking, and pathological process such as cancer metastasis [118-121]. Techniques that assess the biochemical and molecular mechanisms of cell migration afford insight into normal biological processes and underpinnings of pathology [122].

In this chapter, we will discuss the application of level-set image segmentation in biomedical imaging. A Cell Evolution Analysis (CEA) system is presented in this chapter. The proposed CEA system consists of 3 components including cell migration analysis, cell proliferation analysis, and cell dispersion analysis. The cell migration analysis is performed by computing the overall migration rate of the cell cluster (region of interest). The Region of Interest (ROI) is first segmented at different time instances, and then the overall migration rate is represented by the rate of change of the area of the segmented ROI with respect to time. For this purpose, we propose two segmentation schemes for extracting the cell cluster area, which is needed for cell migration analysis: the first scheme is based on piecewise image segmentation and the second scheme is based on texture image segmentation. The cell proliferation analysis is represented by counting the individual segmented cells within the ROI at different time points. This can be done by applying an automatic thresholding procedure to segment and count the individual cells inside the segmented ROI area. The cell dispersion analysis is performed for cells that migrate away from the cell cluster region (ROI). This can be implemented



(a). Original bladder cancer cell image. (b). Enhanced version of the image in (a).

Fig. 3.1. Sample image of the cell population in the bladder cancer cell image at initial time point.



(a). Original bladder cancer cell image. (b). Enhanced version of the image in (a).

Fig. 3.2. Sample image of the cell population in the bladder cancer cell image after 24 hours.

by measuring the distances and counting the migrated cells (outside the ROI) in different directions from the centroid of the ROI area.

In the proposed CEA system, two automatic segmentation methods for the cancer cell images are proposed. The first segmentation method is used to segment the region of interest (ROI) by using level-set-based segmentation techniques to compute the cell migration rate at different time points. The second proposed segmentation method is used to segment and count the individual cells inside the Region of Interest (ROI) in order to compute the cell proliferation at different time instances. The proposed automatic cell evolution analysis system was applied to different types of cancer cell images with poor contrast and high cell concentrations, even when the cells are overlapping and tiny. Furthermore, the proposed texture-based ROI segmentation scheme is robust to noise and

natural artifacts in the considered cell images. Simulation results are presented in this chapter to show the performance of the proposed CEA system for different types of cancer cell images such as breast cancer, pancreatic cancer, glioblastoma, melanoma, lung cancer, and bladder cancer cells.

3.2. Existing Problem and Objectives

The considered cell images consist, each, of three different regions as shown in Fig. 3.1. The three regions can be classified as follows:

1) The cell cluster region: we refer to this cell cluster region as the region-of-interest (ROI), where the cells are concentrated (region 1 in Fig. 3.1(b)).
2) Outer-region: this represents the background region consisting of the area between the ROI and the outer-circle (region 2 in Fig. 3.1(b)).
3) Dark region: this represents the region outside the outer-circle (region 3 in Fig. 3.1(b)). This region is optional and might not exist in some images, in which case there would be only an ROI and a grayish background area.

Those cancer cell images were taken at different time instances from the migration assay using an imaging device. The main goal is to use an automatic method to determine cell migration rates, proliferations, and dispersions from those images. Fig. 3.1(a) shows the image of a cell population at the initial time point. For visual clarity, an enhanced version of this image is shown in Fig. 3.1(b) for illustration of the cell population (region of interest) inside the image. Fig. 3.2(a) shows the image of the same cell population after 24 hours and its enhanced version is shown in Fig. 3.2(b). From Figs. 3.1 and 3.2, the migration of the cells can be clearly seen by looking at the cell cluster regions in the two images. The difference in the areas occupied by the cell population at two time points of the same image allows us to calculate the overall migration rate of the region of interest (ROI).

The considered cancer cell images can be acquired by using low cost lab instruments (an inverted microscope and digital camera). These instruments are available and widely used in labs due to their relative low cost, and the majority of researchers and scientists can afford to buy these instruments (approximately $25K for the camera and microscope). There are several instruments in the marketplace that were designed to acquire high quality images in a high throughput manner such as the IN Cell Analyzer 1000 (GE Healthcare), the ArrayScan (Cellomics), and the Opera LX (PerkinElmer). The cost can be anywhere from $200K and up depending on the extra models/software tools that can be added to the system. These instruments are very expensive and many labs cannot afford to buy them. In many labs, the cell evolution analysis is done manually or by using inaccurate tools to calculate the cell migration rates and for cell counting. In the manual process, the cell cluster region is assumed to be close to a circular shape. Therefore, the migration rate is calculated by measuring the radii r_2 and r_1 of the cluster regions (ROI) at two different time instances t_2 and t_1, respectively. Then, the migration rate is represented by the slope between the two time instances, which is given by $(r_2 - r_1)/(t_2 - t_1)$. This manual procedure is time consuming and tiring, and can produce inconsistent and inaccurate results especially if the cluster region has irregular shapes or if there are several cell images that need to be processed as is usually the case for drug discovery and cancer diagnosis.

3.3. Review of the Existing Methods

Cell segmentation is one of the many challenging tasks in cellular image processing. Several methods have been proposed and developed for cell segmentation and tracking [93, 123-129]. However, most of the existing methods work for specific cell types and under specific constraints. In [123], a gradient-based level-set method is used for cluster segmentation of neural stem cells. However, the method of [123] is not robust in the presence of noise. Moreover, this method is not fully automated as it requires a prior

knowledge of the cell cluster location relative to the initial boundary of the evolving level-set function. In [124], a cell tracking scheme is presented, for in vitro phase-contrast video microscopy, using a combination of mean-shift processes. However, in [124], the user has to select the locations of the cells manually in the first or in the last frame of the video sequence. In addition, the original frames need to be pre-processed by performing contrast enhancement and illumination correction. In [125], a cell cluster segmentation algorithm is presented based on global and local thresholding for In-SITU microscopic images. This method requires noise-free images, non-overlapping cells, and a high-contrast between cells and background. Also, many parameters need to be adjusted to compute the local threshold. In [126], the active contour without edges method of [93, 100] is used for segmenting and tracking the multiple motile epithelial cells during wound healing. In [127], a topology-constrained level-set method is presented to prevent the merging of touching and partially overlapping cells. Level-set methods were also used for cell segmentation and tracking in [128]. In [129], a probabilistic model was proposed for the segmentation of hematopoietic stem cells; the proposed model is based on identifying the most probable cell locations in the image on the basis of cell brightness and morphology. The latter method is sensitive to cell overlap, cell shape, and the used threshold. Moreover, the methods of [127-129] require noise-free images with a good contrast between the cells and the background.

In this chapter, a Cell Evolution Analysis (CEA) system is presented for the segmentation of the cell cluster region and for the analysis of the overall cell migration rate, cell proliferation rate, and cell dispersion rate of different cancer cell images. In the following sections of this chapter, more details will be given about the issues related to the considered cancer cell images and then, we will discuss how these issues are tackled in the proposed automatic cell evolution analysis system.

3.4. Cell Evolution Analysis Scheme

The primary goal of this chapter is to develop an automated cell evolution analysis technique to replace the manual assessment process. This will significantly decrease the time consumed for analyzing the cell migration and cell proliferation rates as compared to manual or semi-manual processing, in addition to increasing the accuracy and consistency of the results as well as allowing high-throughput imaging and processing. The proposed method is ubiquitous as it can work with any imaging equipment (low-end as well as high-end imaging equipment) and various image formats, resolution, and qualities.

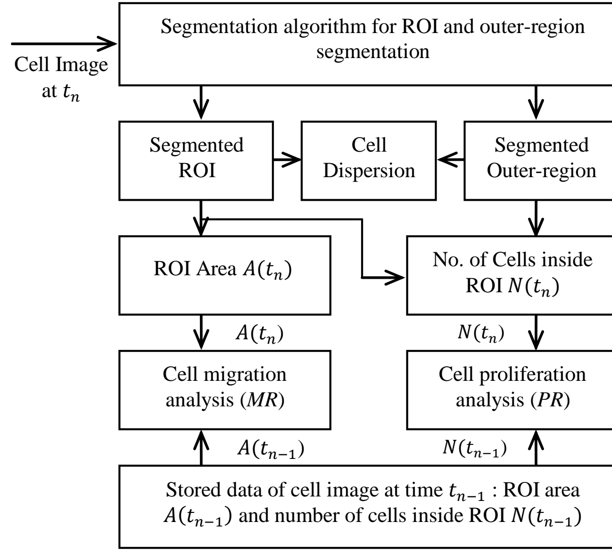


Fig. 3.3. Block diagram of the proposed CEA system.

A block diagram summarizing the proposed cell evolution analysis system is shown in Fig. 3.3. The CEA system consists of three main components: cell migration analysis, cell proliferation analysis, and cell dispersion analysis. The proposed scheme proceeds as follows:

- For each cell image at time t_n, the ROI and outer-region areas are segmented first using a noise resilient segmentation method.
- Then, the segmented ROI mask is used to compute the area inside the ROI at time t_n which is then compared to the previously computed and stored ROI area at time t_{n-1} to get the overall cell migration rate at two time points t_{n-1} and t_n.
- The segmented ROI and the segmented outer-region areas are used to estimate an automatic threshold to segment and count the individual cells inside the ROI at

time t_n. The count at time t_n is then compared with the previously computed and stored number of cells at time t_{n-1} to calculate the proliferation rate.

- The individual migrated cells outside the ROI area are segmented and counted. The angles and the distances from the centroid of the ROI area are taken into consideration to calculate the cell dispersion at time t_n.

An overview of the proposed cell migration analysis, cell proliferation analysis, and cell dispersion analysis is given in the subsections below followed by detailed descriptions in Sections 3.5 to 3.7.

3.4.1. Cell migration analysis

The cell migration analysis can be achieved by segmenting the region where the cells are clustering (ROI), and then by tracking the evolution of the boundary of that region over time. The size and the shape of the ROI in one image change with time due to the duplication and migration of cells. This is can be clearly seen in Fig. 3.1(b) and Fig. 3.2(b) which represent samples of bladder cancer cell images at two different time instances: 16 hour and 40 hour, respectively. The difference between the areas occupied by the cell population at these two time instances allows us to calculate the overall migration rate. Given the two extracted ROI areas, $A(t_n)$ and $A(t_{n-1})$, at two time instances, t_n and t_{n-1} , respectively, the overall cell migration rate is given by

$$MR = \frac{A(t_n) - A(t_{n-1})}{t_n - t_{n-1}} \tag{3.1}$$

where $A(t_n)$ is the area inside the ROI at time t_n. Two methods for ROI segmentation are proposed in this chapter. The two proposed ROI segmentation schemes are based on level-set segmentation, and include a piecewise-based scheme and a texture-based scheme. The texture-based level-set segmentation method is combined with a wavelet-based structure tensor vector in order to produce improved robustness to noise and other artifacts that are present in the lab environment. The proposed methods give a highly

accurate boundary representation of the cell clustering region. Further details about each proposed ROI segmentation method are discussed later in Section 3.5 and Section 3.6 of this chapter.

3.4.2. Cell proliferation analysis

Cell proliferation refers to an increase in the number of cells as a result of cell growth and cell division over time. The analysis of the cell proliferation is achieved by segmenting and counting the individual cells inside the ROI at different time points. By comparing the number of cells at different time points, one can compute the number of new generated cells due to cell growth and cell division over time. Given the number of the individual cells inside the segmented ROI's, $N(t_n)$ and $N(t_{n-1})$ at time t_n and t_{n-1}, respectively, the overall cell proliferation rate is given by

$$PR = \frac{N(t_n) - N(t_{n-1})}{t_n - t_{n-1}} \tag{3.2}$$

where $N(t_n)$ is the number of individual cells inside the ROI at time t_n. A proposed individual cell segmentation method will be presented in Section 3.7 of this chapter. The proposed method is derived from the histogram distributions of both the segmented ROI and the segmented outer-region.

3.4.3. Cell dispersion analysis

Cell dispersion analysis is an important step to study and observe the changes of the collective behavior of different cells. Moreover, it helps in measuring tumor cell aggressiveness [130]. Cell dispersion occurs as cells migrate in clusters or cells migrate individually outside the ROI area. In this work, we measure the dispersion distance between the migrated cells outside the ROI and the centroid of the ROI area as well as the number of cells in different directions outside the ROI. Studying the migration behavior of the dispersed cells helps in determining whether the individual cell migration

is uniform or non-uniform. A cell that migrates faster than the rest of the cells is more likely to be a cancer cell.

3.5. Cell Migration Scheme Based on Piecewise ROI Segmentation

A block diagram of the proposed piecewise ROI segmentation scheme is shown in Fig. 3.4. The proposed scheme proceeds as follows. For each image, a cell merging operation is first performed to fill the gaps between individual cells; this is done using a closing operation (dilation then erosion) with a circular structural element. The structure element radius is very small (3 or 5) because the gaps between the individual cells are not wide. In addition, in the presence of noise, spatial smoothing is performed using a simple averaging filter. This is followed by a segmentation method to segment the ROI region. We use a level-set-based segmentation technique that uses two level-set functions for extracting the ROI together with the outer-circle contour. The resulting segmented image can include small regions due to the artifacts in the original image. These artifacts can be easily discarded due to their small size relative to the ROI's size. The overall migration rate of the cells is determined by tracking the evolution of the ROI size at different time instances. The size and the shape of the ROI area in one image change through time due to the duplication and migration of cells. The cell proliferation behavior is analyzed through a cell counting procedure that is applied to the extracted ROIs at different time points.

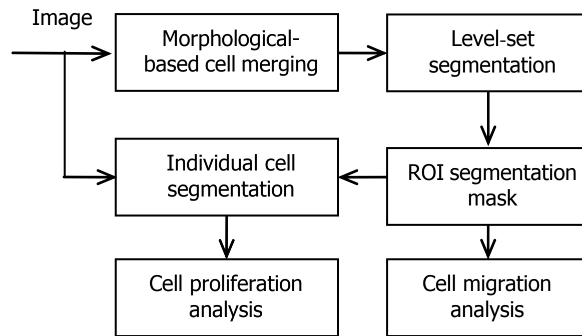


Fig. 3.4. Block diagram of the proposed CEA scheme based on piecewise ROI segmentation.

The considered bladder cancer cell images consist of three regions as described earlier and as shown in Fig. 3.1(b). Most of the considered cancer cell images have a lot of artifacts and noise, poor contrast, and high cell concentrations. An automatic

segmentation method needs to be devised to distinguish between these three regions. Conventional segmentation methods, such as thresholding, watershed, and clustering, do not give satisfactory results in segmenting the ROI, due to the presence of artifacts and due to the poor contrast in the considered images. The best way to overcome these problems is to use a robust segmentation method such as the level-set segmentation methods described in Chapter 2 of this thesis.

Two level-set functions are required to distinguish between the three regions of the considered cell images. The level-set method of [100] is, consequently, not applicable since it uses one level-set function to differentiate between two different regions: one represents the individual cells and one represents the background. Alternatively, when the scheme of Chan & Vese [93], with two level-set functions, is applied directly to these images, it only succeeds in capturing the contour of the outer circle, but it fails to capture the contour of the ROI. This occurs because the considered ROI is not a piecewise constant area and there are many gaps between the individual cells inside the ROI region. Moreover, the intensity mean values of the ROI and the remaining area inside the outer-circle are very close to each other in most cases. That's why the Chan and Vese model fails to segment the ROI.

Two level-set based methods are proposed for the piecewise ROI segmentation method: the weighted level-set method and the two-step level-set method. Both methods give similar segmentation results on the considered cell images. However, the two-step level-set segmentation method converges faster than the weighted level-set method.

3.5.1. Weighted level-set segmentation method

This method is a modified version of the Chan and Vese method [93] with two level-set functions as discussed in Chapter 2. In this modified version, the following weighted energy function is minimized:

$$
\begin{aligned}
F_4(C,\Phi) = \; & \lambda_1 \int_{\Omega} |I_0 - c_{11}|^2 H(\phi_1)H(\phi_2)\; dxdy \\
& + \lambda_2 \int_{\Omega} |I_0 - c_{10}|^2 H(\phi_1)\big(1 - H(\phi_2)\big)\; dxdy \\
& + \lambda_3 \int_{\Omega} |I_0 - c_{01}|^2 \big(1 - H(\phi_1)\big)H(\phi_2)\; dxdy \\
& + \lambda_4 \int_{\Omega} |I_0 - c_{00}|^2 (1 - H(\phi_1))(1 - H(\phi_2))\; dxdy \\
& + v_1 \int_{\Omega} |\nabla H(\phi_1)|\; dxdy + v_2 \int_{\Omega} |\nabla H(\phi_2)|\; dxdy
\end{aligned}
\tag{3.3}
$$

where $\lambda_i, i = 1,2,3,4,$ and $v_i, i = 1,2,$ are constants, I_0 is the considered image, $\Phi = [\phi_1, \phi_2]$ represent the two level-set functions and are used to define four different regions. In (3.3), $C = [c_{11}, c_{10}, c_{01}, c_{00}]$ are constants corresponding to the mean of the image I_0 in the regions $\{\phi_1 > 0, \phi_2 > 0\}, \{\phi_1 > 0, \phi_2 < 0\}$, $\{\phi_1 < 0, \phi_2 > 0\}$, and $\{\phi_1 < 0, \phi_2 < 0\}$, respectively, and are given by

$$
\begin{aligned}
c_{11}(\phi_1, \phi_2) &= \frac{\int_\Omega I_o\, H(\phi_1)H(\phi_2) dxdy}{\int_\Omega H(\phi_1)H(\phi_2) dxdy}, \\
c_{10}(\phi_1, \phi_2) &= \frac{\int_\Omega I_o\, H(\phi_1)\big(1 - H(\phi_2)\big) dxdy}{\int_\Omega H(\phi_1)\big(1 - H(\phi_2)\big) dxdy}, \\
c_{01}(\phi_1, \phi_2) &= \frac{\int_\Omega I_o \big(1 - H(\phi_1)\big)\, H(\phi_2) dxdy}{\int_\Omega \big(1 - H(\phi_1)\big)\, H(\phi_2) dxdy}, \\
c_{00}(\phi_1, \phi_2) &= \frac{\int_\Omega I_o \big(1 - H(\phi_1)\big)\, \big(1 - H(\phi_2)\big) dxdy}{\int_\Omega \big(1 - H(\phi_1)\big)\, \big(1 - H(\phi_2)\big) dxdy}.
\end{aligned}
\tag{3.4}
$$

The curve evolution functions for the two level-set functions ϕ_1 and ϕ_2 are obtained by minimizing the energy function of (3.3) with respect to ϕ_1 and ϕ_2, and are given by

$$\frac{\partial \phi_1}{\partial t} = \delta(\phi_1)\{ v_1\, div(\nabla\phi_1/|\nabla\phi_1|) - (\lambda_1(I_0 - c_{11})^2 - \lambda_3(I_0 - c_{01})^2)H(\phi_2) - (\lambda_2(I_0 - c_{10})^2 - \lambda_4(I_0 - c_{00})^2)(1 - H(\phi_2))\},$$
$$\frac{\partial \phi_2}{\partial t} = \delta(\phi_2)\{ v_2\, div(\nabla\phi_2/|\nabla\phi_2|) - (\lambda_1(I_0 - c_{11})^2 - \lambda_2(I_0 - c_{10})^2)H(\phi_1) - (\lambda_3(I_0 - c_{01})^2 - \lambda_4(I_0 - c_{00})^2)(1 - H(\phi_1))\} \quad (3.5)$$

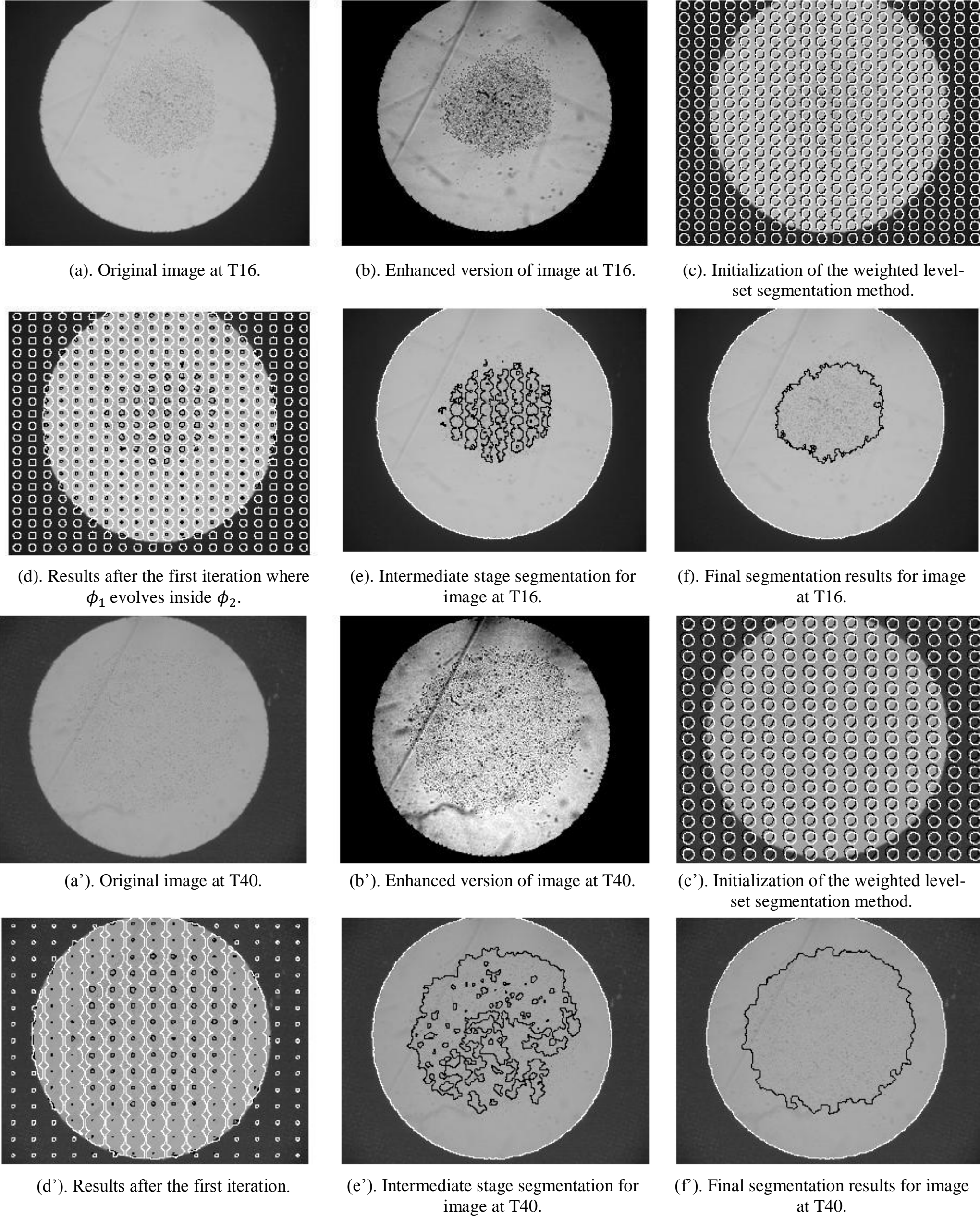
In (3.3), (3.4), and (3.5), ϕ_1 and ϕ_2 represent the two level-set functions, which evolve based on minimizing the energy function in (3.3) to segment the input image I_0. The method described in [93] represents the multiphase level-set representation and is given by (3.3) and (3.5) when setting the weights $\lambda_i, i = 1,2,3,4,$ to be equal to one, and when $v_1 = v_2$. The method of [93] uses equal weight values for each region and, therefore, it fails to segment the ROI of the considered cancer cell images even after performing the cell merging operation. Instead, the ROI is partitioned into different regions by ϕ_1 and ϕ_2. This is due to the fact that the ROI has different mean values instead of one constant mean value (piecewise constant), even after applying the cell merging operation. In addition, the method of [93] can introduce gaps inside the segmented regions when the number of regions is not a power of 2. In order to overcome this problem, the proposed weighted level-set segmentation method is derived by choosing the proper values of the weight parameters $\lambda_i, i = 1,2,3,4,$ in order to force one level-set function ϕ_1 to evolve inside ϕ_2 to capture the ROI area and the other level-set function ϕ_2 to evolve outside ϕ_1 to capture the outer-circle area. If we choose $\lambda_1 = \lambda_2 = \lambda_a$ and $\lambda_3 = \lambda_4 = 1$, then, it can be easily shown that the evolutions of the level-set functions ϕ_1 and ϕ_2 in (3.5) are given by:

$$
\begin{aligned}
\frac{\partial \phi_1}{\partial t} &= \delta(\phi_1)\left\{ v_1\, div\left(\frac{\nabla \phi_1}{|\nabla \phi_1|}\right) - (\lambda_a (I_0 - c_{11})^2 - (I_0 - c_{01})^2) H(\phi_2) \right. \\
&\qquad \left. - (\lambda_a (I_0 - c_{10})^2 - (I_0 - c_{00})^2)(1 - H(\phi_2)) \right\}, \\
\frac{\partial \phi_2}{\partial t} &= \delta(\phi_2)\left\{ v_2\, div\left(\frac{\nabla \phi_2}{|\nabla \phi_2|}\right) - \lambda_a ((I_0 - c_{11})^2 - (I_0 - c_{10})^2) H(\phi_1) \right. \\
&\qquad \left. - ((I_0 - c_{01})^2 - (I_0 - c_{00})^2)(1 - H(\phi_1)) \right\}
\end{aligned} \tag{3.6}
$$

In (3.6), when choosing $\lambda_a > 1$, more weight is given to the region inside $\phi_1 (\phi_1 > 0)$ to evolve inside $\phi_2 (\phi_2 > 0)$. In this way, after few iterations, ϕ_1 starts to evolve inside ϕ_2 to capture the ROI, while ϕ_2 evolves to capture the outer-circle. This method was successfully used to segment the ROI region in the considered cancer cell images. In the following, several examples are used to illustrate the performance of the weighted level-set method to segment the region of interest (ROI) of bladder cancer cell images.

3.5.1.1. Simulation results using the weighted level-set ROI segmentation method

In this section, the weighted level-set segmentation method, defined by (3.3) to (3.5), is used to extract the ROI and the outer-circle areas of the considered bladder cancer cell images. In this method, three parameters need to be selected: $v_1, v_2,$ and λ_a. In the following examples, we choose $v_1 = v_2 = 1,$ and $\lambda_a = 2$. For the stopping criterion, the algorithm stops if the absolute differences between the current and previous values of ϕ_1 and ϕ_2 are less than a certain threshold (0.02 in our implementation). For the initialization, we use multiple randomly generated small circles for both ϕ_1 and ϕ_2. Figs. 3.5(a)-(f) illustrate the results of using the proposed weighted level-set image segmentation to segment the ROI and outer-circle areas of the considered bladder cancer cell image (Fig. 3.5(a)) at a time point equals to 16 hours. Fig. 3.5(d) illustrates the intermediate step, where the level-set function ϕ_1 $(\phi_1 > 0)$ starts to evolve inside ϕ_2 $(\phi_2 > 0)$ to capture the boundary of the ROI, while ϕ_2 starts to evolve outside



(a). Original image at T16. (b). Enhanced version of image at T16. (c). Initialization of the weighted level-set segmentation method.

(d). Results after the first iteration where ϕ_1 evolves inside ϕ_2. (e). Intermediate stage segmentation for image at T16. (f). Final segmentation results for image at T16.

(a’). Original image at T40. (b’). Enhanced version of image at T40. (c’). Initialization of the weighted level-set segmentation method.

(d’). Results after the first iteration. (e’). Intermediate stage segmentation for image at T40. (f’). Final segmentation results for image at T40.

Fig 3.5. ROI segmentation using the proposed weighted level-set method for a bladder cancer cell image at T16 and T40.

$\phi_1(\phi_1 < 0)$ to capture the outer-circle boundary. The final segmentation result is shown in Fig. 3.5(f), where the proposed method succeeds to extract the boundaries of the ROI and the outer circle. The ROI area after 16 hours is approximately $A(T_{16}) = 720{,}820\ \mu m^2$ for a pixel size of $4.926\ \mu m^2/pixel$. Figs. 3.5(a’)-(f’) illustrate the ROI segmentation results of the bladder cancer cell image at a time instance equals to 40

hours (after 24 hours from T16). The final segmentation result is shown in Fig. 3.5(f') with an ROI area equals to $A(T_{40}) = 1{,}994{,}800\ \mu m^2$. The computed overall ROI migration rate equals to $53{,}082\ \mu m^2/h$.

3.5.2. Two-step level-set segmentation method

The weighted level-set segmentation method, which is presented in the previous section, suffers from initialization and coupling problems, in addition to being relatively slow [84]. In this section, the level-set segmentation method of [84] is adopted and applied to the considered cancer cell images. This method applies ϕ_1 and ϕ_2 consecutively, rather than simultaneously. This process ensures that ϕ_1 and ϕ_2 are completely decoupled. This method proceeds in two steps as follows:

Step 1 – Evolve one level-set function ϕ_1 to segment the outer-circle using:

$$\frac{\partial \phi_1}{\partial t} = \delta(\phi_1)\left\{ v_1\, div\left(\frac{\nabla \phi_1}{|\nabla \phi_1|}\right) - (I_0 - c_1')^2 + (I_0 - c_2')^2 \right\}. \tag{3.7}$$

where c_1' and c_2' are the averages of I_0 inside and outside ϕ_1, respectively. The results of this step is the segmentation of the outer-circle region inside ϕ_1.

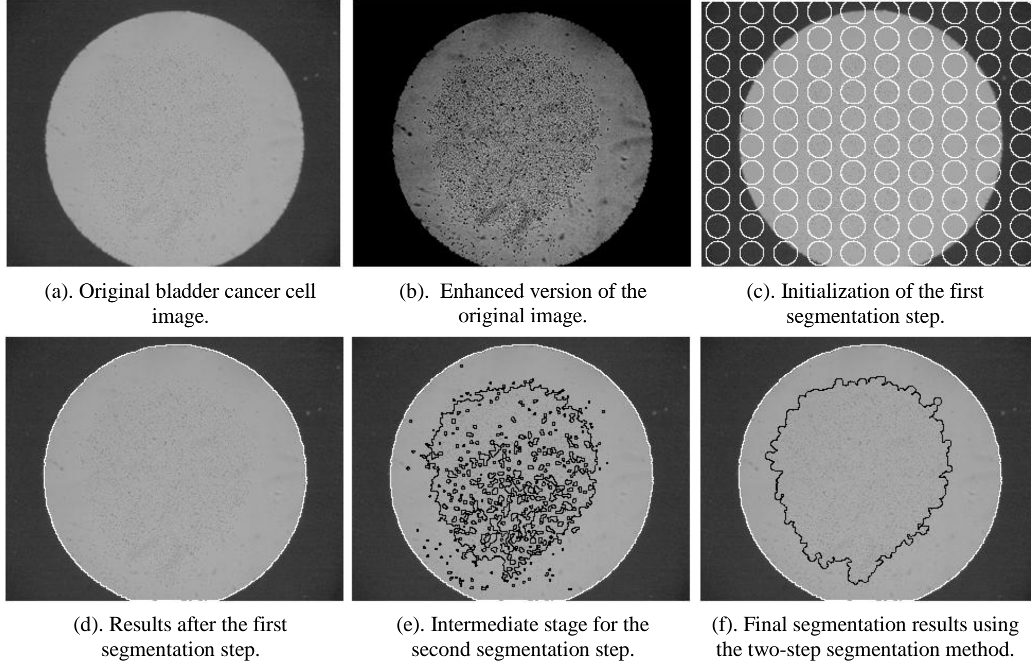
Step 2 – Evolve the level-set function ϕ_2 inside ϕ_1 to segment the ROI using:

$$\frac{\partial \phi_2}{\partial t} = \delta(\phi_2)\left\{ v_2\, div\left(\frac{\nabla \phi_2}{|\nabla \phi_2|}\right) - \lambda_1(I_0 - c_1'')^2 + \lambda_2(I_0 - c_2'')^2 \right\}. \tag{3.8}$$

where c_1'' and c_2'' are the averages of I_0 inside and outside ϕ_2, respectively, when $(\phi_1 > 0)$. In (3.8), the values of λ_1 and λ_2 affect the evolution rate of ϕ_2. The advantages of the two-step level-set method, as compared to the weighted level-set method of Section 3.5.1, include a faster speed of convergence, lower computational complexity, and lower sensitivity to the initial conditions.

3.5.2.1. Simulation results of using the two-step ROI segmentation method

Different types of cancer cell images are used to illustrate the efficiency of the two-step segmentation method. For each example, we proceed as follows: first, we apply one



(a). Original bladder cancer cell image.
(b). Enhanced version of the original image.
(c). Initialization of the first segmentation step.
(d). Results after the first segmentation step.
(e). Intermediate stage for the second segmentation step.
(f). Final segmentation results using the two-step segmentation method.

Fig. 3.6. ROI segmentation using the proposed two-step method for a bladder cancer cell image.

level-set function ϕ_1 to segment the outer-circle using (3.7). This first step takes 4 to 6 iterations to segment the outer-circle. In the second step, one can use the mask of the segmented outer-circle ($\phi_1 > 0$) as an initialization for the second segmentation step ($\phi_2^0 = \phi_1 > \delta$), where δ is a small value. Using ϕ_2^0 for initialization, one can segment the ROI using (3.8). The second step takes 6 to 10 iterations to segment the ROI. For each step, the evolution of the level-set function is stopped when the absolute difference between the current and previous level-set function is less than a certain threshold (0.02 in our implementation). Figs. 3.6(a)-(f) illustrate the process of the proposed two-step segmentation method, where Fig. 3.6(d) shows the result of the first step which segments the outer-circle using (3.7) and Fig. 3.6(f) shows the final segmentation result using the second segmentation step (3.8).

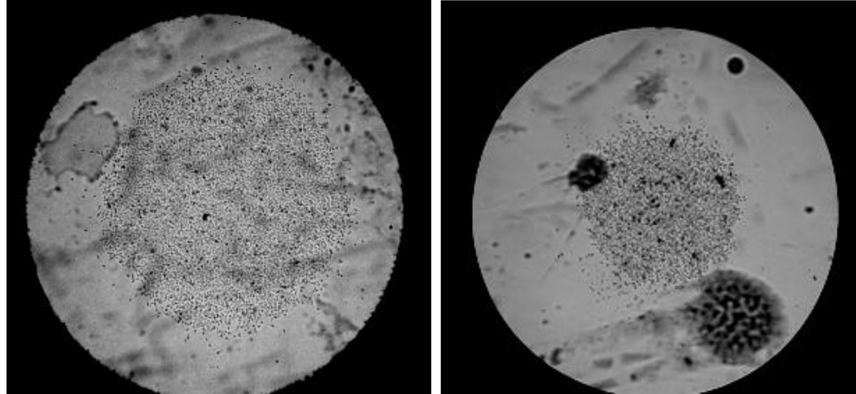


Fig. 3.7. An example of artifacts in cancer cell images.

3.6. Noise-Resilient and Robust Cell Migration using Texture-based ROI Segmentation

In the piecewise level-set segmentation methods, objects can be separated from the background based on the different mean values of the object and the background. In some cases, these mean values can be very close to each other and, therefore, the piecewise segmentation fails to segregate between the object and the background. Moreover, if the input cell image has artifacts, such as the ones shown in Fig. 3.7 between the cell culture area (ROI) and the outer-circle area, the piecewise method will segment the artifacts as part of the ROI area because the artifacts have different mean values that are different from the background mean value. Examples of those artifacts include but not limited to fingerprints, stains, fiber, and water blubs. One way to overcome the disadvantages of the piecewise level-set segmentation is to deal with the cell culture area as a texture region. This assumption is based on the observation that the ROI region has a high variance and has a texture-like structure, such as grains, sands, and other textures [131], as compared to the other present artifacts and non-cell image features. In this section, we present a robust, and noise/artifacts-resilient ROI segmentation method. The proposed method segments the ROI region based on the texture-like characteristics of the cell cluster region, which can be exploited as significant features that help in distinguishing the cell area from the background and artifacts. The proposed method combines the ROI texture

information components with a statistical level-set segmentation method. The proposed method is shown to be more robust to artifacts and noise, and it results in a more accurate segmentation of the ROI area compared to the existing methods. The proposed scheme can be applied to different cell cluster images, in which the cell cluster has texture-like characteristics, even if the considered images suffer from poor contrast, artifacts, and noise.

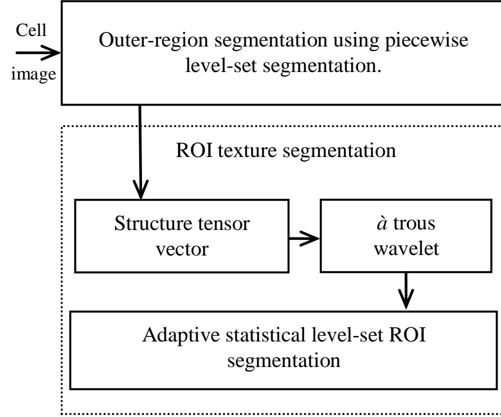


Fig. 3.8. Block diagram of the texture-based ROI segmentation.

A block diagram of the proposed texture-based ROI segmentation scheme is shown in Fig. 3.8. The procedure starts by extracting the outer-region area using a piecewise level-set segmentation by using one level-set function based on (3.7). Then, the ROI area is extracted by using an unsupervised texture-based segmentation procedure. This procedure consists of a statistical based level-set method that is applied to the structure tensor features in the wavelet domain. The structure tensor vector [33] represents the texture features that are used to represent the texture information in the ROI area. The employed statistical level-set method is based on a Bayesian formulation as in [113, 132] and is able to differentiate between objects and background based not only on their mean values but also on their variances. In contrast to the method in [132] which makes use of a non-linear diffusion filter [133] for preprocessing the structure tensor data, the proposed scheme adopts a wavelet-based approach using a fast *à* trous filter [134]. Non-linear diffusion filtering [133] requires a large number of iterations to get satisfactory results, which can be very computationally intensive process especially for large size images such as the ones considered in this paper. In addition, the method of [132] augments the structure tensor vector by adding to it the original grayscale image data. In our case, only

the structure tensor is used because it defines well the structure of the ROI region for the considered cell images. In fact, it was found that, for the considered application, augmenting the structure tensor with the original grayscale image as in [132] not only increases the computational requirements without significant improvements but can also lead to incorrect segmentation results. The following subsections give more details about the proposed texture-based ROI segmentation, which includes outer-circle segmentation, structure tensor computation, wavelet-based *à* trous filtering, and the statistical level-set segmentation method.

3.6.1. Outer-circle segmentation

The considered cell images can be initially segmented into two regions: inside and outside the outer-circle area. The difference between the mean values inside these two regions is significantly large. The outer-circle segmentation can be done by using one level-set function ϕ_1 [100]. The level-set function ϕ_1 is used to distinguish between the two regions: the area inside the outer-circle ($\phi_1 > 0$) and the area outside the outer-circle ($\phi_1 < 0$). The curve evolution function ϕ_1 which is used to segment the outer-circle area is given by:

$$\frac{\partial \phi_1}{\partial t} = \delta(\phi_1) \left\{ v_1 \, div\left(\frac{\nabla \phi_1}{|\nabla \phi_1|}\right) - (I - c_1')^2 + (I - c_2')^2 \right\} \qquad (3.9)$$

where I represents the input image, c_1' and c_2' are the averages of I inside and outside ϕ_1, respectively, $\delta(\phi_1)$ represents the delta Dirac function, and $div(\nabla\phi_1/|\nabla\phi_1|)$ represents the curvature of the level-set function ϕ_1. The segmented outer-circle is represented by the region inside $\phi_1(\phi_1 > 0)$. The level-set evolution function (3.9) stops when the mean squared error between the current and the previous level-set function (ϕ_1^n and ϕ_1^{n-1}) is less than a certain small threshold (0.02 in our implementation).

3.6.2. Structure tensor vector

The key idea in texture image segmentation is to extract some feature components that would form a good descriptor for the texture regions in the input image. The features vector, or descriptor, is used to distinguish between different texture regions inside the input image. In [113, 114, 116, 132, 135], the structure tensor vector is used as a feature descriptor for a wide range of texture images. In this work, the structure tensor vector is used to effectively represent the texture-like structure of the ROI. The structure tensor features are computed from the gradient of the image I as follows:

$$G = \nabla I \; \nabla I^T = \begin{bmatrix} I_x \\ I_y \end{bmatrix} \begin{bmatrix} I_x & I_y \end{bmatrix} = \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix} \tag{3.10}$$

where I_x and I_y represent the derivatives of I in the directions of x and y, respectively. The structure tensor vector $\hat{\boldsymbol{u}} = (u_1, u_2, u_3)$ can be represented using I_x and I_y as follows:

$$\hat{\boldsymbol{u}} = (I_x^2, I_y^2, I_x I_y) \tag{3.11}$$

3.6.3. À Trous Wavelet filtering

The image components u_i ($\mathrm{i} = 1,2,3$) of the structure tensor vector (3.11) have to be regularized first before applying the statistical based level-set segmentation. This regularization step is applied to the cell cluster region inside the outer-circle. The regularization step is essential to smooth texture regions while preserving the edges between different texture regions. This can significantly influence the segmentation results. In the proposed scheme, the undecimated wavelet transform is adopted for this regularization and is implemented using a fast *à* trous filtering [134]. The adopted *à* trous wavelet transform has many nice properties including translation-invariance, efficient implementation, and high correlation among the wavelet coefficients across scales. Moreover, it satisfies the Lipschitz regularity property [136] which can be exploited for differentiating between strong edges and noise or weak-edge singularities. Based on the

Lipschitz property, noise and weak edges diminish at a much faster rate than stronger edges as the scale increases. These properties are exploited in the proposed wavelet-based regularization procedure which can smooth variations, including weak edges, texture, and noise, while preserving the strong edges.

The *à* trous wavelet is implemented using separable filtering with an impulse response $h = [1/16, 1/4, 3/8, 1/4, 1/16]$ for the row-wise and column-wise 1-D filters. Let $A_{0,i}(x,y) = u_i(x,y), (i = 1,2,3)$ represents the input structure tensor image component of the *à* trous wavelet. The output approximation image at the first level after applying a separable convolution is $A_{1,i}(x,y)$, and the detail image is given by $D_{1,i}(x,y) = A_{0,i}(x,y) - A_{1,i}(x,y)$. Similarly, one can get $A_{2,i}(x,y)$ and $D_{2,i}(x,y)$ from $A_{1,i}(x,y)$ for the second level. In general, for any level j ($1 < j < L$), where L is the maximum decomposition level, $A_{j,i}(x,y)$ and $D_{j,i}(x,y)$ can be expressed as follows:

$$\begin{gathered} A_{j,i}(x,y) = \sum_l h(l) \sum_k h(k)\ A_{j-1,i}\left(x - 2^{j-1}k, y - 2^{j-1}l\right) \\ D_{j,i}(x,y) = A_{j-1,i}(x,y) - A_{j,i}(x,y) \end{gathered} \tag{3.12}$$

where j represents the decomposition level and i represents the image component in $A_{j,i}(x,y)$, $i = 1,2,3$. Exploiting the shift invariance property and the high correlation among the resulting wavelet coefficients at different levels $(1 < j < L)$, for each structure tensor image component $u_i(x,y)$, $i = 1,2,3$, a new image $\Psi_i(x,y)$ can be constructed through a point-wise multiscale multiplication as follows:

$$\Psi_i(x,y) = \prod_{j=j_a}^{j_b} A'_{j,i}(x,y), \qquad 1 < j_a \le j_b \le L. \tag{3.13}$$

where $A'_{j,i}(x,y)$ is $A_{j,i}(x,y)$ normalized by the norm of the gradient of the original input image I. The resulting wavelet-filtered structure tensor vector $\boldsymbol{\hat{\Psi}} = (\Psi_1, \Psi_2, \Psi_3)$ consists of smooth areas within the same texture classes while the edges are preserved among different texture classes.

Although the selection of the number of levels L is application and image dependent, it is important to have a default value to achieve unsupervised segmentation. A subjective evaluation [137] indicated that a good compromise between noise removal and detail preservation was obtained using L= 1 for 128 x 128 (or smaller) images, L = 2 for 256 x 256 images, and L = 3 for 512 x 512 images. Hence, for an image of size $m\,x\,n$, the default value for the number of levels L is given by [137]:

$$L = max\{1, 1 + round\left(\log_2\left(min(m,n)\right) - 7\right)\} \tag{3.14}$$

For the considered images of size 1100x1300, L equals to 4, and j_a and j_b in (3.13) were set to 3 and 4, respectively.

3.6.4. Adaptive statistical level-set segmentation

The adaptive statistical level-set segmentation method of [132] is adopted to segment the cell culture region (ROI) which is represented by the generated wavelet-filtered structure tensor images $\widehat{\boldsymbol{\Psi}}$. One level-set function (ϕ_2) is required to segment the ROI inside the considered cancer cell images. The level-set function ϕ_2 is evolving inside the already-segmented outer-circle area ($\phi_1 > 0$) or the background enclosing the ROI region, to capture the ROI. The energy function and the curve evolution equations of the proposed texture-based ROI segmentation method are given by

$$\begin{aligned} E(\phi_2) = \int_{\Omega} \Bigg(& v_2 |\nabla H(\phi_2)| + \sum_{i=1}^{3} \log_e\, p_2\big(\Psi_i(x,y)\big) \\ & + \sum_{i=1}^{3} \log_e \frac{p_1\big(\Psi_i(x,y)\big)}{p_2\big(\Psi_i(x,y)\big)} H(\phi_2) \Bigg) dxdy \end{aligned} \tag{3.15}$$

and

$$\frac{\partial \phi_2}{\partial t} = \delta(\phi_2)\left(v_2\, div\left(\frac{\nabla\phi_2}{|\nabla\phi_2|}\right) + \sum_{i=1}^{3} \log_e \frac{p_1(\Psi_i)}{p_2(\Psi_i)} \right) \tag{3.16}$$

where Ψ_i is the ith component of the wavelet-filtered structure tensor vector $\widehat{\boldsymbol{\Psi}}$, and $p_r(\Psi_i)$ is the conditional probability density function of Ψ_i given a region *r (r=1,2)*,

with r = *1* corresponding to the ROI ($\phi_2 > 0$) and r = *2* corresponding to the region outside the ROI ($\phi_2 < 0$). $p_r(\Psi_i)$ is approximated using a Gaussian distribution as follows:

$$p_r\big(\Psi_i(x,y)\big) = \left(1\Big/\sqrt{2\pi\sigma_{r\,i}^2}\right) e^{-\frac{(\Psi_i(x,y)-\mu_{r\,i})^2}{2\sigma_{r\,i}^2}} \tag{3.17}$$

where μ_{ri} and σ_{ri}^2 in (3.17) represent, respectively, the mean and variance of the wavelet-filtered structure tensor component $\Psi_i(x,y)$ in region r ($r = 1{,}2$), and are given by

$$\mu_{1i} = \frac{\int_\Omega \Psi_i(x,y) H\big(\phi_2(x,y)\big) dxdy}{\int_\Omega H\big(\phi_2(x,y)\big) dxdy},$$

$$\sigma_{1i}^2 = \frac{\int_\Omega (\Psi_i(x,y) - \mu_{1i})^2 H\big(\phi_2(x,y)\big) dxdy}{\int_\Omega H\big(\phi_2(x,y)\big) dxdy},$$

$$\mu_{2i} = \frac{\int_\Omega \Psi_i(x,y) \Big(1 - H\big(\phi_2(x,y)\big)\Big) dxdy}{\int_\Omega \Big(1 - H\big(\phi_2(x,y)\big)\Big) dxdy},$$

$$\sigma_{2i}^2 = \frac{\int_\Omega (\Psi_i(x,y) - \mu_{2i})^2 \Big(1 - H\big(\phi_2(x,y)\big)\Big) dxdy}{\int_\Omega \Big(1 - H\big(\phi_2(x,y)\big)\Big) dxdy}.$$

where $H(\phi)$ is the Heaviside function that is equal to 1 when $\phi > 0$ and 0 when $\phi \leq 0$.

3.6.5. Simulation results for the texture-based ROI segmentation

The first example is used to compare the performance of the two proposed ROI segmentation schemes: piecewise and texture-based segmentation methods. In this example, the proposed schemes have been applied on noisy and poor-contrast images of 34 different bladder cancer cell lines interacting on different matrix substrates. Each cell line consists of 60 cancer cell image samples that were taken at two time instances (16 and 40 hours). The images' size is 1100x1300 with a resolution of 4.926 μm^2 per pixel. Fig. 3.9 shows an example for comparing between the proposed piecewise-based [79] and texture-based [78] ROI segmentation methods when applied to segment a bladder cancer cell image. Fig. 3.9(a) shows the original input image which exhibits low contrast in addition to artifacts in the form of large gray spots. For illustration and visual clarity, an

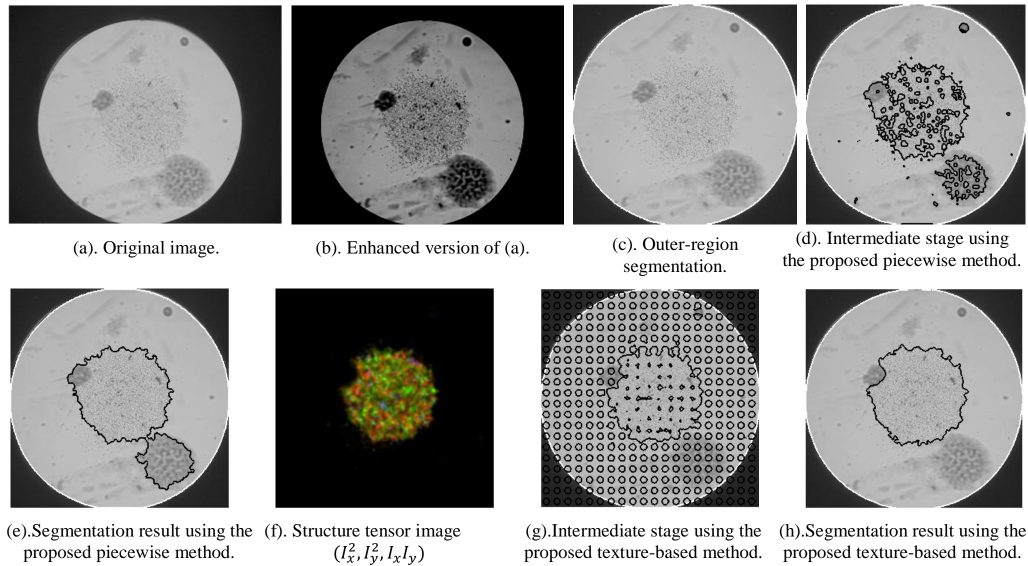


Fig. 3.9. Cell cluster segmentation using the proposed texture-based [78] and piecewise-based methods [79].

enhanced version of Fig. 3.9(a) is shown in Fig. 3.9(b), although the proposed scheme is applied directly on the original low-contrast and non-enhanced image. The first segmentation step is used to segment the outer-region area (Fig. 3.9(c)) using (3.9) as discussed in Sections 3.5 and 3.6.1. The segmented outer-region area is further segmented to get the ROI area using a second segmentation step as described in Sections 3.5 and 3.6. The intermediate and the final segmentation results using the two-step piecewise level-set segmentation scheme [79] are shown in Fig. 3.9(d) and Fig. 3.9(e), respectively. The wavelet-filtered structure tensor vector is shown in Fig. 3.9(f), which represents accurately the texture area inside the segmented outer-region. Figs. 3.9(g) & (h) show, respectively, the intermediate and the final segmentation results for the ROI area using the proposed textured-based method. Comparing the final ROI segmentation results that are obtained using the proposed piecewise [79] and the texture-based segmentation [78] schemes, it is obvious that the piecewise method fails to segment the true ROI area due to the present artifacts, while the textured-based method [78] results in an accurate segmentation of the ROI area.

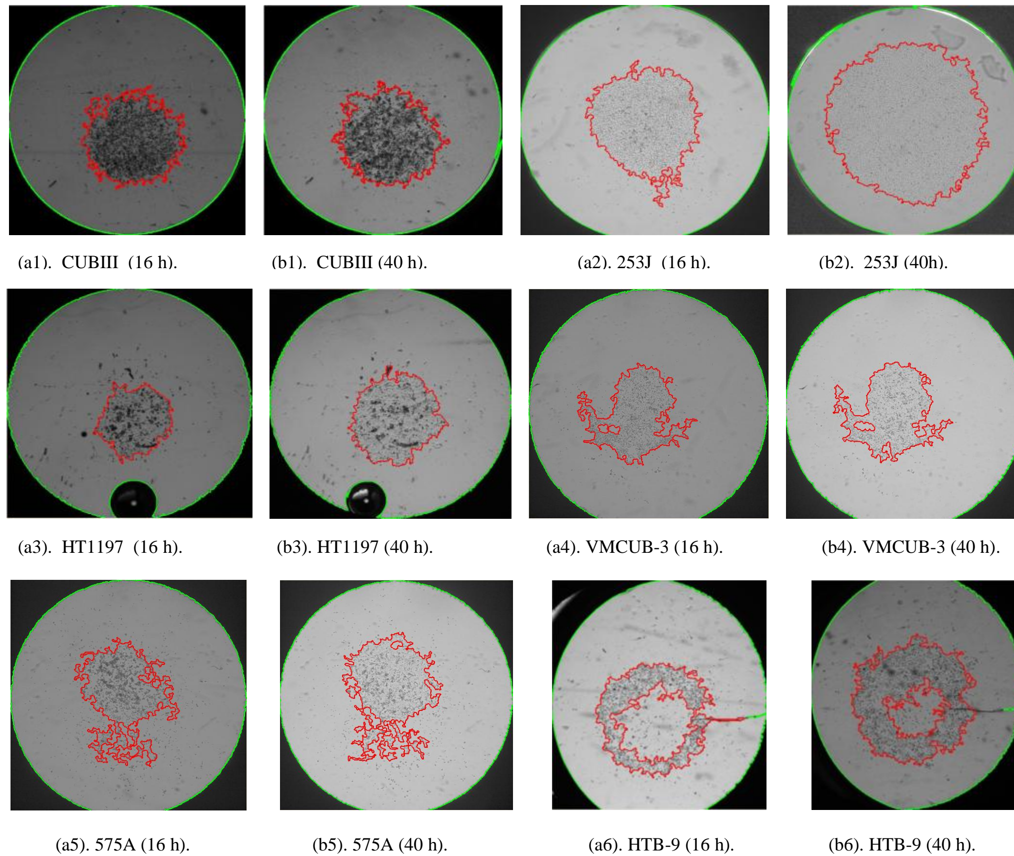
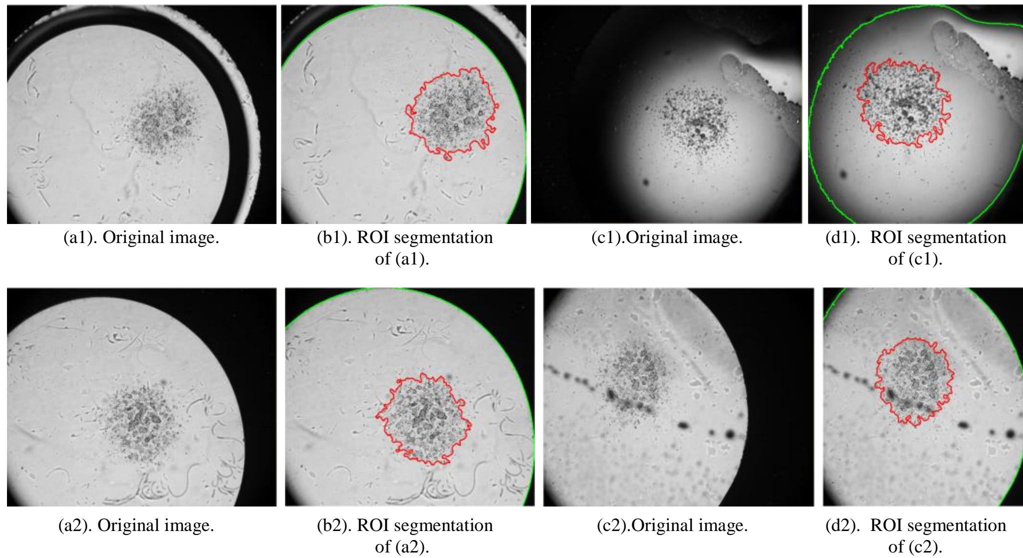


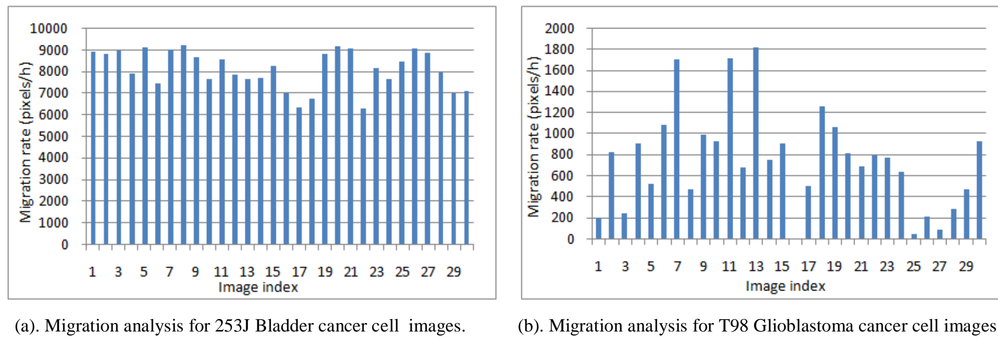
Fig. 3.10. Examples of ROI segmentation for different types of bladder cancer cell images using the proposed texture-based ROI segmentation.

The second example is given for different types of bladder cancer cell images. Fig. 3.10 shows different bladder cancer cell images with poor contrast, and different artifacts and noise in the original images. Fig. 3.10 shows the outer-circles and the ROIs segmentation results using the proposed texture-based segmentation scheme for different types of bladder cancer cell images that were taken at two different time instances 16 and 40 hours. Figs. 3.11(a1), (a2), (c1) & (c2) represent T98 glioblastoma cancer cell images, where the input image has a lot of artifacts such as black spots, fiber, bubbles, and distortions. The proposed textured-based ROI segmentation method is used in this example resulting in robust and accurate segmentation results. The final segmentation results for the outer-regions and ROIs are shown in Figs. 3.11(b1), (b2), (d1) & (d2) for the input images given in Figs. 3.11(a1), (a2), (c1) & (c2), respectively. From the results shown in Figs. 3.10 & 3.11, it can be concluded that the proposed scheme can segment


(a1). Original image. (b1). ROI segmentation of (a1). (c1).Original image. (d1). ROI segmentation of (c1).

(a2). Original image. (b2). ROI segmentation of (a2). (c2).Original image. (d2). ROI segmentation of (c2).

Fig. 3.11. Examples of ROI segmentation for T98 Glioblastoma cancer cell images using the proposed textured-based ROI segmentation.



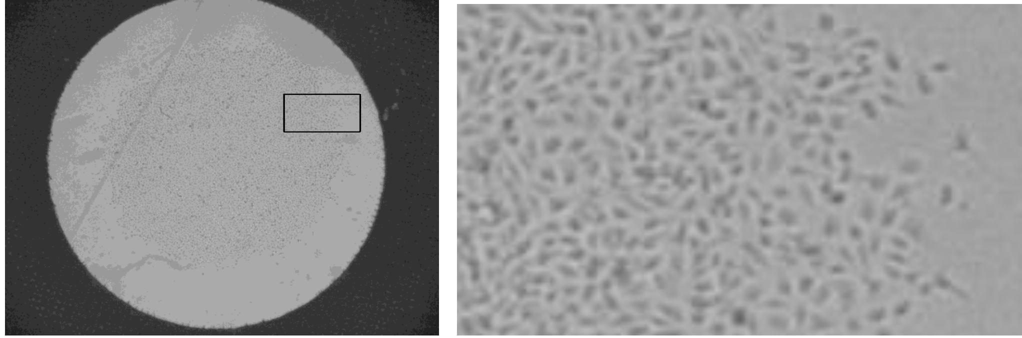
(a). Migration analysis for 253J Bladder cancer cell images. (b). Migration analysis for T98 Glioblastoma cancer cell images.

Fig. 3.12. Cell migration rates for 30 cancer cell images (bladder and Glioblastoma cancer cells) at two different time points.

the ROI areas accurately even if the input images represent different cancer cells and even if the images are corrupted with noise and artifacts. From the segmented ROIs, the cell migration rate for 30 253J-bladder cancer cell images and 30 T98-Glioblastoma cancer cell images are shown in Figs. 3.12(a) and (b), respectively.

3.7. Cell Proliferation and Dispersion Analysis

Cell counting at each time point helps in cell proliferation analysis. In the considered cell images, the cell population inside the ROI is in the range of 1000 to 8000 cells/image. Due to the large number of images and cells, manual counting of cells by humans has proven to be impractical, non-reliable, and exhaustive. Therefore, an automatic and reliable counting procedure is essential to have accurate measurement in a



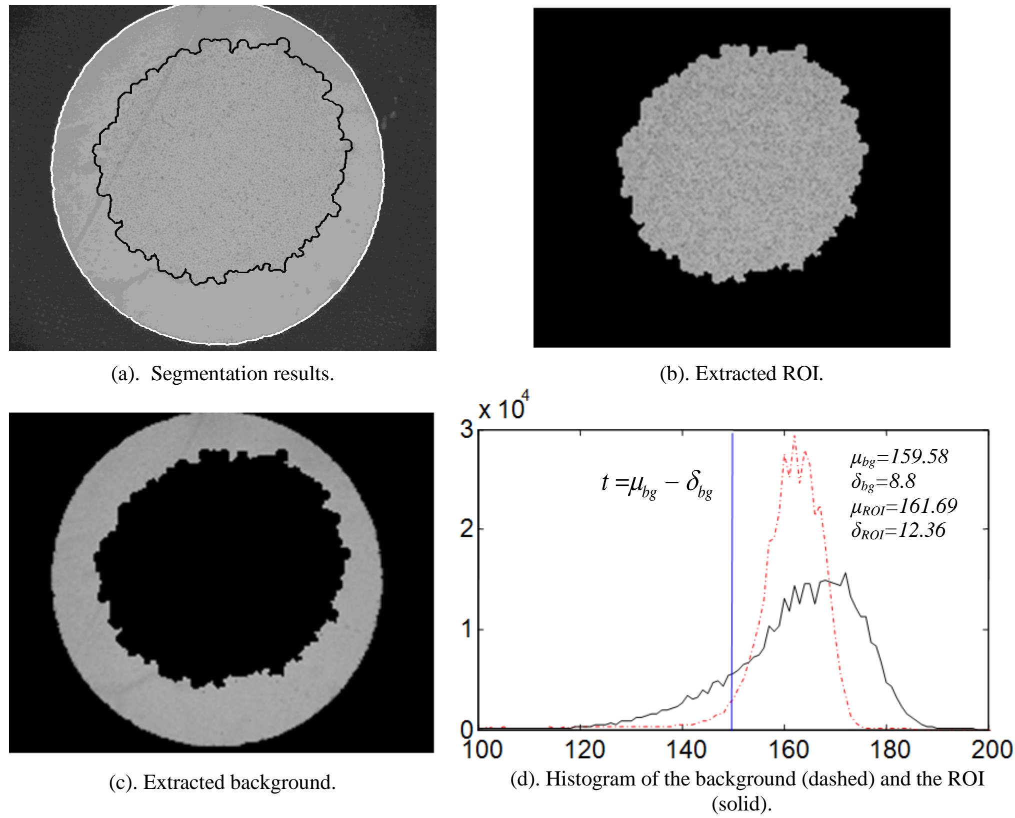
(a). Original bladder cancer cell image. (b). Zoomed in portion in (a) showing individual cells.

Fig. 3.13. Intensities of the individual cells and background are very close in values.

very short time. Several automatic or semi-automatic methods have been proposed to segment the boundaries of the cells in order to count the number of cells. In [138], an *à* trous wavelet transform is used to detect spots in the presence of noise. The method in [138] works under certain assumptions such as a smooth background, a good contrast between cells and background, and disconnected cells. In [139], a comparison is presented for seven thresholding methods for breast tumor cell segmentation, and it was found that Otsu's method performed best. Otsu's method, however, does not give satisfactory results if the image contrast is low. Most of the existing methods fail to segment or count the cells if they are connected. In [127], the images have a high contrast between the cells and background, and cell populations are in the range of 350 to 750 cells/image, which is significantly lower than the cell density in the cell images considered in this work.

3.7.1. Proposed individual cell counting and segmentation method

The main challenges with the considered cancer cell images include the following: heavily populated (high cell density), tiny cells, touching and overlapping cells, presence of noise in the images, and very low contrast as the intensity values of cell boundaries and background are very close. From Fig. 3.13(b), it can be noticed that the cells and background intensities are very close and, therefore, it is hard to find a threshold to distinguish between the background and individual cells. The best way to overcome this



(a). Segmentation results.

(b). Extracted ROI.

(c). Extracted background.

(d). Histogram of the background (dashed) and the ROI (solid).

Fig. 3.14. Segmentation masks and histograms for the ROI and background of a bladder cancer cell image.

problem is to segment the individual cells based on their centroid (nucleus) since, as it can be seen from Fig. 3.13(b), the centroid of the cell has a lower intensity than the cell boundary (halo). Consequently, our strategy is to count the centroids of the cells instead of counting based on the cell boundary [124]; in this way, one can also separate connecting and overlapping cells by their nuclei as long as the nuclei are separated.

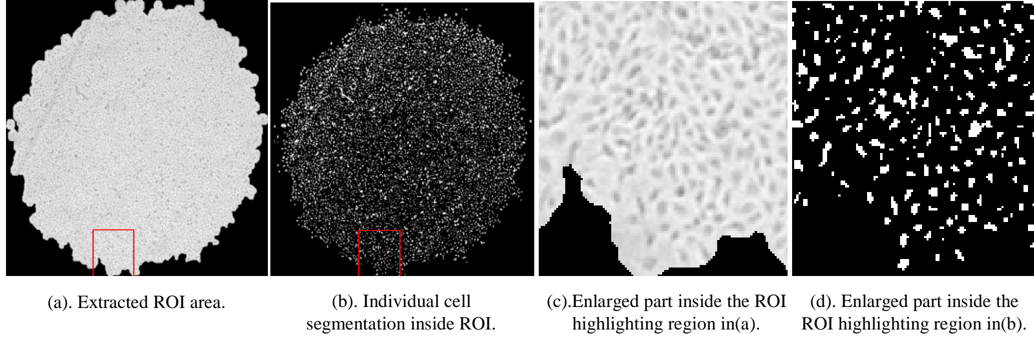
The proposed individual cell segmentation and counting procedure can be summarized as follows:

- From the segmentation masks of the ROI and the outer-circle, we can extract the background region, which represents the region between the ROI and the outer-circle. The ROI and the background masks are shown in Figs. 3.14(b) & 3.14(c), respectively.

- The histogram, mean, and standard deviation of the ROI region and the background region are computed as shown in Fig. 3.14(d).
- From the means and standard deviations of both the background and the ROI, a threshold is computed and used to classify the ROI into significant (cells' centroids) and non-significant regions (Fig. 3.14(f))
- The resulting connected significant regions are counted using a labeling procedure.

The histogram distributions of the background (dashed curve) and ROI (solid curve) are shown in Fig. 3.14(f). It can be seen that the background and the ROI are close in mean values ($\mu_{bg} = 159.58$ and $\mu_{ROI} = 161.69$). However, the background has a small standard deviation ($\delta_{bg} = 8.8$) as compared to the standard deviation of the ROI ($\delta_{ROI} = 12.36$). The increase of the variance inside the ROI, is due to the large population of cells and the intensity variations between the centroid and the surface of each cell. The non-significant regions inside the ROI have similar intensity values to those in the background region. The background region has a relatively low variance due to less concentration of cell clustering inside this region. An accurate threshold can be computed based on the mean and the standard deviation of the background to segment the cells' nuclei (significant ROI regions) from the non-significant ROI background-like regions. The classification threshold is computed as $t = \mu_{bg} - \delta_{bg}$, where μ_{bg} and δ_{bg} represent the mean and standard deviation of the background, respectively. A pixel in the ROI is classified as significant (as belonging to a cell's centroid) if its intensity is less than the threshold t; otherwise, it is classified as non-significant. A labeling process is then applied to count the obtained cell centroids.

Fig. 3.15 gives an example of individual cell segmentation using the proposed cell segmentation and counting method. In Fig. 3.15, the total number of individual cells inside the ROI using the proposed method is around 7400 cells. To test the accuracy of

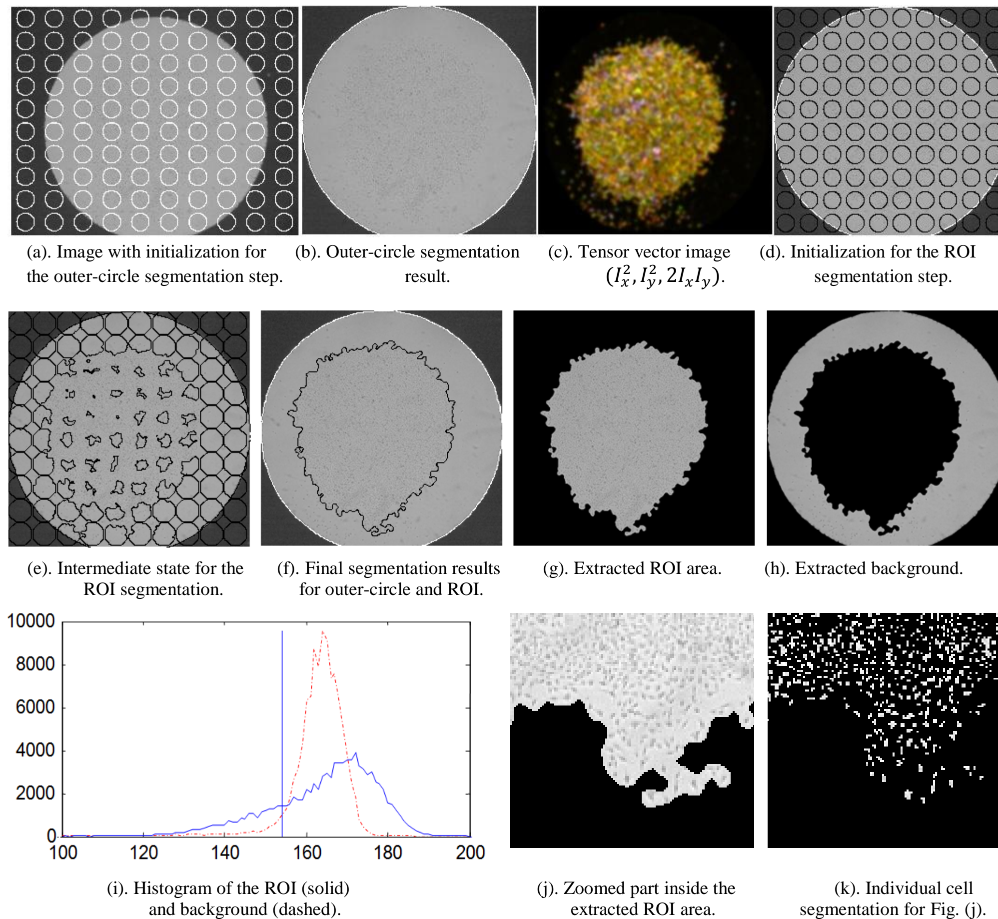


(a). Extracted ROI area. (b). Individual cell segmentation inside ROI. (c).Enlarged part inside the ROI highlighting region in(a). (d). Enlarged part inside the ROI highlighting region in(b).

Fig. 3.15. Process for individual cell segmentation and counting for the bladder cancer cell image shown in Fig. 3.14(a).

the proposed counting method, several subjects were asked to manually count the cells in selected ROI sub-images with different cell concentrations; the average count was then computed for each considered ROI sub-image and compared to the count obtained using the proposed counting scheme. It was found that the proposed counting method achieves an accuracy of 96% on average. In some cases, when the individual cells' centroids overlap, it will be very hard to separate them either manually or using the proposed method. In that case, the method will count the cells with overlapping centroids as one cell.

In the following two examples, the proposed texture-based ROI segmentation scheme is applied to two different bladder cancer cell images using two different initializations followed by individual cell segmentation. Fig. 3.16(a) shows the original image with an initialization for the outer-circle segmentation step. The outer-circle segmentation result is shown in Fig. 3.16(b), where the image size is reduced to fit only the outer-circle mask, which helps in speeding up the process in the second ROI segmentation step. Fig. 3.16(c) shows the regularized texture tensor image $(I_x^2, I_y^2, 2I_xI_y)$. The initialization for the second ROI segmentation step is shown in Fig. 3.16(d), and an intermediate segmentation result is shown in Fig. 3.16(e). The final segmentation masks, the extracted ROI, and the extracted background are shown in Figs. 3.16(f), (g), and (h), respectively. Fig. 3.16(i) shows the histogram distribution of the extracted ROI and the extracted background.



(a). Image with initialization for the outer-circle segmentation step. (b). Outer-circle segmentation result. (c). Tensor vector image $(I_x^2, I_y^2, 2I_xI_y)$. (d). Initialization for the ROI segmentation step.

(e). Intermediate state for the ROI segmentation. (f). Final segmentation results for outer-circle and ROI. (g). Extracted ROI area. (h). Extracted background.

(i). Histogram of the ROI (solid) and background (dashed). (j). Zoomed part inside the extracted ROI area. (k). Individual cell segmentation for Fig. (j).

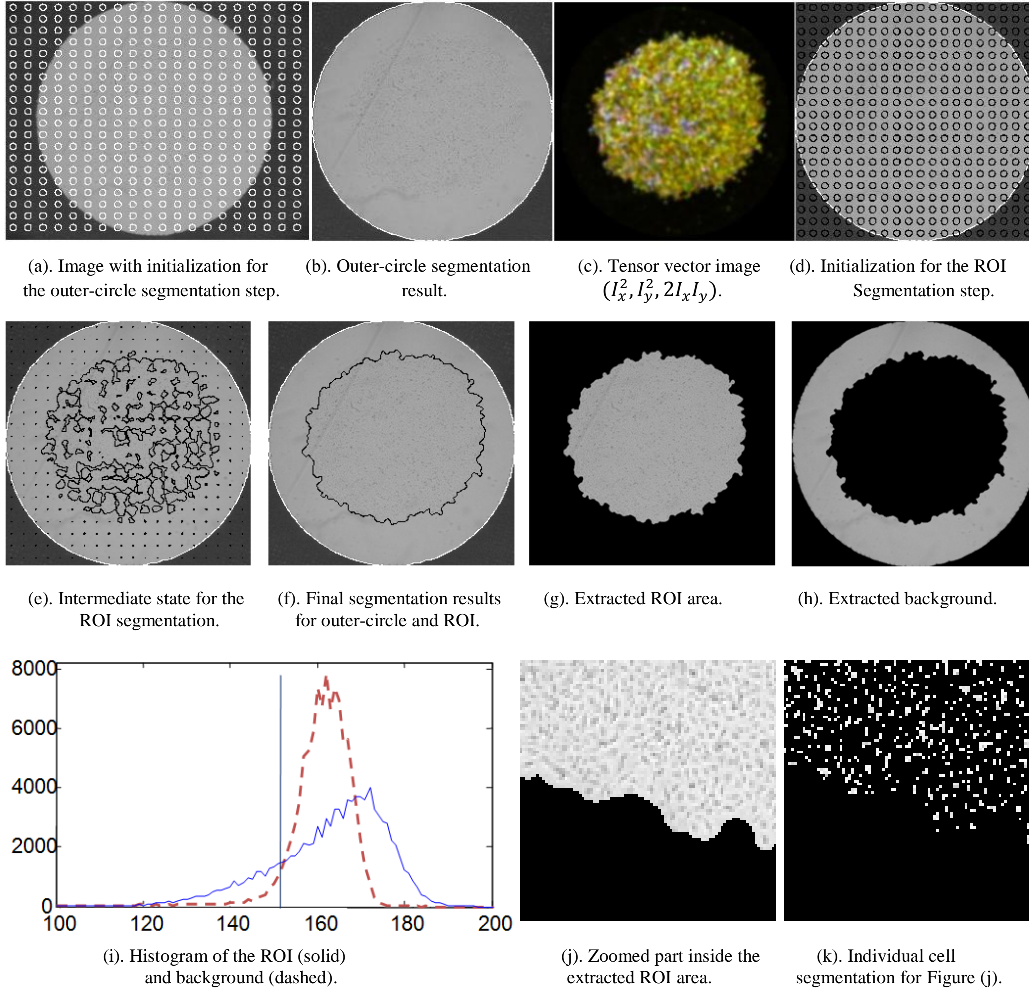
Fig. 3.16. Example 1: Texture-based ROI segmentation followed by individual cell segmentation for a cancer cell image.

Fig. 3.16(k) illustrates the individual cell segmentation results for a zoomed area inside the ROI that is shown in Fig. 3.16(j). Figs. 3.17(a)-(k) illustrate the same process as in the previous example but for a different image and different initialization.

3.7.2. Cell dispersion analysis

Cell dispersion analysis is applied to the cells in the background area (the region outside the cell cluster region and inside the outer-circle). First, the individual cells in the background area are segmented by using the same procedure that is used to segment the individual cells inside the ROI area. Then, the following two approaches are used to perform the cell dispersion analysis:

- Number of cells versus distances from the ROI's centroid: this can be represented by a curve, where values on the horizontal axis represent quantized distances bins



(a). Image with initialization for the outer-circle segmentation step. (b). Outer-circle segmentation result. (c). Tensor vector image $(I_x^2, I_y^2, 2I_xI_y)$. (d). Initialization for the ROI Segmentation step.

(e). Intermediate state for the ROI segmentation. (f). Final segmentation results for outer-circle and ROI. (g). Extracted ROI area. (h). Extracted background.

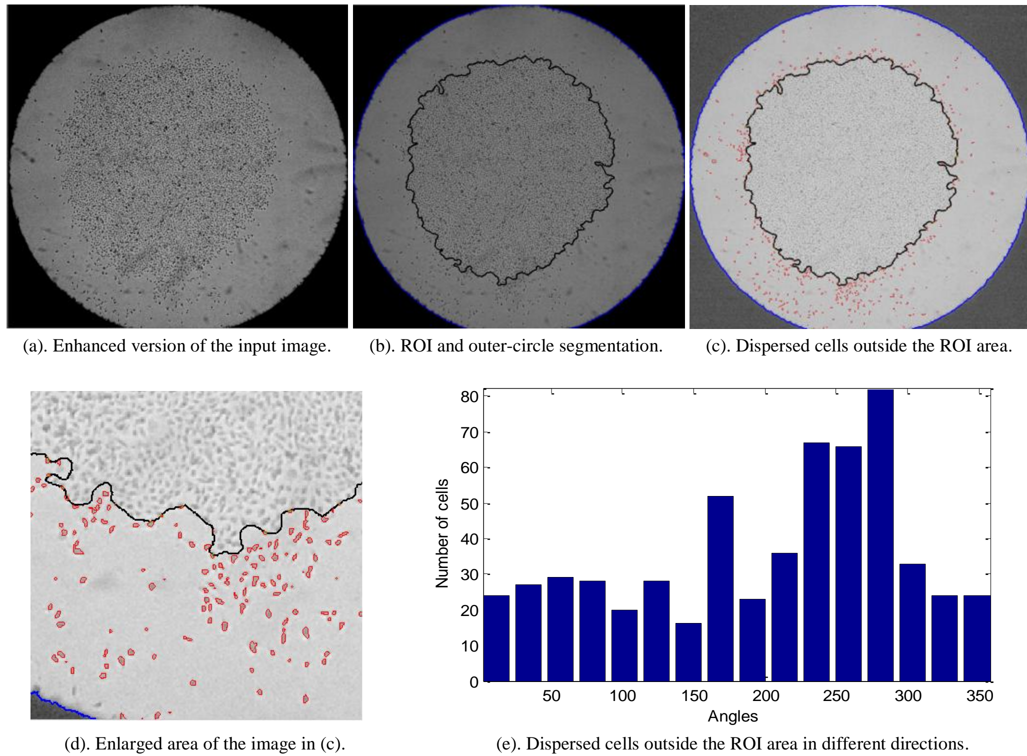
(i). Histogram of the ROI (solid) and background (dashed). (j). Zoomed part inside the extracted ROI area. (k). Individual cell segmentation for Figure (j).

Fig. 3.17. Example 2: Texture-based ROI segmentation followed by individual cell segmentation for a cancer cell image.

representing, each, a distance range from the ROI's boundary, and where the vertical axis represents the total number of cells within a distance bin from the ROI's centroid.

- Number of cells in a certain direction: in this case, the region outside the ROI is divided into N sectors, with respect to the ROI's centroid, with an angle equal to $2\pi/N$ for each sector.

The combination of those two approaches will give information about the cell dispersion in terms of number of cells in a certain direction and within a certain distance from the



(a). Enhanced version of the input image. (b). ROI and outer-circle segmentation. (c). Dispersed cells outside the ROI area.

(d). Enlarged area of the image in (c). (e). Dispersed cells outside the ROI area in different directions.

Fig. 3.18. Cell dispersion analysis.

ROI's boundary. The dispersed cell distribution helps in studying the regular/irregular migration of the dispersed cells.

An example of the proposed cell dispersion analysis is shown in Fig. 3.18. An enhanced version of the input image is shown in Fig. 3.18(a), the ROI and the outer-circle segmentations are shown in Fig. 3.18(b), and the segmentation of the dispersed cells outside the ROI is shown in Fig. 3.18(c). An enlarged portion of the image given in Fig. 3.18(c) is shown in Fig. 3.18(d). The numbers of dispersed cells with respect to different directions are shown in Fig. 3.18(e), where the area around the centroid of the ROI is divided into *16* sections with equal angles (22.5^{o}), then the dispersed cells are segmented and counted in each section in the region between the ROI boundary and the outer-circle. From Fig. 3.18(e), it can be seen that the dispersed cells are concentrated around angle 270°. From the dispersed cell distribution with respect to angles, one can classify the migration of the dispersed cells as regular or irregular migration. The cell

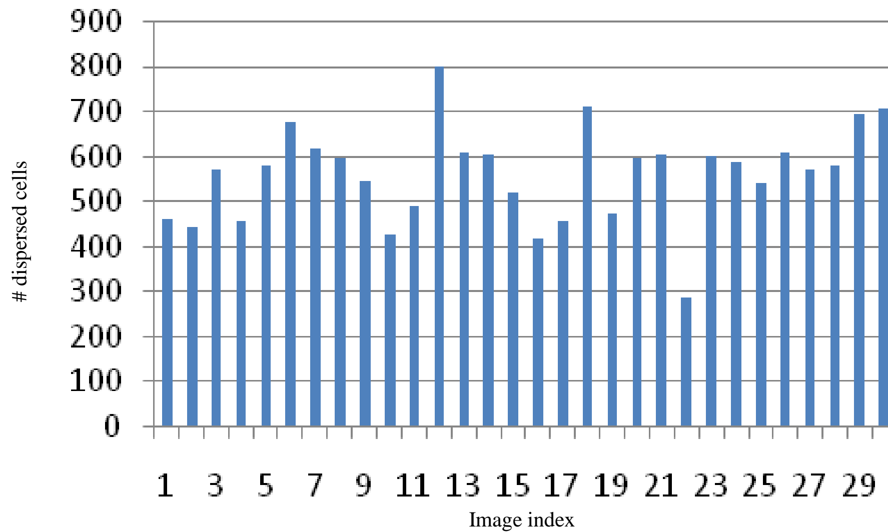


Fig. 3.19. Cell dispersion for 30 bladder cancer cell images after 16 hours.

dispersion analysis for 30 bladder cancer cell images at a time point equal to 16 hours is shown in Fig. 3.19. In each image sample, the total number of dispersed cells outside the ROI is computed.

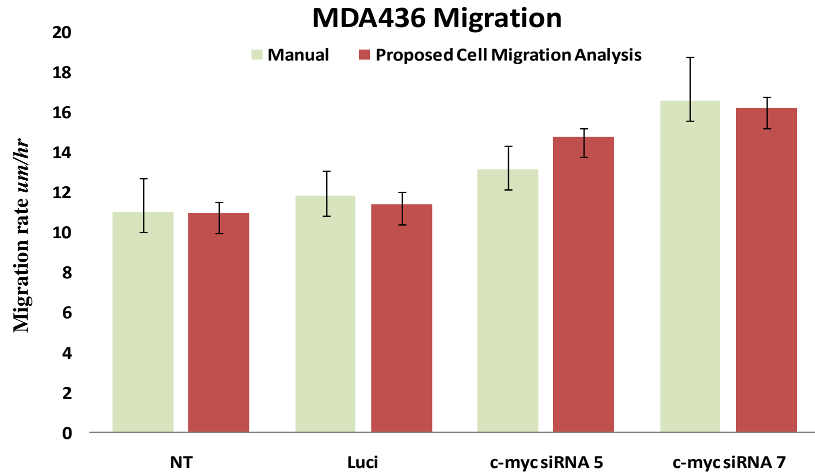
3.8. Simulation Results

Motility rates of human cancer cells were assessed [140] in vitro using a radial monolayer migration assay [141]. Epithelial cancer cells, including breast cancer (MDA436) and pancreatic cancer (PANC-1), and neural crest-derived cell lines, including Glioblastoma multiforme and melanoma (UACC903), were evaluated for changes in migration rates. Different treatments were used for the aforementioned cancer cells to study the effect of each treatment on the migration rate after 24 hours. The treatments [140, 141] that were used in the experiment (NT, Luci, c-mys siRNA 5, and c-mys siRNA 7) are used for each cancer cell line. Digital images of populations of cells in the radial dispersion assay [142] were captured at 0 and 24 hrs post seeding. Images were analyzed by eye (through expert operators) for the radius of a putative circle drawn around the cell culture region. The manual results were compared with the results obtained from the proposed automated method as given in Table 3.1 and Fig. 3.20. In this

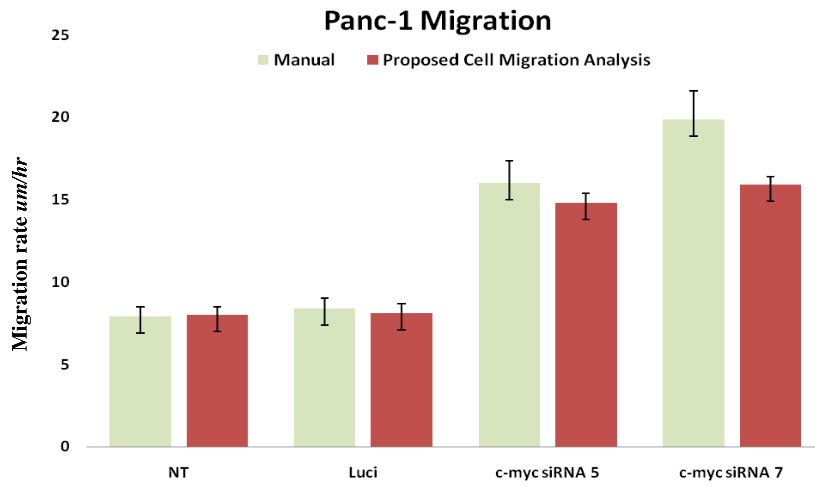
Table 3.1: Comparison between manual and proposed method.

Cell line	Treatment	Manual		Proposed Cell Migration Analysis	
		Avg	Std	Avg	Std
MDA436 Migration	NT	11.0119	1.7057	11.0	0.5
	Luci	11.8686	1.2249	11.4	0.6
	c-myc siRNA 5	13.1773	1.1579	14.8	0.4
	c-myc siRNA 7	16.5640	2.2161	16.2	0.6
Panc-1 Migration	NT	7.8941	0.5817	8.0	0.5
	Luci	8.3805	0.6253	8.1	0.6
	c-myc siRNA 5	16.0099	1.3433	14.8	0.6
	c-myc siRNA 7	19.8481	1.7441	15.9	0.5
UACC903 Migration	NT	8.8465	1.2108	10.2	0.5
	Luci	9.0727	1.0798	10.5	0.4
	c-myc siRNA 5	13.2923	1.2609	15.2	0.5
	c-myc siRNA 7	15.1186	1.6637	15.0	0.5

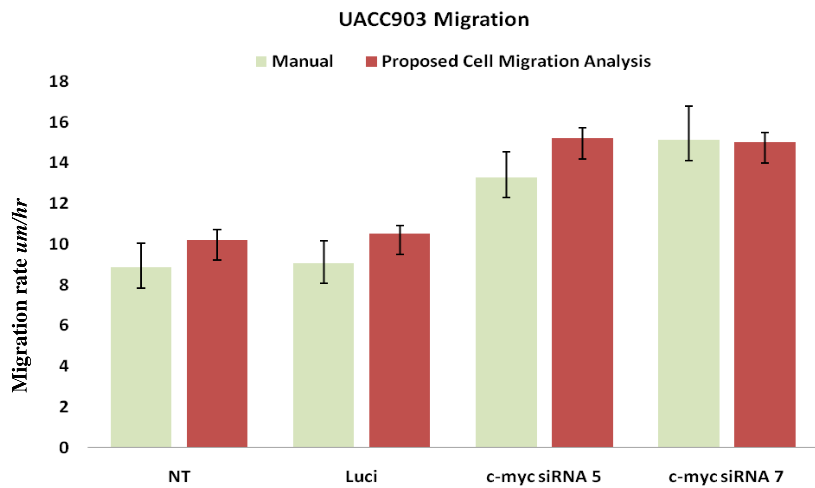
case, the migration results are expressed in $\mu m/hr$ instead of $\mu m^2/hr$. This was done by fitting the area of the segmented ROI (A_{ROI}) into a circle with an area ($\pi\, r^2$) and, then, the circle radius is calculated as $r = \sqrt{A_{ROI}/\pi}$ to track the migration rate in $\mu m/hr$. From the results in Table 3.1 and Fig. 3.20, it can be concluded that the proposed automated method gives high correlation with the manual data. In addition, the proposed method results in a significantly lower standard deviation as compared to the manual calculation when an initial calibration step is performed using one sample image before batch processing [140]. The cell migration analysis using the proposed method takes milliseconds to process one image. On the other hand, it takes several minutes to analyze the migration of cells in one image manually by the operator. It can be seen from Fig. 3.20 that some treatments (drugs) decrease the migration of cancer cells while other treatments promote an increase in the migration rate of the cancer cells. This experiment helps to understand the effect of different treatments on the migration of cancer cells, which plays an important role in drug discovery.



(a). Breast cancer (MDA436).



(b). Pancreatic cancer (panc-1).



(c). Glioblastoma and melanoma cancer (UACC903).

Fig. 3.20. Cell migration analysis (*um/hr*) using manual versus proposed method for different cancer cell images with different treatments.

3.9. Summary

An automatic cell evolution analysis (CEA) system was presented in this chapter. The proposed CEA system includes components for cell migration, cell proliferation and cell dispersion analysis. The proposed texture-based ROI segmentation method is robust to noise and natural artifacts in the considered cancer cell images. Moreover, it gives accurate ROI segmentation. Cell proliferation analysis is achieved by segmenting and counting the individual cells inside the ROI using a proposed threshold based on the histogram analysis of the background and ROI areas. The cell proliferation analysis step is able to segment and count the individual cells inside the ROI even if the cells are touching, noisy, or overlapping partially by their boundaries. Cell dispersion analysis is achieved by segmenting and counting the migrating individual cells outside the ROI region in different directions, or at different distances from the ROI boundary. The proposed scheme was successfully applied on different types of cancer cell images. The proposed scheme was able to get accurate results even if the input images suffer from poor contrast, cell overlap, and are corrupted with artifacts and noise.

CHAPTER 4: AUTOMATED DEFECT DETECTION AND CLASSIFICATION IN SEMICONDUCTOR UNIT IMAGES

4.1. Introduction

The assembly test process has many steps where defects can be created. The manual handling of parts in carriers and trays causes some of these defects. Other defects are caused by machine faults or, in some cases, a less than optimally clean environment. These defects can result in value being added to an already "dead" unit or in the worst case, a customer return of a unit that does not meet specifications. Assembly and test manufacturing has many critical steps which must be conducted within a short period of time. Defects can occur at any time during the process. The defect detection and classification system must be capable of operating while carriers or trays are in motion during the process so as not to interrupt critical process steps. Catching defects close to their occurrence enables performance of timely root cause analysis to get to the source of a problem and reduces the potential for a large excursion instead of only a few defective units. Manual inspection efforts are missing visual defects that are causing field failures and/or are an aesthetic issue. Operators take time to get trained in the recognition of these defects. An operator gets tired after a very short period of time performing the detailed final visual inspection and, as a result, it is difficult to maintain alertness and also operators who are motivated to perform this work.

Automatic defect detection and classification is an essential step in the industry to locate and isolate defective parts in an early stage during the inspection process before sending products to customers. Therefore, this not only results in customers' satisfaction and reduces the product return, but also helps to improve the productivities and reliabilities of the products. Studying the defect detection and classification helps also to know what the most frequent defects in a certain product are and, therefore, enables engineers in knowing how to reduce the occurrence of those defects.

Three automated defect detection and classification methods are proposed and will be discussed in this chapter. Two different applications to automatically detect and classify the defects that are related to solder joints or solder balls are considered including: (i)“Non-Wet” defect which occurs in the solder joints in processor sockets and can cause motherboard failures; and (ii)“Void” defect which occurs in the solder joints and solder balls and can cause incorrect scrapping and rework. The third proposed method is used for defect detection and classification on the die area of the semiconductor unit. Some of the defects related to the die area include scratches, cracks, foreign materials (FM), and stains. The proposed methods [50-54] give high accuracy and are inexpensive to implement compared to the results obtained by state-of-the-art x-ray machines.

This chapter is organized as follows. An automated non-wet defect detection method is presented in Section 4.2 for solder joints of processor sockets. Void detection in solder balls is presented in Section 4.3. In Section 4.4, a proposed scheme for the detection and classification of defects in the die area of the semiconductor units is presented.

4.2. Automated Non-Wet Defect Detection in Solder Joints

Non-wet solder joints in processor sockets are causing mother board failures. These board failures can escape to customers resulting in returns and dissatisfaction. The current process to identify these non-wets is to use a 2D or advanced x-ray tool with multi-dimension capability to image the solder joints of processor sockets. The images are then examined by an operator who determines if each individual joint is defective or non-defective. There can be an average of 150 images for an operator to examine for each socket. Each image contains more than 30 joints. These factors make the inspection process time consuming and the output variable depending on the skill and alertness of the operator. This section presents an automatic defect detection and classification system to differentiate between defective (non-wet) and non-defective solder joints. The main components of the proposed system consist of region of interest (ROI) segmentation,

feature extraction, reference-free classification, and automatic mapping. The ROI segmentation process is a noise-resilient segmentation method for the joint area. The centroids of the segmented joints (ROIs) are used as feature parameters to detect the suspect joints. The proposed reference-free classification can detect defective joints in the considered images with high accuracy without the need for training data or reference images. An automatic mapping procedure which maps the positions of all joints to a known Master Ball Grid Array file is used to get the precise label and location of the suspect joint for display to the operator and collection of non-wet statistics. The accuracy of the proposed system was determined to be 95.8% based on the examination of 56 sockets (76,496 joints). The false alarm rate is 1.1%. In comparison, the detection rates of currently available advanced x-ray tools with multi-dimension capability are in the range of 43% to 75%. The proposed method reduces the operator effort to examine individual images by 89.6% by presenting only images with suspect joints for inspection. When non-wet joints are missed, the presented system has been shown to identify the neighboring joints. This fact provides the operator with the capability to make 100% detection of all non-wets when utilizing a user interface that highlights the suspect joint area. The system works with a 2D x-ray imaging device, which saves cost over more expensive advanced x-ray tools with multi-dimension capability. The proposed scheme is relatively inexpensive to implement, easy to set up and can work with a variety of 2D x-ray tools.

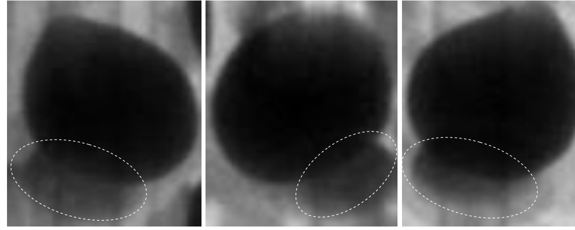


Fig. 4.1: Examples of defective solder joints (non-wet).

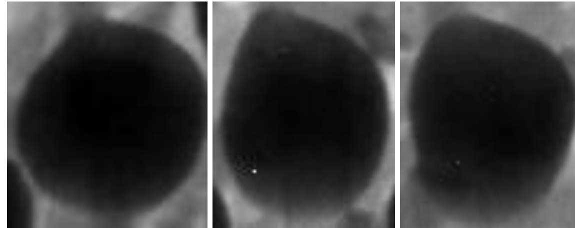


Fig. 4.2: Examples of non-defective solder joints.

4.2.1. Problem statement

A non-wet solder joint is the joint where the solder has not fully flowed to make solid contact between the unit and the board thus leaving a potentially weak or totally open signal path. Non-wet solder joints are difficult to detect using manual inspection alone. One current solution to this problem involves the set up and programming of a 2D x-ray machine to image the processor sockets of the board. The images of the sockets are taken at oblique angles to make the non-wet joints visible to the operator. The operator must then look through a large set of 100 to 200 images in order to identify all the non-wet joints in a socket. Each image can contain between 10 and 45 solder joints. This is a very time consuming and error-prone process due to the limited number of gray scale levels in the acquired images and the repetitive nature of the images. Examples of defective (non-wet) and non-defective solder joints are shown in Fig. 4.1 and Fig. 4.2, respectively. The non-wet is seen as the shadow underneath the solder joint as shown in Fig. 4.1. The operators have difficulty identifying the non-wet joints accurately, with repeatability, and in a timely manner, with so many images per socket and multiple sockets on a board to be examined. The constant checking of homogenous images is fatiguing and can result not only in missed detections but also in ergonomic issues. The current manual process takes more than 35 minutes from the start of imaging to the completion of an inspection of a single socket. A means of automatically detecting suspect joints and classifying the defects on those units is needed to mitigate these issues. This can significantly decrease the number of images that need to be manually checked and can thus allow the engineer to more quickly and easily isolate process issues.

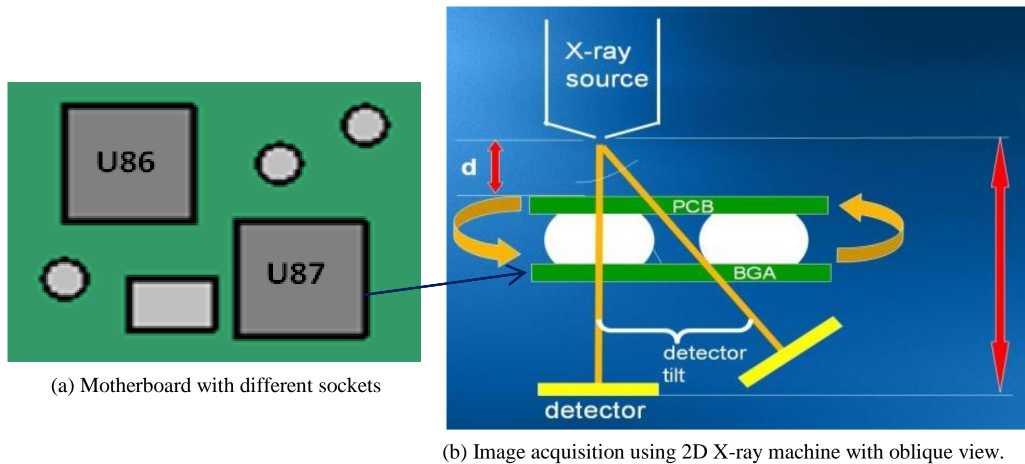
Our investigation of other relevant work showed that the proposed system components are novel and unique and have not been proposed previously in the literature. None of the existing methods of non-wet detection are using the techniques we are proposing in this chapter. There are many existing efforts [143-162] that deal with

different issues in solder joints such as getting better image quality using x-ray machines, cause of damage and defects in Ball grid array (BGA), failure analysis of BGA packages, assembly quality, physical properties of solder balls and defect detection in different types of solder joints (different applications). Relevant work that deals with defect detection in solder joints was presented in [143-150]. In [143], different images of solder joints were taken in slices of a Ball Grid Array (pad, ball, package slices) to be used for the classification process. The method requires manual calibration and calculation of parameters at the beginning of the experiment before detecting defects in the Ball Grid Array (BGA) images. Some of these parameters are ball diameters at different slices, centroids at ball and pad slices, distances between pad, ball and package slices. These parameters are compared between different slices to detect defective joints. The method used in [143] is a reference-based method since it requires knowledge of the diameters of the non-defective joints in different slices in addition to specific setups and calibrations. In [144-146], the authors proposed a scheme for defect detection in BGA. The images were taken by using a computed tomography scanner where the resulting imaged joints have non-defective joints of regular shape and size. The tomography scanner is very time consuming to use and as such cannot be inserted into a manufacturing process without severely impacting the inspection times. In [2-4], the shape of a non-defective solder joint is assumed to closely correspond to a circle. The solder joint area, which is called Region of Interest (ROI), is segmented by using a fixed threshold value equal to 54 to get a binary image. After the ROI segmentation, a roundness factor (compactness) [10] is used for classification which is based on the assumption that the non-defective joint has a circular shape. If the shape of the considered joint deviates significantly from a circle, the joint is classified as defective. One drawback of the method of [2-4] is that it is not robust to illumination changes, which occur in practice due to the lighting variation during the image acquisition process. In [147], a scheme is proposed for defect detection in BGA.

The scheme of [5] is similar to the scheme of [144-146], except that the threshold value, which is used to convert the grayscale image to a binary image, can be automatically determined based on the mean and variance of the considered image. Then, a coarseness factor is used to classify each segmented joint as a defective or non-defective joint. The coarseness factor is based on the ratio between the areas of the solder joint and the circle containing the solder joint. The coarseness factor in [147] is similar to the compactness factor in [144-146] in that they are both based on the assumption that a non-defective joint has a circular shape. If the defective joint in [144-147] has a shape very close to a circle, the method will not be able to detect the defective joint. If non-defective joints have slightly irregular shapes, they will be classified as defective using the methods in [144-147]. The imaging process can often introduce changes due to lighting and positioning of the unit undergoing inspection so it is more desirable to have a method that does not depend on the specific shape of the joints. In [148-150], a scheme is proposed for defect classification in solder joints by using a neural network for training and classification. The method makes use of a neural network for training different types of defects in solder joints by using a training set of images that represent different types of defects. During the training process, feature parameters are extracted first from the shape and/or histogram of the joints. A circular shape is also used here to model non-defective joints and a non-circular shape is used to model defective joints. In this latter case, a library of defective and non-defective joints is generated by an expert operator. The generated library and a lookup table that is formed during the training process are then used to classify the solder joints into 3 categories: normal, insufficient, and excessive soldering. The correct classification rate of this method depends heavily on the generated library and on the accuracy of the training process. The generation of the library and training data is time consuming. The processing associated with using the library to detect the defective joints is intensive and could result in the method not being usable in

real time depending on the implementation. It should be noted that none of the existing techniques [1-8] deal with non-wet solder joints, but rather these methods are concerned with other types of defects and they cannot be applied to the detection of non-wet joints, which is the focus of this section.

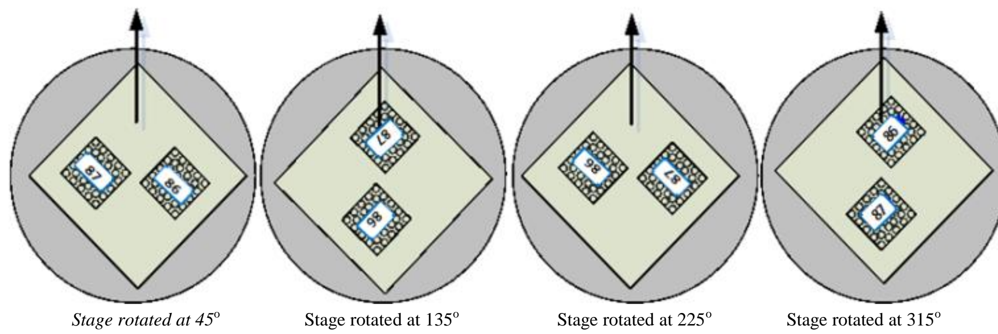
There are some state-of-the-art x-ray machines that have embedded algorithms that detect defects in solder joints, for example, advanced x-ray tools with multi-dimension capability used on many factory floors. However, the advanced capability x-ray machines with embedded algorithms typically produce a 43% to 75% detection rate with an operator in the loop locating the non-wet joints. The advanced capability x-ray machines require manual tuning during the setup and, in addition, are 5 to 10 times the cost of the 2D x-ray machine. In this work, the interest is to detect the defective solder joints automatically from images that are taken by using a low cost 2D x-ray imaging tool that provides images in a timely manner during the manufacturing process. The x-ray tool makes the non-wet joints visible to the human eye if the images are taken at oblique angles. The non-wet shows up as a shadow extending below the body of the joint as shown in Fig. 4.1. For comparison, Fig. 4.2 shows examples of non-defective joints. Taking images with an angle perpendicular to the processor socket will not show the non-wet joints because the x-ray is looking directly at the joints and, thus, the non-wets are not distinguishable from the good joints. The images are taken while the x-ray collector was set at 46° degrees and the socket was rotated at 4 different oblique angles: 45°, 135°, 225°, and 315° as shown in Figs. 4.3 and 4.4. These angles were experimentally determined by the engineer to be suitable angles to see the defect areas clearly. However, taking images at oblique angles does not keep the consistency of the size and shape of the imaged solder joints but introduces shape and size variations across images as well as within a single image.



(a) Motherboard with different sockets

(b) Image acquisition using 2D X-ray machine with oblique view.

Fig. 4.3. Image acquisition using 2D x-ray machine.



Stage rotated at 45° Stage rotated at 135° Stage rotated at 225° Stage rotated at 315°

Fig. 4.4. Rotation motion of the x-ray stage in the X-Y plane during inspection.

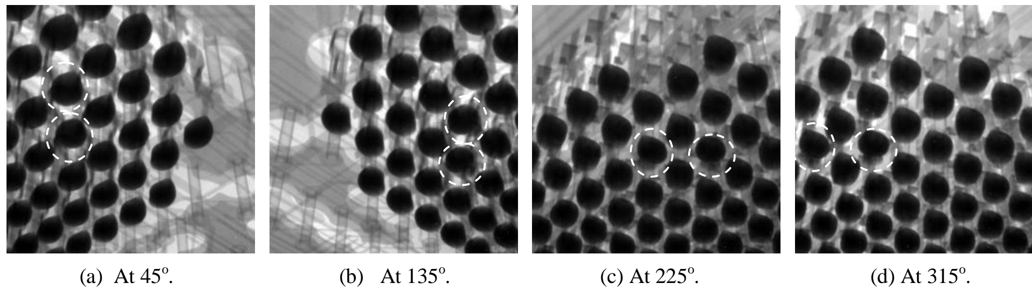
None of the above methods [143-150] are suitable for our considered x-ray images with oblique angles, because, in our case, the imaged joints do not have consistent shapes and the shapes of the joints vary as a function of their position within the image and the camera angle (circle, egg-shaped, ellipse, and other shapes). Thus, the circular shape assumption for non-defective joints is not valid in our case. Moreover, the solder joint images captured at oblique angles present other challenges including inconsistency in size, lighting variation, image blurring, overlapping between neighboring joints and the repetitive pattern of the joints. The images are also corrupted with noise during the image capturing process. Using existing classification methods for the considered images will produce a high false positive rate and would require a large training data set to create a library that represents all possibilities of the existing defects. Moreover, it is time consuming to create the library and perform the training. Since none of the existing

methods [143-150] can deal with the aforementioned issues in the considered images, it is desirable to develop a robust method to automatically detect suspect joints that are imaged at oblique angles using a 2D x-ray tool. In the following subsections, an automatic defect identification and classification scheme is presented for solder joints in processor sockets. The proposed defect detection scheme is reference-free in the sense that it does not make use of known training data or reference images to identify defective joints in processor sockets. The primary goal of this work is the identification of suspect joints in processor sockets during the manufacturing process, before they leave the factory and result in returns. Another goal is to assist engineers with rapid, new technology development through early identification of non-wet issues occurring during the manufacturing process. This work accomplishes this critical early assistance by enabling a thorough identification of non-wet joints thus allowing the engineer to identify process issues quickly and correct them.

The following subsections are used to show the procedures of the proposed method for non-wet detection. Section 4.2.2 presents the multi-view x-ray imaging set-up, Section 4.2.3 presents the proposed automatic defect detection and classification scheme, Section 4.2.4 presents an automatic joint mapping procedure to uniquely label the detected defective joints, and the experimental results are presented in Section 4.2.5 in order to illustrate the performance of the proposed system.

4.2.2. Multi-View x-ray imaging set-up

As shown in Fig. 4.3, the image capture process begins with the operator placing the board to be inspected onto a stage with a fixture that holds the board inside the 2D x-ray machine. The fixture is a circular stage with a square hole in the middle with a clamp that permits various sizes of boards to be mounted securely in the fixture. The board is placed on the stage with the component side down in order to avoid contact between the x-ray source and the board components and, thus, possible damage to the components. The



(a) At 45°. (b) At 135°. (c) At 225°. (d) At 315°.

Fig. 4.5. Sample images obtained at different oblique angles using the x-ray machine. Dotted circles are used to highlight non-wet joints.

collector can be moved to focus the collection as needed. The operator selects the image capture program to be run based on the socket to be inspected. The operator enters parameters manually into the machine to provide the serial number of the board and the socket being inspected. The 2D x-ray machine executes the program and automatically places each image taken in a file folder with the name entered by the operator. The x-ray collector is set at a 46° degree angle as shown in Fig. 4.3 and the x-ray stage moves through four angles for inspection as shown in Fig. 4.4. The stage is rotated in this way to allow the x-ray to "look through" the sockets at multiples of 45 degrees to maximize the detection of non-wet joints. Once the stage has been positioned at the desired angle for inspection, the board is moved horizontally and vertically to allow images to be taken for the entire socket perimeter. An example of two different sockets in one board is shown in Figs. 4.3 and 4.4. Fig. 4.5 gives an example of x-ray images which were taken at different angles. The images in Fig. 4.5 show the same socket corner but at four different angles: 45°, 135°, 225°, and 315°. Using 4 angles means that each joint has a chance to be shown at least 4 times and maybe more due to the overlap between the images during the scanning process. One might not be able to see the defect in a joint in all four views, because this depends on the location of the defect. The non-wet joints are highlighted by using dotted circles in Fig. 4.5, where the non-wet joints can be seen clearly as a shadow underneath the joints at the angles of 225°(Fig. 4.5(c)) and 315° (Fig. 4.5(d)). One can notice the difference between the non-wet joints and the good joints if one visually

inspects the images very thoroughly. However, the repetitive pattern of the joints makes it hard for the operator to quickly notice the non-wet joints. As indicated previously, one main goal of this work is to develop an automated detection method that can assist the operator by significantly reducing the number of images to be inspected to a small subset of images that contain suspect joints.

4.2.3. Proposed automatic defect classification scheme for solder joints

This section describes the proposed automatic defect detection scheme for solder joint inspection. The main idea behind the proposed method is to segment the area of the solder joint, which we refer to as the Region of Interest (ROI), and extract some features that are crucial for detecting the suspect joints. One property of the imaged solder joints in processor sockets is that they are aligned in parallel horizontal lines and parallel vertical lines when the images are captured at a 90° angle relative to the board (top view). However, as stated before, top-view images do not provide a good detection capability for non-wets as these become hard to distinguish from the non-defective joints. Taking images with oblique angles helps in distinguishing non-wet joints from the good ones. However, in this latter case, the joints are no longer aligned on vertical and horizontal lines. The solder joints are still aligned in straight lines; however, these lines are not necessarily parallel. Also, the imaged joints exhibit variations in shape and size within the same image and across images due to perspective effects when imaging at oblique angles.

The proposed system consists of novel components including: 1) a novel adaptive ROI segmentation that is resilient to noise and illumination changes in order to extract the non-wet joints and in which the segmentation threshold that minimizes the segmentation error is found through statistical modeling using a mixture of Gaussian distributions; 2) a novel no-reference classification component in which each extracted solder joint area is classified as defective or non-defective using a novel model-based procedure that exploits

the fact that the centroids of the solder joints should be aligned in two directions; for this purpose, a novel mathematical model-based procedure was devised to extract the centroids of the segmented joint areas and to find the directions that best fit the extracted centroids (directional clusters) despite the challenges due to imaging at oblique angles and the fact that the joints are no longer aligned on parallel lines due to perspective distortions; for this purpose, angular deviations are computed locally for each joint's centroid and are first used in determining the directional cluster to which each joint belongs; the computed local angular deviations are then used in classifying the considered joint into defective (when the deviation with respect to other centroids in the considered directional cluster, is significant) and non-defective (when the deviation is insignificant); 3) the third component is an automatic optional mapping procedure which finally maps the positions of all joints to a known master ball grid array.

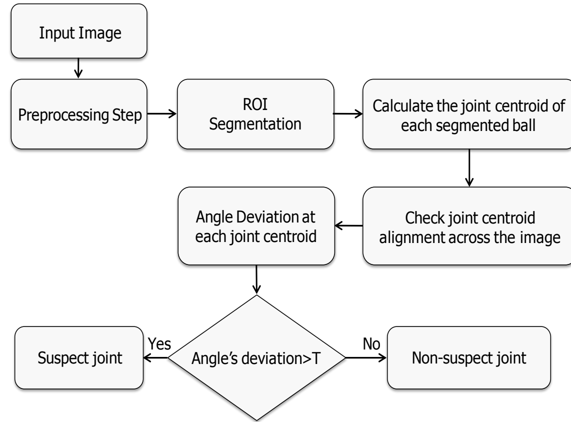
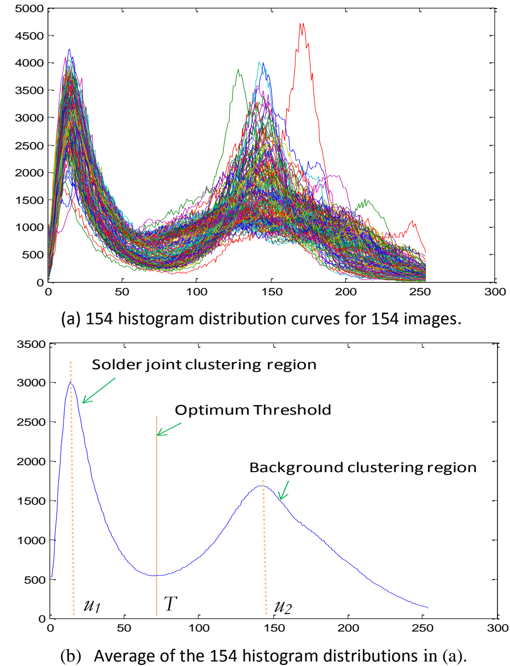


Fig. 4.6. Block diagram of the proposed automatic defect classification scheme for non-wet joints in processor socket.

A block diagram of the proposed non-reference scheme is shown in Fig. 4.6. First, the input image is pre-processed in order to reduce the artifacts that are introduced by the x-ray machine during the image capture process. Most notably, the image contains salt and pepper noise in the form of black and white dots. The pre-processing step consists of applying a simple and fast 3x3 median filter [10, 163] which can effectively remove this type of noise without significantly blurring the edges in the image. The joints in the image are then segmented using a proposed adaptive histogram-based thresholding procedure that is robust to illumination variations within the image. This is followed by the computation of the centroids of the segmented joints and by checking the alignment

of those joints' centroids in order to detect suspect joints. Details about the ROI segmentation, joint centroids computation, joint centroids alignment, and the centroid-based classification of the joints, are given below.



(a) 154 histogram distribution curves for 154 images.

(b) Average of the 154 histogram distributions in (a).

Fig. 4.7. Thresholding using histogram analysis.

4.2.3.1. ROI segmentation

The purpose of this step is to find an accurate method to segment the solder joint area. The captured solder joint images do not exhibit uniform lighting and, thus, using a fixed threshold for solder joint segmentation fails to properly segment the desired joints. In this work, an automatic thresholding method which is based on histogram analysis is used to segment the solder joints. Fig. 4.7(a) shows an example of the 154 histogram curves from 154 images of one processor socket. The curves are all similar in gray level distribution. Taking the average of the 154 histograms produces a single histogram which contains two distinct (cluster) regions represented by two distinct peaks as shown in Fig. 4.7(b). The two resulting regions correspond to the background and the solder joint regions, respectively. The solder joint area has lower gray level values (left part of the histogram in Fig. 4.7(b)) as compared to the background area (right part of the histogram in Fig. 4.7(b)). The two cluster regions that can be seen in the histogram of Fig. 4.7(b) can be represented by a mixture of two Gaussian distributions with two different means and variance parameters. In order to segment the two regions, an automatic threshold should be selected adaptively for each image midway between the mean values of the two Gaussian distribution functions. In order to compute the segmentation threshold for each image, the means of the Gaussian distributions need to be estimated first. For this

purpose, we adopt an iterative procedure to calculate, adaptively for each image, the mean values which are then used to compute the segmentation threshold. The steps of the procedure for computing the mean values of the two Gaussian distributions and the segmentation threshold for each image can be summarized as follows:

- *Step 1:* Compute the probability p_i from the histogram distribution of the input image $h(i)$, where i denotes the 8-bit image intensity, as:

$$p_i = \frac{h(i)}{\sum_{j=0}^{255} h(j)}, \text{ where } i = 0,1,\dots,255. \tag{4.1}$$

- *Step 2:* At the nth iteration, the updated mean values of the two Gaussian functions, u_1^{n+1} and u_2^{n+1}, are given by

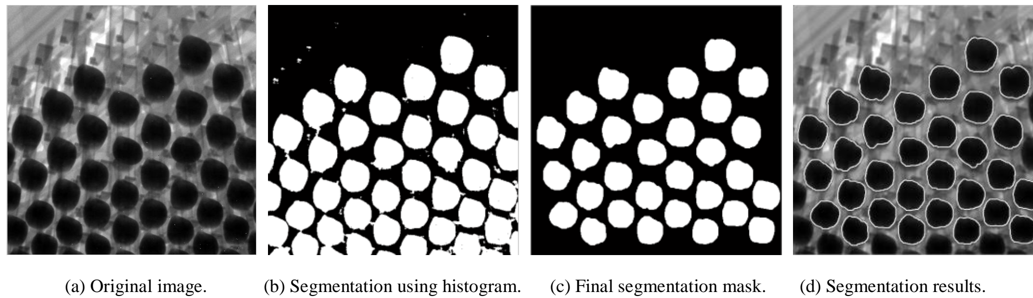
$$u_1^{n+1} = \frac{\sum_{k=0}^{T^n} k\, p_k}{\sum_{k=0}^{T^n} p_k}, \quad u_2^{n+1} = \frac{\sum_{k=T^n+1}^{255} k\, p_k}{\sum_{k=T^n+1}^{255} p_k} \tag{4.2}$$

where T^n denotes the threshold at the current n^{th} iteration. For $n = 0$, the initial threshold T^0 is selected to be the mean of the input image.

- *Step 3:* The new updated threshold is obtained from the mean values u_1^{n+1} and u_2^{n+1} as

$$T^{n+1} = \frac{u_1^{n+1} + u_2^{n+1}}{2} \tag{4.3}$$

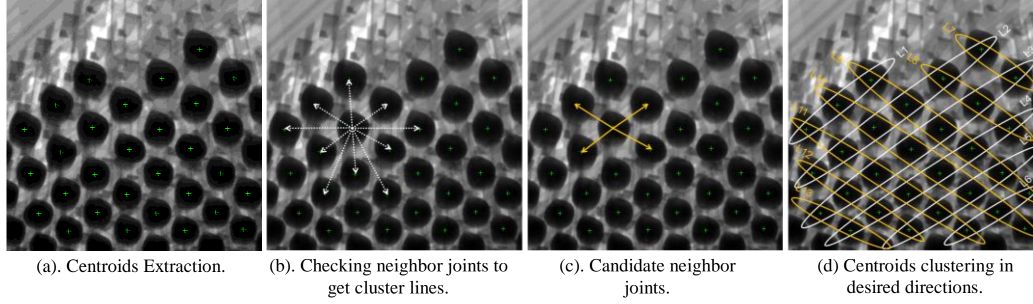
Steps 2 and 3 of the above process are repeated until the threshold value T^{n+1} is no longer changing. This procedure converges in 4 to 6 iterations on average. Fig. 4.8(b) shows an example of the segmentation mask that is obtained by applying the adaptive histogram-based thresholding algorithm to the image shown in Fig. 4.8(a) in order to segment the ROIs (solder joint areas). From Fig. 4.8(b), it can be seen that there are unnecessary regions that need to be removed such as small areas of background and incomplete joints that correspond to joints near the image borders. Another issue that can be observed from Fig. 4.8(b) is the fact that some ROIs (white areas in Fig. 4.8(b)) touch and overlap. This overlap may result in the erroneous merging of separate ROIs and,



(a) Original image. (b) Segmentation using histogram. (c) Final segmentation mask. (d) Segmentation results.

Fig. 4.8. ROI segmentation of candidate solder joints.

consequently, in the misclassification of the considered solder joints. Following the proposed adaptive histogram-based thresholding operation (Fig. 4.8(b)), a more accurate segmentation that can distinguish between separate solder joints is achieved by using the erosion and dilation mathematical morphology operations. The erosion operation is first performed followed by dilation using a circular structuring element [10, 164, 165]. Similarly, the undesired regions, including the incomplete joints near the image borders and small regions in the background can also be easily removed by using morphological operations. The small regions are removed by using the image opening operation (erosion followed by dilation) which is used to remove white areas with small sizes. The joints that are touching the image borders are removed if any pixel's location inside the segmented joint is lying on the border. Fig. 4.8(c) shows the final resulting segmentation mask which is used to get the desired segmented ROI area and which is obtained by applying the aforementioned mathematical morphology operations to the mask shown in Fig. 4.8(b). Fig. 4.8(d) shows the result of applying the mask of Fig. 4.8(c) to the original image in Fig. 4.8(a). It can be seen that the image is segmented into two regions, the solder joint areas (ROI) and the background area. In Fig. 4.8(d), the closed white contours are used to show the segmented joint areas. Each individual solder joint is then extracted by using an image labeling procedure [165]. The labeling procedure is applied to the binary segmentation mask image (Fig. 4.8(c)) which consists of 0's in the background region and 1's in the isolated white regions (corresponding to the segmented



(a). Centroids Extraction. (b). Checking neighbor joints to get cluster lines. (c). Candidate neighbor joints. (d) Centroids clustering in desired directions.

Fig. 4.9. Centroids and directional clustering: classification of centroids into different groups based on alignment along two directions.

solder joints). Each white region, consisting of 4 or more connected pixels, is uniquely labeled by assigning a label number to all pixels in the considered region. The label number is increased by 1 and the process is repeated when a new white region is detected. The result of the labeling process is an image that contains N label numbers, from 1 to N, where N represents the total number of segmented joints. Each individual joint can then be extracted and identified by its label number. For example, joint J_l with label l can be extracted from an image $f(x,y)$ which contains integer numbers representing the labels for the segmented joints as follows:

$$J_l = \{(x,y), \quad \text{where } f(x,y) = l\} \tag{4.4}$$

where x and y represent the pixels' locations inside image f and l represents the label number of each segmented region ($l = 1,2,\dots,N$).

4.2.3.2. Joint centroid computation and alignment

A novel aspect introduced as part of this work is the use of the centroids of the segmented joints as feature parameters that can be used for checking the degree of alignment between the solder joints despite the perspective effects and the non-parallel lines introduced by the multi-view imaging system at oblique angles. The centroid locations C_x^l and C_y^l for a joint J_l in the labeled image $f(x,y)$, which contains the label numbers of the segmented joints, are given by

$$C_x^l = \{Average(x), \quad \text{where } f(x,y) = l\}, \tag{4.5}$$

$$C_y^l = \{Average(y), \quad \text{where } f(x,y) = l\}.$$

Fig. 4.9(a) shows the extracted centroids ('+' symbol) of the segmented joints of Fig. 4.8(d). In order to cluster the centroids into directional clusters by grouping together centroids that lie on or close to a straight line, we make use of the fact that, in the considered setup, the images are captured at known oblique angles. In our case, the angles are 45°, 135°, 225°, 315° (Fig. 4.4), which are multiples of 45°, and which correspond to two main directional clusters at angles 45° and -45°. This fact is exploited in developing an efficient clustering scheme in which the search range for the alignment is limited to finding two directions θ_1 and θ_2 around 45° and -45°, respectively, where $45^o - \theta_\varepsilon \leq \theta_1 \leq 45^o + \theta_\varepsilon$ and $-45^o - \theta_\varepsilon \leq \theta_2 \leq -45^o + \theta_\varepsilon$ (in our implementation, $\theta_\varepsilon = 15^o$). The search for θ_1 and θ_2 is performed by computing the angles between the considered joint and its surrounding neighbors as shown in Fig. 4.9(b). This would result in determining the two neighboring joints (out of the surrounding neighbors) with angles in the acceptable range, the one whose angle is closest to 45° corresponding to θ_1 and the one whose angle is closest to -45° corresponding to θ_2. This process needs to be repeated at each joint in order to eliminate outlier neighboring joints. Fig. 4.9(c) shows the result of the search algorithm in locating the surrounding joints that lie in the desired directions. Following the same procedure locally at each joint, one can determine the groups (clusters) of joint centroids that lie on or near the same line as shown in Fig. 4.9(d). We refer to these clusters as directional clusters.

4.2.3.3. Centroid-based classification of good and suspect joints

A robust classification method is needed to classify each individual centroid in each directional cluster into suspect (non-wet) or good joints. This classification is performed by checking the degree of alignment of each joint's centroid with respect to joints in the

same directional cluster. The proposed reference-free classification procedure can be described as follows. Assume that an image has M directional clusters of joints and that each cluster i, $i = 1,..,M$, contains N_i number of joints J_{ik}, $k = 1,2,...,N_i$. Let C_x^{ik} and C_y^{ik} be the x and y coordinates of the centroid of joint J_{ik}. As illustrated in Fig. 4.10(a), for each joint J_{id}, $d = 1,...,N_i$, the degree of alignment is obtained by first calculating the angles φ_{idk} of that joint relative to the other joints J_{ik} , $(k = 1,2 ... N_i$ and $k \neq d)$ in the same directional cluster i as follows:

$$\varphi_{idk} = \tan^{-1}\frac{\left(C_y^{ik} - C_y^{id}\right)}{\left(C_x^{ik} - C_x^{id}\right)}, \quad k = 1,2,...,N_i, \qquad k \neq d \tag{4.6}$$

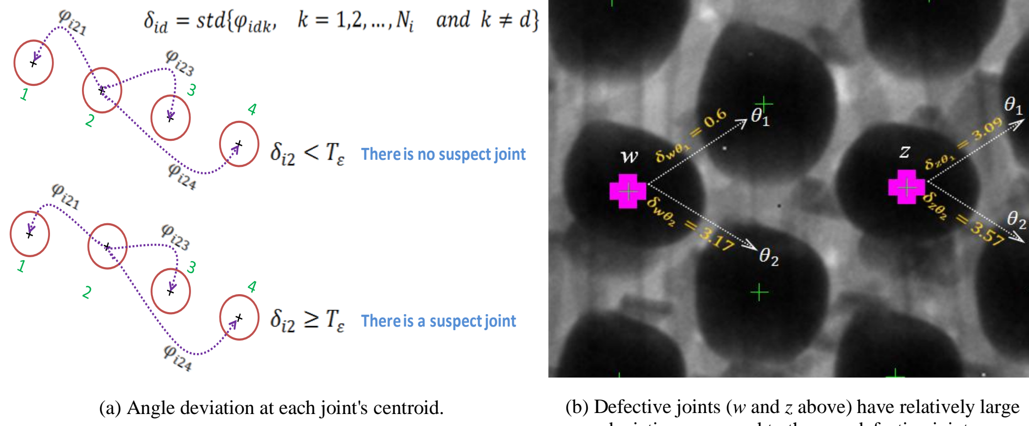
The angle deviation δ_{id} at joint d is then computed in directional cluster i using the standard deviation of the angles φ_{idk} as follows:

$$\delta_{id} = \sqrt{\frac{1}{N_i - 1}\sum_{k=1,\ k\neq d}^{N_i}\left(\varphi_{idk} - \frac{1}{N_i - 1}\sum_{s=1,\ s\neq d}^{N_i}\varphi_{ids}\right)^2} \tag{4.7}$$

A deviation threshold T_ε is used to classify each joint d in line cluster i into good or suspect joint. The classifier output can be expressed as:

$$Classifier\ output = \begin{cases}\text{Good joint}, & if \quad \delta_{id} < T_\varepsilon \\ \text{Suspect joint}, & if \quad \delta_{id} \geq T_\varepsilon\end{cases} \tag{4.8}$$

From (4.8), it can be seen that the classifier output depends on the deviation threshold value T_ε. If T_ε is smaller than the optimum value, the number of suspect joints would increase and consequently the false positive rate would increase, which in turn would increase the inspection time. If the deviation threshold T_ε is higher than the optimum value, the number of false positive joints would decrease; however, in this latter case, many non-wet joints would not be detected. A conservative deviation threshold which produces a low false positive rate and which results in a relatively high detection rate of non-wet joints was determined experimentally and based upon statistical analysis. In the



(a) Angle deviation at each joint's centroid.

(b) Defective joints (w and z above) have relatively large deviation compared to the non-defective joints.

Fig. 4.10. Detecting suspect joints using the proposed reference-free classification method.

experiment, a set of images which include defective (non-wet) and non-defective joints were used to calculate the angle deviation value for each joint in the given set of images as described in (4.6) and (4.7). The distribution of the angle deviation values obtained for defective and non-defective joints is close to a mixture of two Gaussian distributions. By analyzing the statistical distributions of the angle deviation values of defective (non-wet) versus non-defective joints, a threshold value that minimizes the false negative rate (missed detections) while constraining the false positive rate to be less than 1.5%, was determined to be 2. The process of calculating the optimum deviation threshold that minimizes the probability of classification error can also be obtained using a maximum-likelihood based optimization procedure. Let P_d be a Gaussian probability distribution with mean u_d and standard deviation σ_d, corresponding to the probability distribution of the defective joints. Similarly, let P_n be a Gaussian probability distribution with mean u_n and standard deviation σ_n, corresponding to the probability distribution of the non-defective joints. The ML-based optimum deviation threshold is given by [10]:

$$T_\varepsilon = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}, \tag{4.9}$$

where $a = (\sigma_d^2 - \sigma_n^2), b = 2(u_d\sigma_n^2 - u_n\sigma_d^2)$ and $c = u_n^2\sigma_d^2 - u_d^2\sigma_n^2 + 2\sigma_d^2\sigma_n^2 \ ln\,(\sigma_n P_d/\sigma_d P_n)$

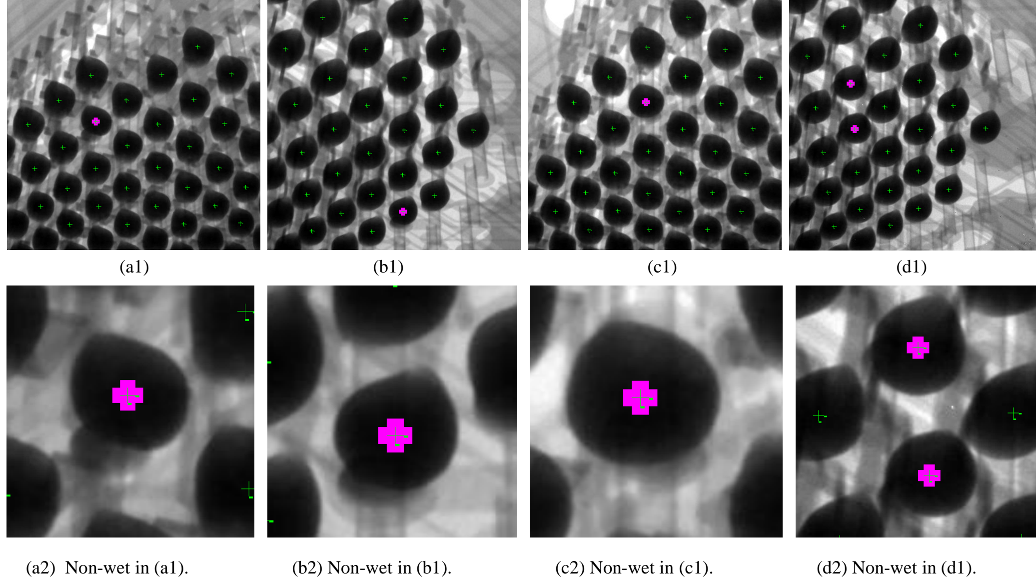


Fig. 4.11. Examples of non-wet solder joints detection using the proposed algorithm.

Since each joint belongs to two directional clusters, the degree of alignment for each joint is checked twice as described above, once for each cluster. This allows each joint to be checked twice for defect, which helps in increasing the detection rate of non-wet joints because, in some cases, the non-wet joints cannot be detected in one direction as they may be aligned in that direction with other joints, but can be detected due to misalignment in the second direction. This is illustrated in Fig. 4.10(b) which shows the output of the classifier for a sample image when the deviation threshold T_ε equals 2. In this case, the classifier detects two suspect joints, w and z in Fig. 4.10(b), which correspond to non-wet joints with maximum deviation angles 3.17° and 3.57°, respectively. From Fig. 4.10(b), it can be seen that non-wet joint w is aligned with the good joints in the θ_1 direction ($\varphi_{w\theta_1} = 0.6$), but is misaligned relative to the good joints in the θ_2 direction ($\varphi_{w\theta_2} = 3.17$). In other cases, a non-wet joint can be detected as a suspect joint in both the θ_1 and θ_2 directions as it can be seen for joint z in Fig. 4.10(b). Figs. 4.11(a1) to (d1) show examples of detecting non-wet joints in different boards using the proposed scheme. The detected non-wet joints are marked with a spot in their centers. Figs. 4.11(a2) to (d2) show, respectively, an enlarged view of the non-wet joints detected

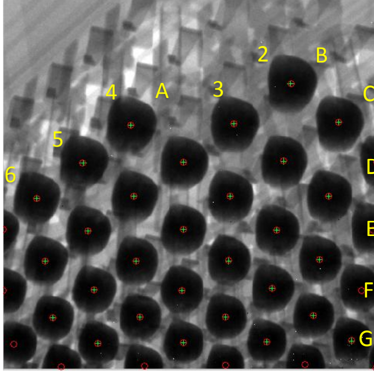


Fig. 4.12. Extracting joints' centroids in reference image.

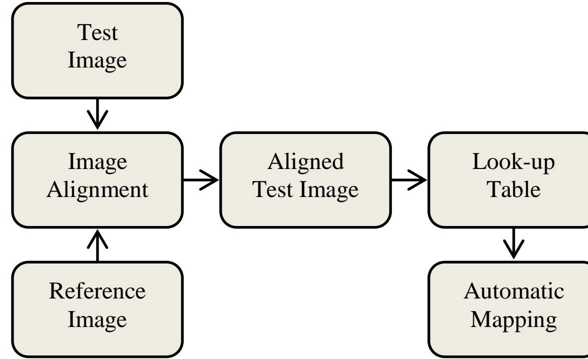


Fig. 4.13. Automatic mapping scheme.

in Figs. 4.11(a1) to (d1). Despite the fact that some of the non-wet joints are hard to notice by the human eye, such as for example the non-wet joints shown in Figs. 4.11(c1) and (d1), the proposed algorithm is capable of detecting these non-wet joints accurately.

4.2.4. Automatic joint mapping

Due to the fact that a joint can appear at least in 4 different images, each one corresponding to a different view that is taken at a different angle, the same non-wet joint can be detected several times. It follows that it is difficult to know whether a detected non-wet joint has been detected before or whether the detected joint corresponds to a new detection. To know if the joint has been detected before or not, the operator has to have a map which provides a unique label to the same joint in each of the views. The operator has to look at 150 images with approximately 30 joints each and locate each defective joint manually using a feature of the 2D x-ray that allows a map display through a series of several zooming steps. Getting the corresponding labels manually for the defective joints out of 4500 repeated joints in each socket is time consuming and tedious work for the operator and, in addition, it can produce inaccurate results if the joint is not pin pointed accurately. It is therefore crucial to have an automatic joint mapping algorithm which would not only save time but can also produce accurate results.

The main concern in automatic mapping is to get some data such as images of a reference socket that was already mapped to the master ball grid array of the considered

unit, and a lookup table. The images are taken in a specific sequence. Two types of sockets are considered in the examples shown in this method: U86 and U87. Each socket has 1366 unique joints and each joint has a unique label which consists of letters that represent rows, and numbers that represent columns in the master BGA file of each processor socket, i.e., AB5, B5, Y17, AW43....etc. Fig. 4.12 shows an example of joint mapping. There are some challenges to get an accurate automatic mapping directly from the images of the reference socket (referred to as reference images) such as lighting variation, incomplete joints in reference images, and misalignment between the test and the reference images. Fig. 4.13 presents the block diagram of the proposed automatic joint mapping. The look-up table is used to save the information obtained from the reference images such as image sequence number, joint label and XY coordinates of each joint's centroid. The process of generating the look-up table consists of first obtaining the centroid of each joint in each reference image as shown in Fig. 4.12, then saving the reference image sequence number, joint's label, and the XY coordinates of a joint's centroid in a table. An output from a computer-aided-design software socket design file containing the label and specific coordinates for each joint can be used and combined with the sequence of reference images in order to map the joints’ labels to their XY coordinates in the reference images. The generated lookup table can be used later as a reference to get the automatic mapping of each joint of the same socket in different boards. Using the saved look-up table, the automatic mapping is performed by comparing the XY coordinates of the joints' centroids in the test image to the XY coordinates of the same image sequence number, which are saved in the look-up table. The test image has to be aligned with respect to the reference image first before checking the look-up table. More details about the image alignment and automatic mapping will be discussed in the following subsections.

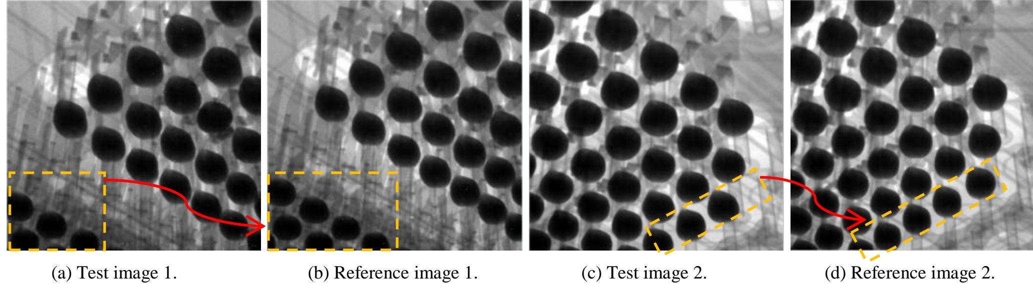
4.2.4.1. Automatic image alignment and mapping

Each socket has around 150 images obtained by using the x-ray machine. The x-ray machine is programmed to start scanning from the same position of the socket each time. This means that the image sequence numbers from 1 to 150 correspond to the same reference image sequence numbers for the same socket type. However, this correspondence is not exact as typically there are shifts between the test and reference images due to placement inaccuracies of the board relative to the x-ray stage. Unfortunately, these shifts produce misalignments between the test and reference images as shown in Fig. 4.14. To get an accurate joint mapping, the locations of the corresponding joints' centroids in the reference and test images should be aligned, otherwise the automatic mapping would produce inaccurate results. In general, there are different types of distortions between the reference and test images which can be summarized as translation, rotation, affine, and projective distortion [166, 167].

In the considered set-up, the distortions between the test and reference images during the image capture are translational only. Therefore, an alignment procedure based on the cross-correlation coefficient was found to work well for the considered application. The cross-correlation coefficient based alignment procedure consists of first shifting the test image by Δx and Δy relative to the reference image and then computing the cross-correlation coefficient $\rho(\Delta x, \Delta y)$ for the considered shift. The shift $(\Delta x, \Delta y)$ that results in the maximum cross-correlation coefficient is the one utilized to align the images. The cross-correlation coefficient $\rho(\Delta x, \Delta y)$ between test image T and reference image R is given by

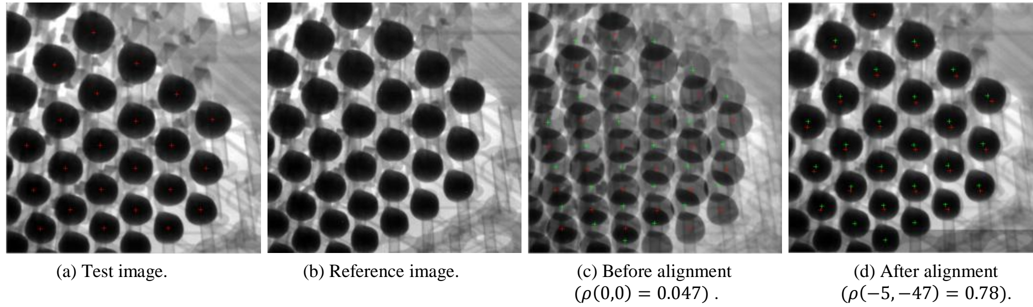
$$\rho(\Delta x, \Delta y) = \frac{\sum_x \sum_y \left(R(x,y) - \overline{R(x,y)}\right)\left(T(x+\Delta x, y+\Delta y) - \overline{T(x+\Delta x, y+\Delta y)}\right)}{\sqrt{\sum_x \sum_y \left(R(x,y) - \overline{R(x,y)}\right)^2 \sum_x \sum_y \left(T(x+\Delta x, y+\Delta y) - \overline{T(x+\Delta x, y+\Delta y)}\right)^2}} \tag{4.10}$$

where $\overline{T(x + \Delta x, y + \Delta y)}$ and $\overline{R(x, y)}$ are the mean values of images $T(x + \Delta x, y + \Delta y)$ and $R(x, y)$, respectively. In (4.10), x and y represent the index of each pixel inside



(a) Test image 1. (b) Reference image 1. (c) Test image 2. (d) Reference image 2.

Fig. 4.14. Misalignment between test and reference images.



(a) Test image. (b) Reference image. (c) Before alignment ($\rho(0,0) = 0.047$) . (d) After alignment ($\rho(-5,-47) = 0.78$).

Fig. 4.15. Image alignment between test and reference images.

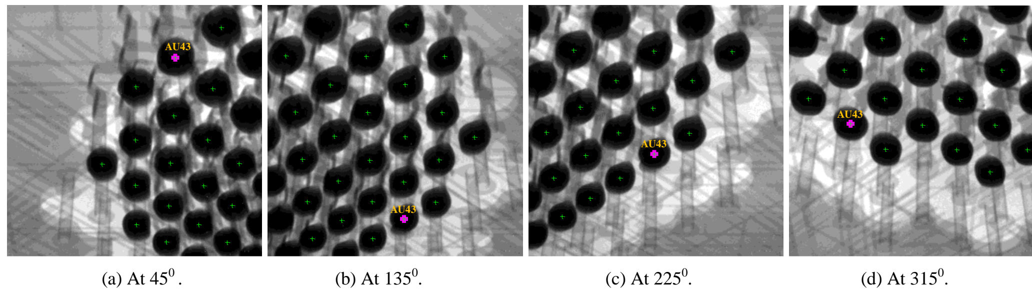
the image, Δx and Δy represent the shift parameters in the x and y directions, respectively. The output cross-correlation coefficient $\rho(\Delta x, \Delta y)$ is a function of the shift parameters Δx and Δy. The final computed shifts between the test and reference images are the values of Δx and Δy that correspond to the highest cross-correlation coefficient in (4.10). Figs. 4.15(a) and (b) show test and reference images, respectively. Fig. 4.15(c) shows the test and reference images superimposed above each other in order to illustrate the misalignment between them. This misalignment results in a poor cross-correlation factor ($\rho(0,0) = 0.047$). Fig. 4.15(d) shows the two superimposed images after correcting for the misalignment by using the proposed alignment algorithm. For this example, the alignment shifts, which maximize the cross correlation coefficient, were found to be -5 and -47 in the directions of x and y, respectively. After aligning the test image relative to the reference image, the automatic mapping is achieved by comparing the XY coordinates of the joints' centroids in the aligned test image to those saved in the look-up table for the same image sequence number. This can be done by using an appropriate distance function to get the label of the closest reference joint.

Table 4.1: Non-wet detection using the proposed algorithm with automatic mapping.

Detected non-wet joint	Number of Detections per joint
AY36	1
M2	2
R2	2
AU43	4
B5	6

Table 4.1 shows the results of running the proposed automatic mapping algorithm on a series of images at different oblique angles from a single socket. Since several images of the same socket are taken from different angles, the same joint can appear in several images (views). In this case, the number of non-wet joints which were detected by the algorithm, and confirmed by the operator, is 15 *non-unique* joints; however, the actual non-wets are 5 unique joints as given in Table 4.1, which shows that the same non-wet joint can be detected more than once (due to the fact that it appears in several views). Therefore, without the proposed automated non-wet detection and mapping system, an operator would have to examine manually the same defective joint in several views. The proposed automatic defect detection algorithm can locate the non-wet (defective) joints in all the views and determine which ones correspond to the same joint. Thus, with an operator is in-loop, once the operator confirms that a joint is defective, that joint is not shown again to the operator in subsequent images, saving thus significant time. Without the proposed system, the operator would have to detect and confirm 15 non-wet joints, although 10 of these are redundant and only 5 non-wet joints are unique. At least 2/3 of the operator's time can be saved since the proposed system is able to present to the operator only the 5 non-wet unique joints.

Fig. 4.16 shows different images in which the AU43 non-wet joint was detected more than once. The AU43 non-wet joint can be clearly noticed in the images shown in Figs. 4.16(a), (b) and (c), while it cannot be seen clearly in the image of Fig. 4.16(d). However, the algorithm was successful in detecting it as a suspect joint in all 4 images because the AU43 joint's centroid exhibits a significant deviation relative to the centroids of the good joints that belong to the same cluster.



(a) At 45^0. (b) At 135^0. (c) At 225^0. (d) At 315^0.

Fig. 4.16. Results of the proposed algorithm with automatic mapping showing different-view images in which the same non-wet joint (AU43) was detected more than once. The detected non-wet joint is marked with a spot in its center.

4.2.5. Performance results and statistics with ground truth data

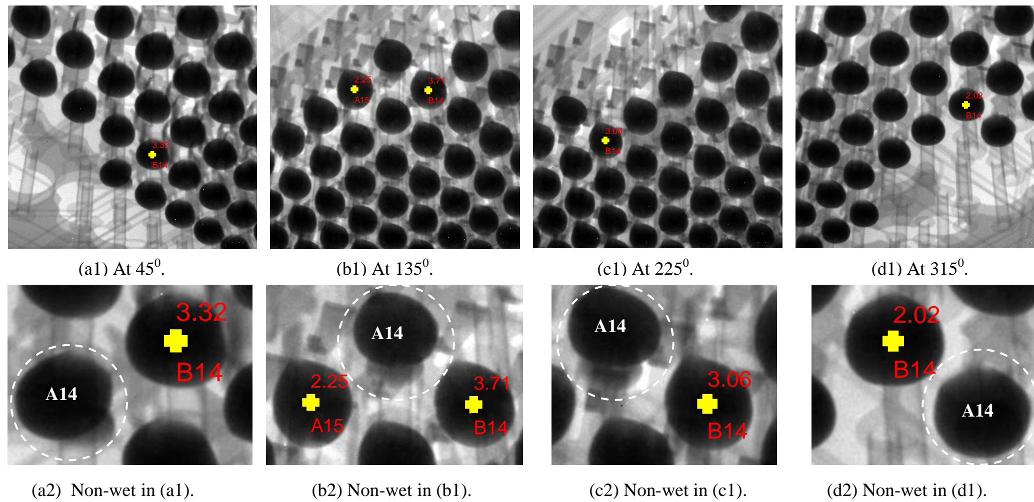
Ground truth data clearly identifying the non-wet joints is required to verify the accuracy of the proposed algorithm with automatic mapping. Dye and pull validation was used to get ground truth data. Dye and pull testing consists of immersing the board with the unit attached in a red dye. The dye will penetrate the socket and leave a residue to enable an inspection of the goodness of the joint. The unit is then physically pulled off of the board. The non-wet joints show up as an abnormal separation point between the board and the unit and are noted by the solder residue remaining on the unit. Each unit is inspected by an operator to validate the separation point to physically confirm the presence of a non-wet joint. Dye and pull validation was run for 56 sockets. Each socket has 1366 unique joints, which gives a total number of joints equals to 76,496 joints in the 56 sockets. The number of non-wet joints that were detected by the dye and pull was 47 as given in Table 4.2. Table 4.2 gives the results of the 47 non-wet solder joints in 56 sockets after investigating each joint in each processor socket using the dye and pull ground truth method and the proposed method. Each non-wet solder joint in Table 4.2 was represented by its board number, processor socket number, and unique joint's label. The total number of non-wet joints that were detected by the proposed algorithm and that matched the dye and pull results are 45 out of 47 non-wet joints. This gives a detection rate equal to 95.8% (45/47). The algorithm detected a total of 911 suspect joints of which 866 were false positives. Given the total number of unique joints in 56 sockets, which is

76496 joints, the resulting false positive rate is 1.1% (866/76496). The average number of suspect joints per socket is 16 joints (911/56). Assuming 1 suspect joint per image, this requires an average of 16 images to be inspected instead of 150 images in a single socket, which reduces the operator's manual inspection time by approximately 89.6%. For the 2 missed non-wet joints (A14 and K1), the algorithm detects neighboring joints as given in Table 3.2 and as shown in Fig. 4.17, which gives a detection rate of 100% with an operator in the loop. The reason that the detection of neighboring joints can give a detection rate of 100% with the operator in the loop is that a user interface that specifically points out the suspect joints to the operator enables the focus of the attention to a specific area. That area will always contain the non-wet if one is present.

Table 4.2: Non-wet joints obtained from the dye and pull test and the proposed algorithm.

Board No.	Socket No.	Dye and pull Results	Algorithm Results	Comment
1	U86	E1	E1	Match
1	U86	A40	A40	Match
1	U86	G42	G42	Match
2	U87	C3	C3	Match
3	U86	AV41	AV41	Match
3	U86	AW37	AW37	Match
3	U86	B33	B33	Match
3	U87	C5	C5	Match
3	U87	D4	D4	Match
4	U87	C2	C2	Match
5	U87	AW14	AW14	Match
6	U87	AW38	AW38	Match
6	U86	C5	C5	Match
7	U86	A14	A15 & B14	Miss
7	U86	B5	B5	Match
7	U87	AT41	AT41	Match
7	U87	C5	C5	Match
8	U86	BA13	BA13	Match
8	U87	D3	D3	Match
8	U87	E2	E2	Match
9	U86	AY26	AY26	Match
9	U87	AW23	AW23	Match
10	U86	C3	C3	Match
11	U86	C39	C39	Match
11	U87	G2	G2	Match
12	U87	AY36	AY36	Match
13	U87	K1	K1	Match
14	U86	AY41	AY41	Match
14	U86	N2	N2	Match
14	U86	P2	P2	Match
14	U87	D2	D2	Match
15	U87	AU40	AU40	Match
15	U87	AL42	AL42	Match
15	U87	AW39	AW39	Match
15	U87	C5	C5	Match
16	U87	AM42	AM42	Match
16	U87	AL42	AL42	Match
16	U87	AY17	AY17	Match
17	U87	AW22	AW22	Match
18	U86	C30	C30	Match
19	U86	AY14	AY14	Match
20	U86	B37	B37	Match
20	U87	B5	B5	Match
21	U87	F1	F1	Match
21	U87	K1	L1	Miss
22	U86	AY13	AY13	Match
22	U87	L2	L2	Match

The reasons why these two joints were miss-detected are because the deviation threshold was determined using (4.9) as described in Section 4.2.3.3 in order to minimize



(a1) At 45⁰. (b1) At 135⁰. (c1) At 225⁰. (d1) At 315⁰.

(a2) Non-wet in (a1). (b2) Non-wet in (b1). (c2) Non-wet in (c1). (d2) Non-wet in (d1).

Fig. 4.17. Examples showing the detection of the neighbor joints to the missed non-wet joints (A14) in different images using the proposed algorithm.

the classification error in a statistical sense (i.e., minimize the probability that a joint is being classified as non-defective given that it is defective). Even when the threshold is determined experimentally, this is done through a statistical analysis of the data. From statistics and probability theory, there will be some joints which would appear to belong to the non-defective class even though they are indeed defective based on the distributions of the angular standard deviations for the defective and non-defective class, which cross each other. Currently, the accuracy of the system is significantly high (95.8%). In addition, as indicated previously, the system can detect the neighboring joints of a missed defective joint raising the accuracy to 100% with an operator in the loop.

A comparison between the results of the proposed algorithm with one of the up-to-date advanced capability x-ray machines with an embedded non-wet algorithm was run for the same sample sockets. The data shows a non-wet detection rate in the range of 43% to 75% for the advanced capability x-ray machines. The proposed algorithm gives a 95.8% detection rate of the non-wet joints, which is 21% to 53% more accurate than the results of the advanced capability x-ray machines. If the above statistics are calculated for non-unique joints, the actual number of the joints that are investigated by the algorithm for suspects are approximately 252,000 non-unique joints due to the repetitive imaging of

the same joints at different oblique angles. The total number of detected suspects is 1281 non-unique joints and the total number of actual non-wets is 159 non-unique joints, which gives a false positive rate of 0.45%.

The algorithms have been implemented using the Intel® Integrated Performance Primitives and Intel® Math Kernel Library. The code was then embedded in a C++ program. The code is multi-threaded and performance optimized using Intel® Threading Building Blocks and Intel® Parallel Studio. The overall performance of the proposed method is fast and the inspection process is very efficient due to the fact that the instant that the 2D x-ray image is available, it is processed and, if suspect joints are found, the image is immediately made available to the operator. This enables the inspection to be completed at the same time or very close to the time that the x-ray itself has finished imaging the socket.

4.2.6. Summary of non-wet detection

An automatic defect detection algorithm was presented in the previous section to accurately locate the non-wet joints in processor sockets. The algorithm is capable of locating the non-wet joints in socket images that were taken at different oblique angles. The conducted performance evaluation and resulting statistics show that the proposed algorithm provides a detection rate of 95.8%. This is 21% to 53% better than commonly used state-of-the-art inspection machines. The proposed algorithm has the potential of resulting in a detection rate of 100% with an operator in the loop due to the identification of neighboring joints to the missed defective ones. Moreover, the proposed scheme reduces the false positive rate, false negative rate, and the inspection time, and requires only a relatively inexpensive 2D x-ray imaging device, which makes it cost effective and manufacturing friendly. Moreover, the algorithm has been implemented into an application which is currently being used in technology development at Intel Corporation. Future directions in our research include investigating the applicability of the proposed

algorithm to multiple processor sockets and ball grid arrays. In addition, an improved automatic mapping method, without using images of a reference socket, is being developed. Other future work include considering new features beside the angular deviations of the joints' centroids, and characterizing other types of defects in solder joints.

4.3. Robust Automated Void Detection in Solder Balls and Joints

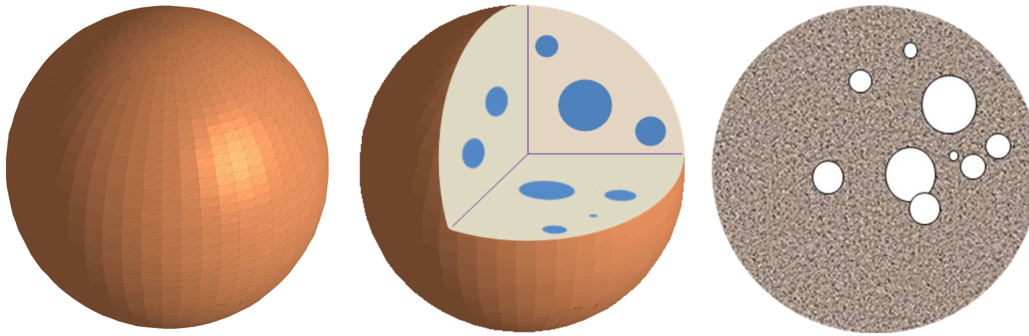
Accuracy in solder balls and joint void detection is very important. If voids are incorrectly identified, board yield will be affected by incorrect scrapping and rework. Voids are difficult to detect using manual inspection alone. One current solution to make voids visible involves the use of a 2D x-ray system to image the boards. Some existing x-ray inspection systems have void detection algorithms that require the use of intensive, time consuming, fine tuning operations. These algorithms typically use two different global thresholds to segment the balls or joints and the voids using operator trial and error. However, using global thresholding over the entire image is invalidated due to varying image brightness. Existing methods also eliminate balls or joints that are partially occluded by other components due to difficulties in segmentation. The results are that many voids that can be easily observed by the human eye are missed by the existing automated methods.

In this section, a robust, accurate, and automatic void detection algorithm is proposed. The method is applicable to either pre-SMT (Surface Mount Technology) solder balls or post-SMT solder joints. For simplicity, the term balls will be used throughout the document. The proposed method is able to detect voids with different sizes inside the solder balls, including the ones that are occluded by board components and under different brightness conditions. The proposed method consists of segmenting individual balls, extracting occluded balls, and segmenting voids inside the solder balls. The segmentation of the individual balls is achieved by using the proposed histogram and

morphological based segmentation method. A voting procedure is used to segment the occluded balls where the pixels inside the occluded area are checked to obtain candidate pixels representing the occluded joint's or ball's centroids. An independent edge detection procedure is used to get candidate voids inside individual balls. Mathematical morphology operations are used to locate all possible valid voids and remove non-void areas. The proposed algorithm was applied to 3 different Intel products. The results of the proposed method were compared to the results obtained by an automated algorithm in an existing state-of-the-art 2D x-ray inspection system, the results obtained by trained operators from 2D x-ray images, and the results obtained by trained operators from 3D CT scan images. The results (pre-SMT solder balls) show that the proposed method is capable of successfully locating all possible visible voids inside the solder ball even the ones that were missed by using other methods as well as those that are hard to see by the human eye. The results also show a high correlation with ground-truth data obtained from 3D CT scan and experienced operators. The algorithm is fully automated, benefits the manufacturing process by reducing operator effort, and provides a cost effective solution to improve output quality. The results were presented and published in [52, 53].

4.3.1. Problem statement

Voids are one of the major defects in solder balls and are defined as cavities formed inside the solder joint due to the amount of out-gassing flux that gets entrapped in the solder joint during reflow. Some causes of voids are trapped flux that has not had enough time to be released from the solder paste, and contaminants on improperly cleaned circuit boards. Voids in solder balls are also caused by the reduction or metallic oxides by the soldering fluxes [168]. Voids appear as a lighter area inside the solder balls and joints on a 2D x-ray image and are typically found randomly throughout the package [169]. Previous studies show that the existence of voids decreases the solder joint's life [170]. In [171], the authors concluded that smaller voids grow much more slowly than the bigger

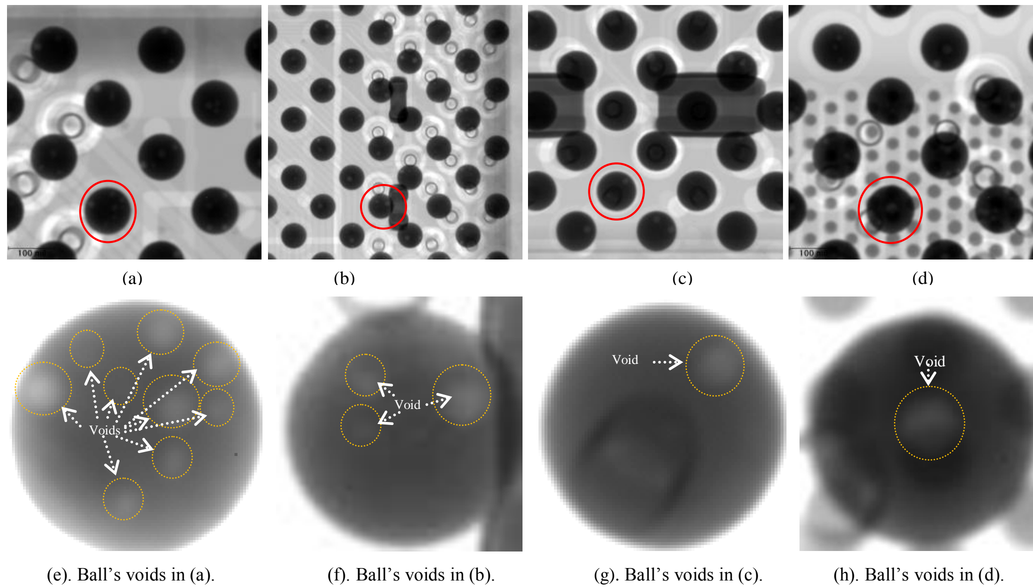


(a). 3D image of solder ball. (b). Cross-section in 3D image that shows voids inside solder ball. (c). Voids distribution in 2D image.

Fig. 4.18. Solder ball shape in 3D and the distribution of voids in both 3D and 2D images.

voids. The extensive use of solder balls and joints on printed circuit boards (PCB) necessitates reliable, void detection in the balls and joints to prevent infant mortality failures. The Institute for Printed Circuits (IPC) and the Joint Electron Device Engineering Council (JEDEC) have developed standards for inspecting assemblies of electronic products. IPCA610D specifies a 25% or less cumulative voiding percentage in post-SMT solder joints. JEDEC void inspection criteria are based on the size of the void and the cumulative percentage of voiding in an individual ball. The new JEDEC guideline for void inspection criteria was recently issued by the JC14-1 committee and specifies a 15% or less cumulative voiding percentage in a solder joint.

A solder ball has a spherical shape as shown in Fig. 4.18(a). Voids are distributed randomly inside the solder ball as shown in Fig. 4.18(b) which gives a cross-sectional view inside the spherical solder ball. The 2D top view of the spherical solder ball is shown in Fig. 4.18(c). Voids are hard to locate inside solder balls and joints using manual inspection tools. 2D x-ray machines are used to make the voids inside the solder balls and joints visible to the operator as shown in Fig. 4.19. The output of the x-ray machine is shown in Figs. 4.19(a)-(d) which represent the 2D images of different products. Figs. 4.19(e)-(h) show zoomed-in images for balls that are highlighted in Figs. 4.19(a)-(d), respectively. The images in Figs. 4.19(e)-(h) are enhanced for visual clarity to show all possible voids inside the highlighted solder balls in Figs. 4.19(a)-(d). One of the



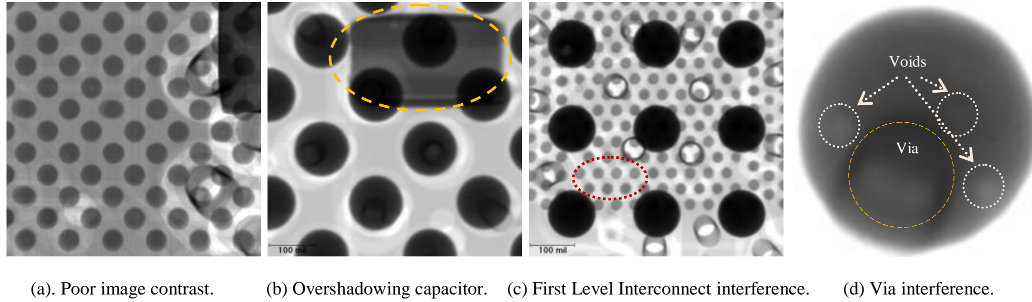
(a) (b) (c) (d)

(e). Ball's voids in (a). (f). Ball's voids in (b). (g). Ball's voids in (c). (h). Ball's voids in (d).

Fig. 4.19. Example of 2D x-ray images that show voids inside solder balls in different product lines.

methods used to locate voids inside solder balls is to use existing image enhancement software to examine the x-ray images manually after changing the image brightness and contrast. The operator must then detect and measure each observed void manually. A skilled operator takes around 4 minutes to locate and measure one void inside a solder ball which makes the manual process very tedious and time consuming. The process is also highly variable due to the difference in training, skill and other human characteristics of operators.

The necessity of getting a robust and reliable automated void detection algorithm is important. Before discussing the existing void detection methods, let us address the main challenges that are observed in the acquired 2D x-ray images. These challenges can be summarized as follows: (i).poor image contrast at some balls, which makes the voids difficult to detect by the human eye; (ii).interference of other components in the unit such as void-like artifacts, die bonds, vias (plated through holes), via reflection, and overshadowing capacitors; (iii).irregular shapes (non-circular) caused by the fact that there can be overlapped voids present in the 2D images that do not conform to the predominant circular void shape; (iv).missing or overlapped voids not visible in the 2D x-



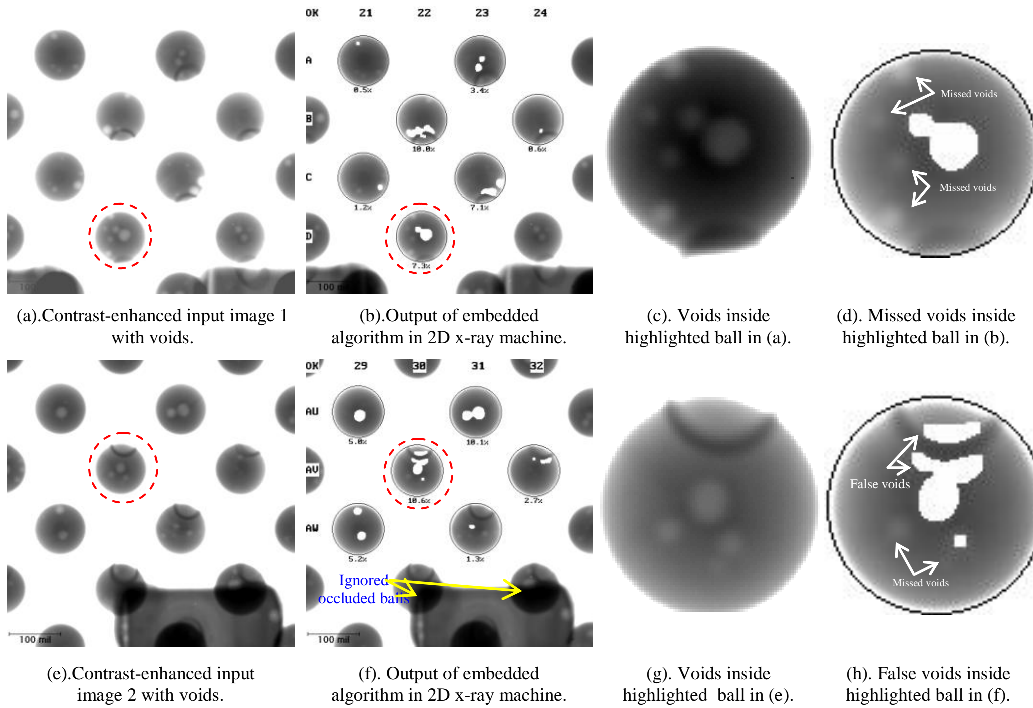
(a). Poor image contrast. (b) Overshadowing capacitor. (c) First Level Interconnect interference. (d) Via interference.

Fig. 4.20. Examples of some Challenges in the 2D x-ray images.

ray images compared to the details obtained from the 3D CT scan images. Fig. 4.20 shows some examples of the aforementioned challenges in the 2D x-ray images. Most of the existing void detection systems do not provide solutions to tackle these issues, which results in missed and false called voids and, thus, inaccurate detection.

Using a 2D x-ray machine is much faster and less expensive to measure each ball as compared to a 3D or multi-dimensional x-ray machine, but at a loss of some accuracy as mentioned above. Some existing 2D x-ray machines have embedded inspection systems that include void detection algorithms. However, the existing void detection algorithms that are used in 2D x-ray machines require intensive preprocessing steps and manual fine tuning. An example of the steps of image acquisition and void detection in a 2D x-ray machine with an embedded void detection algorithm, follows:

- The first step is to set up the 2D x-ray machine to make sure that there is a visible gray scale difference between the solder balls, background and voids. The operator manually uses different gray scale levels for solder ball, background and voids depending on the intensity of the void level. The setting of the gray scale level is achieved by changing the current and the voltage of the x-ray beam.
- The second step is to set up the void detection software. A typical software setup includes defining the expected solder ball size and additional inspection features. Some of these features can include segmentation threshold values such as the threshold used to identify the gray scale difference between a solder ball and background area, and the



(a).Contrast-enhanced input image 1 with voids.

(b).Output of embedded algorithm in 2D x-ray machine.

(c). Voids inside highlighted ball in (a).

(d). Missed voids inside highlighted ball in (b).

(e).Contrast-enhanced input image 2 with voids.

(f). Output of embedded algorithm in 2D x-ray machine.

(g). Voids inside highlighted ball in (e).

(h). False voids inside highlighted ball in (f).

Fig.4.21. Results using 2D x-ray machine with an embedded void detection algorithm.

threshold used to identify the difference between voids and solder ball. Threshold values are determined by the operator by trial and error. After using the selected threshold to segment the solder ball from the background, some feature parameters such as predefined size (area) of a solder ball are used to decide which one of the segmented regions should be processed and which one should be ignored. A global thresholding method cannot be optimized to provide sufficient adjustment to capture all voids with the lighting variations typically encountered in an x-ray image. In addition, this method is unable to segment the balls that are occluded by overshadowing components.

- The third step involves detecting the voids in the considered set of images using the software with the selected threshold values and inspection features.

An example of void detection using the above system is shown in Fig. 4.21. The results shown in Fig. 4.21 show that the 2D x-ray machine with an embedded void detection algorithm misses voids, detects many false voids, and fails to process balls that are

occluded by overshadowing capacitors. The system is also troubled by vias under the balls and classifies the vias as voids. In addition, this system requires a lot of manual operations and parameters tuning for each new product line.

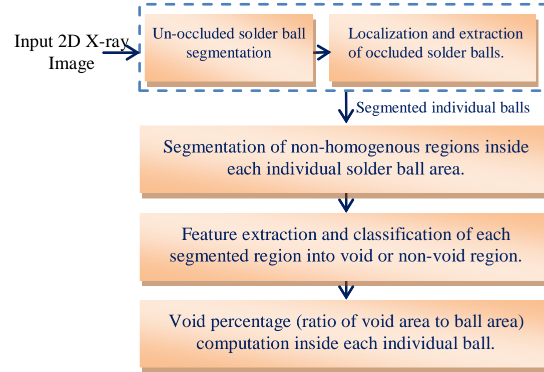


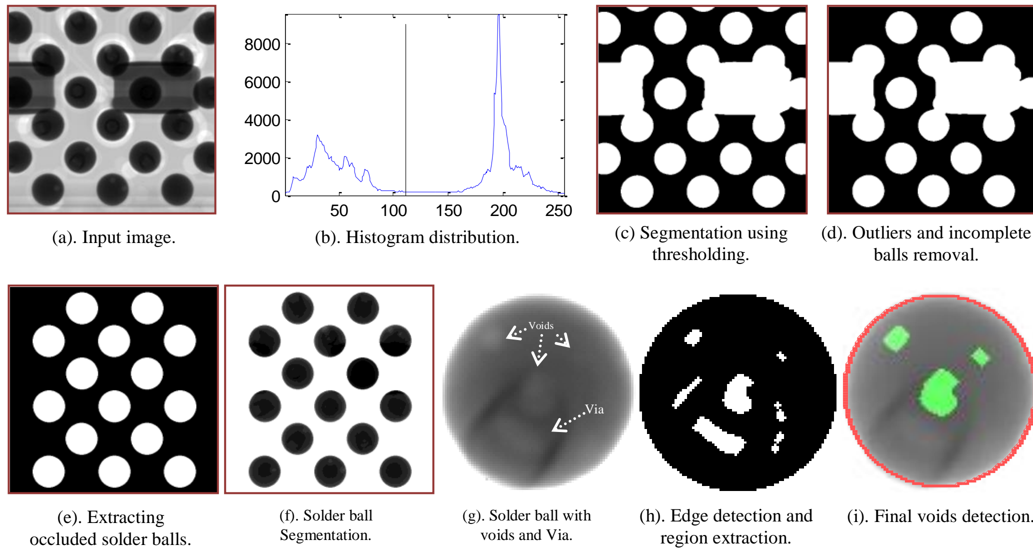
Fig 4.22. Block diagram of the proposed method for void detection inside solder balls.

The following subsections are used to describe the procedures of the proposed method for void detection in solder balls. Section 4.3.2 presents the steps of the proposed void detection algorithm. Performance results and comparison with existing methods are presented in Section 4.3.3. Section 4.3.4 gives an improved proposed method to detect voids in the presence of via and via reflection. Simulation results for the proposed improved void detection method are presented in Section 4.3.5. A summary is provided in Section 4.3.6.

4.3.2. Automated void detection method

The main goal of this work is to provide a reliable, highly accurate, and fully automated scheme for void detection in 2D x-ray images. The proposed algorithm is designed to be robust to the challenges that are present when dealing with 2D x-ray images as discussed in Section 4.3.1. In this section, we present more details about the proposed void detection algorithm which is capable of detecting voids with different sizes inside the solder balls and under different brightness conditions.

Fig. 4.22 shows the block diagram of the proposed automated void detection method. The block diagram summarizes the steps of the proposed method including solder ball segmentation, the extraction of solder balls that are occluded by overshadowing capacitors, the extraction of candidate regions inside each segmented solder ball, the selection of feature parameters, and the classification of the candidate regions inside the



(a). Input image. (b). Histogram distribution. (c) Segmentation using thresholding. (d). Outliers and incomplete balls removal.

(e). Extracting occluded solder balls. (f). Solder ball Segmentation. (g). Solder ball with voids and Via. (h). Edge detection and region extraction. (i). Final voids detection.

Fig. 4.23. Steps of the proposed void detection method.

solder ball in order to exclude all non-void regions. Fig. 4.23 illustrates the steps of the proposed method. More details about the individual solder ball segmentation and the candidate void detection and classification are given in the following subsections.

4.3.2.1. Individual solder ball segmentation

Due to the fact that the 2D x-ray images suffer from inconsistent lighting, in order to segment the solder ball area, it is important to use an adaptive threshold value which is based on the image intensity distribution instead of using a fixed threshold value. In the proposed method, an automatic thresholding value based on the image histogram analysis is applied to the original image to segment the solder balls. Fig. 4.23(b) shows the histogram of the 2D x-ray image shown in Fig. 4.23(a) where the histogram contains two main cluster regions represented by two distinct peaks: the solder ball and background regions. In order to segment the solder balls regions (ROIs) from the background region, the segmentation threshold is determined automatically based on the distribution (histogram) of the considered image. From the distribution of the input image, the mean of the image is computed and is used as the initial segmentation threshold value. The distribution range is then divided into two parts, also referred to as clusters,

corresponding to pixel values below and above the current threshold, respectively. The threshold value is efficiently refined by computing the cluster means directly from the input image distribution without the need to segment the image. The updated threshold estimate is set to be the average of the two cluster means. This refinement step is repeated until the two cluster mean values no longer change, resulting in the final threshold estimate. This procedure converges in an average of 4 to 6 iterations. Fig. 4.23(c) shows the results after applying the above adaptive segmentation threshold computation procedure. Incomplete balls that are touching the image border are excluded from further void location processing. The removal of those incomplete balls is achieved by comparing each incomplete ball's area (A_i) with respect to the area of the individual complete ball (A_b), where A_b is taken as the median area of the segmented ROI areas inside the input image. Fig. 4.23(d) shows the results of excluding incomplete balls that are touching the image border and have an incomplete ball area less than 0.7 of the individual complete ball area (A_b).

Existing methods such as embedded algorithms in x-ray machines are unable to process balls that are occluded by overshadowing components, even though some of these balls may have high void percentages. In the proposed method, a voting procedure is devised to extract the occluded balls by locating their centroids. This can be achieved by exploiting the fact that the solder balls are aligned along different directions, including $0°, \pm 45^{\circ}, \pm 90^{\circ}$ as shown in the input image in Fig. 4.23(a). The extraction of the occluded balls' centroids using the proposed voting procedure is done as follows: (i). locate the occluded regions as the segmented regions whose area is greater than 1.2 of the individual ball's area A_b; (ii). calculate the centroids of the individual complete balls outside the occluded regions; (iii). calculate the directional angles between each pixel inside the occluded region and the centroids of the individual balls outside the occluded

region; (iv). keep only those pixels that match at least two angles in the directions of $0°, \pm45^{\circ}, \pm90^{\circ}$ inside the occluded region; the results of this step produce isolated regions consisting of clusters of connected pixels; (v). calculate the centroid of each region using an image labeling procedure [165]; (vi). extract each occluded ball by drawing a circle centered at that extracted centroid with a radius $R = \sqrt{A_b/\pi}$, where A_b is the previously determined complete solder ball area; (vii). remove any false centroids by checking the distance between neighboring balls' centroids (remove if distance is less than 1.2 of the individual ball's diameter). The result of the proposed scheme after locating the occluded balls is shown in Fig. 4.23(e) which shows the final segmentation mask. Using the final segmentation mask, one can easily extract each individual complete ball from the original input image including the balls occluded by overshadowing capacitors as shown in Fig. 4.23(f).

4.3.2.2. Candidate void detection

Locating voids inside each segmented solder ball requires a robust classification procedure to detect actual voids and remove non-void regions. There are many challenges when it comes to automatically detecting actual voids and excluding non-void regions inside solder balls such as: lighting variation, die balls and via interference. It is thus important to process each individual ball independently to locate voids. This is needed in order to tackle the lighting variation and die ball interference issues. In the proposed scheme, each ball is extracted and treated independently by using an image labeling procedure [165]. To locate the contours of all possible regions inside the segmented solder ball for further processing, an edge detection procedure [7-12] is applied to the segmented solder ball. In our implementation, a simple Laplacian of Gaussian (LoG) edge detection method was found to produce accurate results [11, 12, 172]. The LoG edge detection is a combination of two filters' kernels: a Gaussian filter and a Laplacian

filter. The Gaussian filter is used to reduce the high frequency components and the noise effect in order to obtain a regularized image before using the Laplacian filter. The Laplacian filter is applied to the regularized image in order to locate the edges by using the second derivative of the regularized image. Since the convolution operation is associative, one can convolve the Gaussian filter kernel with the Laplacian filter kernel first, and then convolve this combined filter with the input image to speed up the calculations. The edge detection can however result in some open contours. Closing open contours is a necessary process to extract each segmented region. The procedure for closing open contours is performed by first labeling each open contour, and then checking whether there are neighboring open contours, within few pixels, that can be connected together to form an enclosed region. This step is repeated to close each open contour inside each segmented solder ball. Each region inside a closed contour is then extracted by filling the region inside each closed contour by inverting the image to have 1's inside the closed contour and 0's on the contour boundaries, followed by a labeling procedure [165]. Fig. 4.23(h) shows the results of edge detection and candidate region extraction for the ball shown in Fig. 4.23(g).

4.3.2.3. Feature extraction

Each extracted candidate void region inside the solder ball should be classified as a void or non-void region. However, some artifacts (such as vias and die bonds) have void-like properties. Therefore, the devised classification method should be robust to such artifacts in order to detect actual voids and exclude suspect voids. For this purpose, the proposed automated classification method makes use of feature parameters that well describe the actual voids and are robust to artifacts such as vias and other interferences in the segmented ball. The employed features exploit the following properties of the void region: (i). voids are brighter compared to surrounding area (some vias are brighter as shown in Fig. 4.23(g)); (ii). voids have shapes close to a circular shape (can have

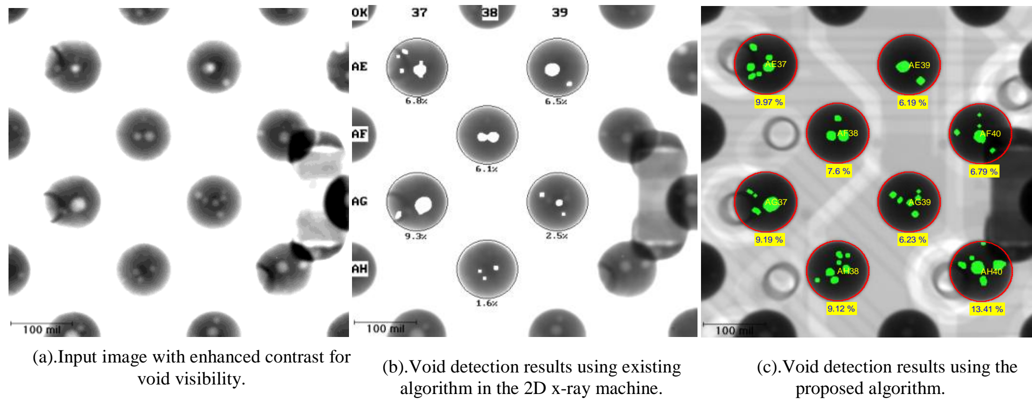
irregular shape if the 2D image has overlapped voids). Consequently, the proposed feature parameters that are used for classification consist of an adaptive and automated classification threshold which is adaptively determined based on the brightness of each region and its surrounding contour, and which can be used to remove non-void regions, and of a compactness factor [10] which can be used to exclude regions with irregular shapes. The compactness factor of a considered segmented region is computed as $CF_l = p_l^2 / 4\,\pi\, A_l$, where p_l and A_l represent, respectively, the perimeter and the area of the considered region. If the region has a shape close to a circular shape, its compactness factor is close to "1", otherwise the region has a non-circular shape.

4.3.2.4. Void classification

The proposed classification consists of 3 main stages: (i). the first stage is used to keep all candidate regions with void-like characteristics by using the aforementioned adaptive classification threshold and a large compactness factor filter; (ii). the second stage is used to refine the results of the first stage and eliminate more non-void regions (false calls) by using adaptive dilation to obtain a regulated shape followed by a smaller compactness factor filter (dilation removes small gaps inside the void that cause small shape irregularities); (iii). the third stage is used to remove vias while keeping all possible voids by using two feature parameters that are based on the area and the principal axis ratio of the segmented region. Fig. 4.23(h) shows all detected void and non-void regions inside the solder ball of Fig. 4.23(g) before applying the proposed classification procedure, while Fig. 4.23(i) shows the resulting detected voids inside the solder ball of Fig. 4.23(g) after using the proposed classification method.

4.3.3. Simulation results

The proposed method was applied to different Intel product lines: A, B, and C. Each one of these product lines has different challenges, in addition to die bond interference and inconsistent lighting, such as: (i). product line A has limited via interference, and has



(a).Input image with enhanced contrast for void visibility.

(b).Void detection results using existing algorithm in the 2D x-ray machine.

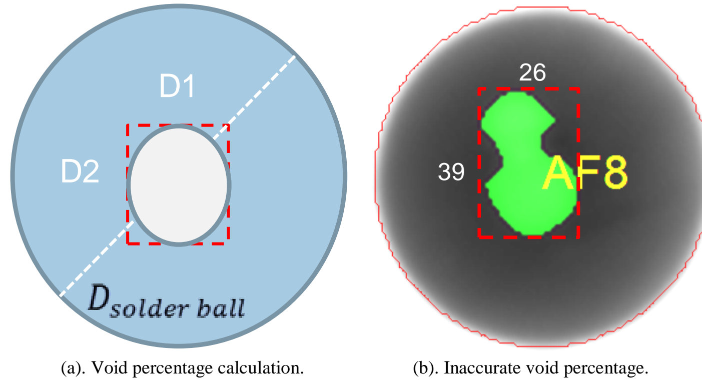
(c).Void detection results using the proposed algorithm.

Fig. 4.24. Comparison between existing void detection algorithm in 2D x-ray machine and proposed void detection algorithm in product line B.

a low void density per ball, (ii). product line B has limited via interference, and has a high void density per ball, and (iii). product line C has more via interference and medium void density per ball. The results of the proposed algorithm were compared to the results obtained from the existing latest 2D x-ray void detection algorithms that are provided as part of the x-ray imaging machine, the results obtained from 2D x-ray images by a trained operator, and the results obtained from 3D CT scan x-ray images which represents ground truth data.

4.3.3.1. Comparison with the data obtained from 2D x-ray embedded algorithm

The existing void detection algorithm in the 2D x-ray machine was applied to the three Intel products lines: A, B, and C. The performance of the algorithm shows satisfactory results in product line A, unsatisfactory results in product line B and produces a complete failure for product line C. Fig. 4.24 shows an example that illustrates the performance results of product line B for both the proposed method and the existing embedded algorithm that is provided with the 2D x-ray machine. From Fig. 4.24(b), it can be clearly seen that the existing embedded void detection algorithm produces false voids, misses a lot of voids, and does not process balls occluded by overshadowing components. In comparison, Fig. 4.24(c) shows that the proposed method outperforms the existing embedded void detection algorithm of the 2D x-ray machine and



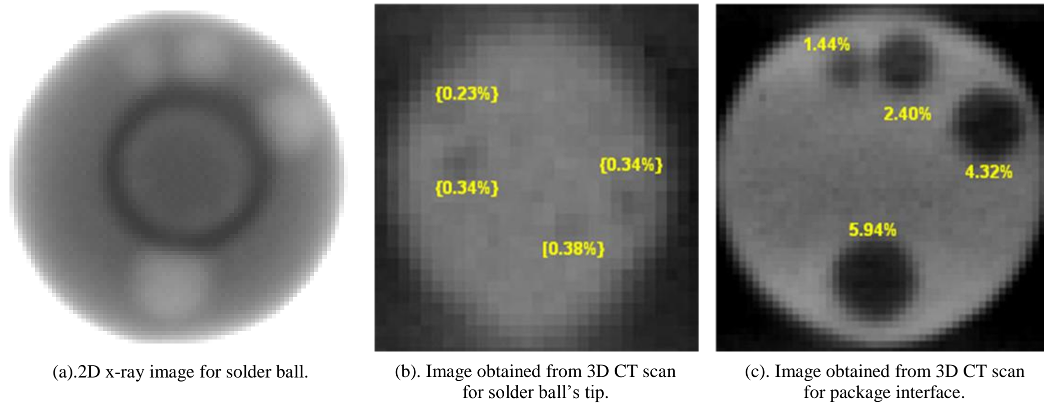
(a). Void percentage calculation. (b). Inaccurate void percentage.

Fig. 4.25. Manual void percentage calculation.

is able to detect voids in the occluded balls in addition to the ones in the non-occluded balls.

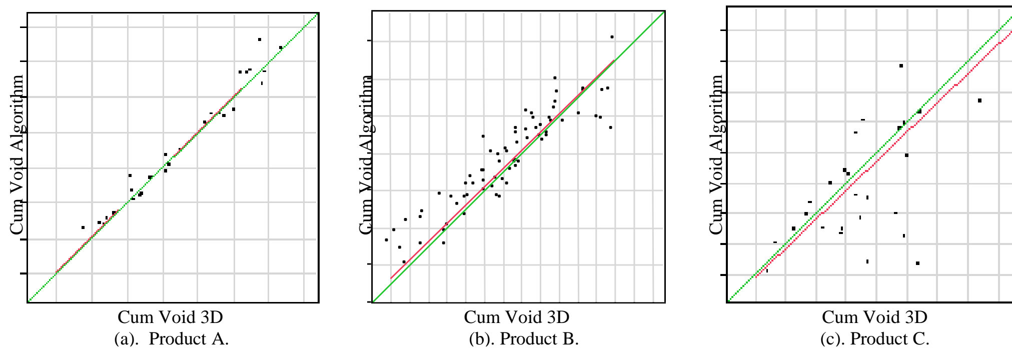
4.3.3.2. Comparison with the data obtained from 2D X-ray images by trained operator

Voids in many random balls in product lines A and B, were measured manually by trained operators. The operator takes 4 minutes to locate and measure one void in the ball. The operator locates and measures the void percentage by manually changing the 2D x-ray image intensity and contrast using an image processing program and then manually using a mouse to determine the extent of a void. The actual void percentage is the ratio of void area to the solder ball area. However the operator measures the void percentage by $\left(D_{av}/D_{solder\ joint}\right)^2$, where $D_{solder\ joint}$ is the solder ball diameter and D_{av} is the average of void width (D_1) and void height (D_2) as illustrated in Fig. 4.25(a). The operator method is accurate if the void has a circular shape, but it is not accurate if the void has an irregular shape such as the one shown in Fig. 4.25(b). In this latter case, the operator method produces an error equal to 2.7% of the actual void percentage value. The data obtained by the trained operators was compared to the data obtained by the proposed algorithm. The proposed method results in a correlation squared value with the operator data of 96% and 93% for product lines A and B, respectively, with no significant bias. It



(a).2D x-ray image for solder ball. (b). Image obtained from 3D CT scan for solder ball's tip. (c). Image obtained from 3D CT scan for package interface.

Fig. 4.26. 2D x-ray images versus 3D CT scan images.



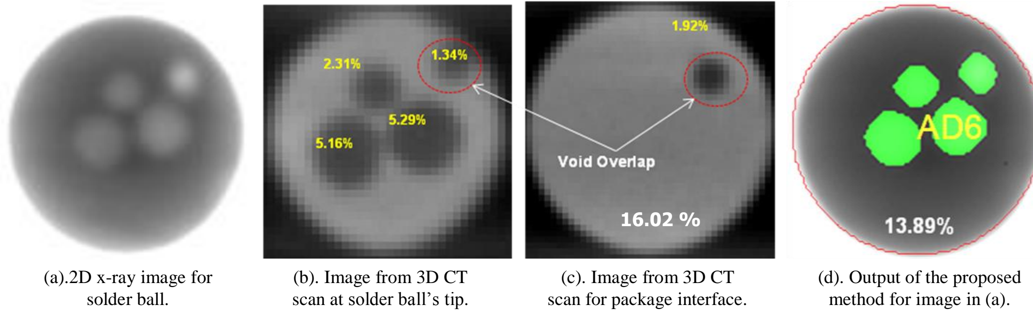
(a). Product A. (b). Product B. (c). Product C.

Fig. 4.27. Cumulative voiding comparison between proposed algorithm and manual 3D CT scan data.

should be noted that a correlation squared value greater than or equal to 75% with no significant bias corresponds to well correlated data.

4.3.3.3. Comparison with data obtained from 3D CT scan by trained operator

Using 2D x-ray images does not give information regarding the void's depth within the ball. Furthermore, the use of the 2D images will produce inaccuracies in the case of overlapping voids. Using a 3D CT scan allows the operator to have 2D images at different layers (depths) of the solder ball, which helps to see the isolated voids clearly without the interference of vias and die balls. Fig. 4.26(a) shows the 2D x-ray image where 4 visible voids around the via are present, while Figs. 4.26(b) and (c) show the 3D CT scan images at the solder ball's tip and package interface, respectively. Figs. 4.26(b) and (c) show 8 different voids which are seen as 4 voids in the 2D x-ray image due to the via interference which occludes a portion of those voids. The trained operators measure the visible voids in the 3D CT scan images using a manual calculation which produces



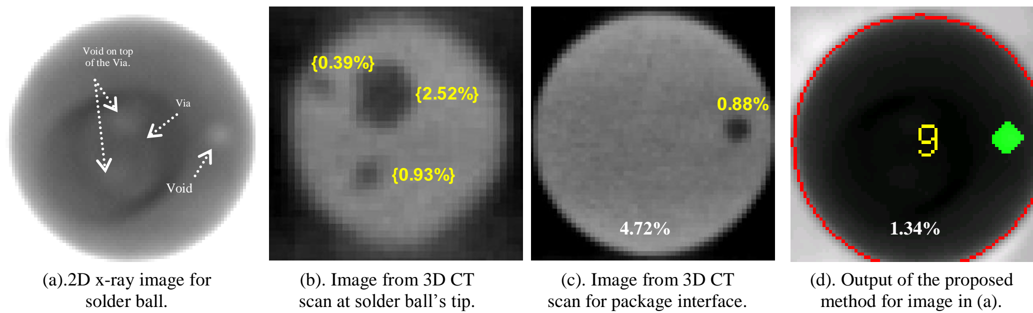
(a).2D x-ray image for solder ball.

(b). Image from 3D CT scan at solder ball's tip.

(c). Image from 3D CT scan for package interface.

(d). Output of the proposed method for image in (a).

Fig. 4.28. Comparison between the results obtained by the proposed method from 2D x-ray images and the results obtained by trained operators from the 3D CT scan images.



(a).2D x-ray image for solder ball.

(b). Image from 3D CT scan at solder ball's tip.

(c). Image from 3D CT scan for package interface.

(d). Output of the proposed method for image in (a).

Fig. 4.29. Limitations of processing the 2D x-ray images.

less error in this case due to the fact that the voids do not overlap and have circular shapes. Voids in many random balls in product lines A, B, and C were measured manually by the trained operators. The data obtained by the trained operator from the 3D CT scan images is used as ground truth and is used for comparison with data obtained from the proposed algorithm when it is applied to the 2D x-ray images. The proposed method results in a correlation squared with the ground truth data of 97%, 91%, and 77% for product lines A, B, and C, respectively, with no significant bias as shown in Figs. 4.27(a), (b), and (c), respectively. It should be noted that, for product line C, balls with voids occurring on top of vias were excluded from the statistics.

The mismatch between results obtained from 2D x-ray images and the results obtained from 3D CT scan images is due to the missing information in the 2D x-ray images as compared to the 3D CT scan images, especially if there are overlaps or via interference in the image. These are the limitations of processing 2D x-ray images. The following examples show the limitations of the proposed method when applied to 2D x-

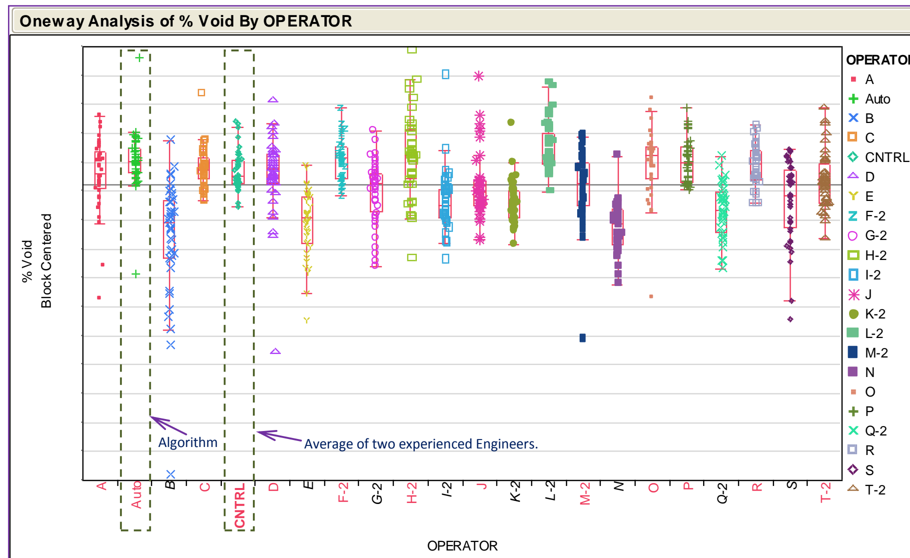
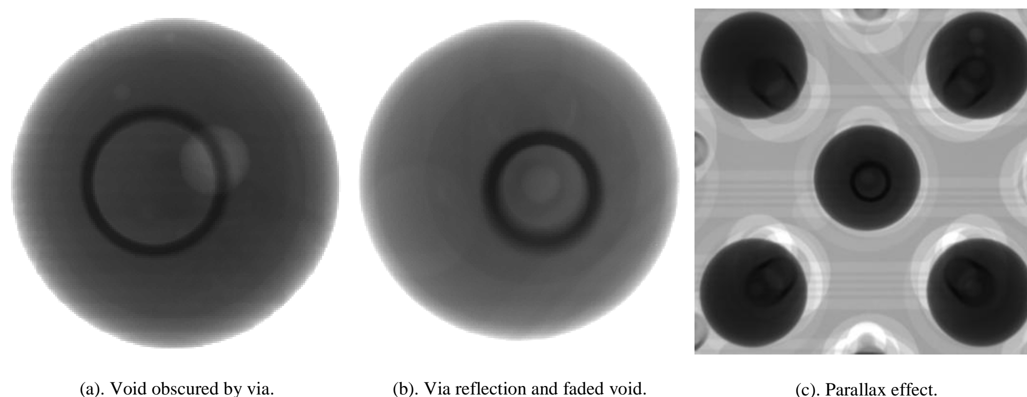


Fig. 4.30. Proposed algorithm (Auto) versus engineer: algorithm performance (Auto) compares well to the experienced engineer (CNTRL).

ray images. Fig. 4.28(a) shows the 2D x-ray image where 4 voids can be seen clearly. Figs. 4.28(b) and (c) show the images from the 3D CT scan images at the solder ball's tip and package interface, respectively. Figs. 4.28(b) and (c) show 5 different voids where 2 voids at different depths overlap and are seen as one void in the 2D x-ray image. The result of the proposed method is shown in Fig. 4.28(d). It can be seen that the proposed algorithm detects all visible voids in the input 2D x-ray image of Fig. 4.28(a). The proposed method gives a total void percentage equal to 13.89% versus 16.02% from the 3D CT scan ground truth data. The error between the two different methods is 2.34%, which is due to the overlapped voids and the human tolerance error of the operator. The proposed method detects the voids that are visible inside the 2D x-ray images and it cannot detect the overlapped void due to the limitations of the 2D x-ray image. Another example is given in Fig. 4.29 which illustrates the limitation of the 2D x-ray images when the voids lie on top of the via, which makes it hard to segregate the voids from the via due to the similarities of the gray level values in both the voids and the vias.

Fig. 4.30 shows the comparison between the results obtained by the proposed algorithm and trained operators. Many trained operators were asked to calculate the void



(a). Void obscured by via. (b). Via reflection and faded void. (c). Parallax effect.

Fig. 4.31. More challenges related to 2D x-ray images.

percentage for the largest void in an image manually for some random solder balls in product line C. The operators' results are compared with each other and show a 7% inconsistency. The result of two experienced engineers were averaged and given in Fig. 4.30 as "CNTRL". The result from the proposed algorithm ("Auto" in Fig. 4.30) compares well to the experienced engineer ("CNTRL" in Fig. 4.30). The proposed algorithm not only saves time, but also produces consistent results when compared to operators' results.

4.3.4. Void detection in the presence of vias and other artifacts in solder balls

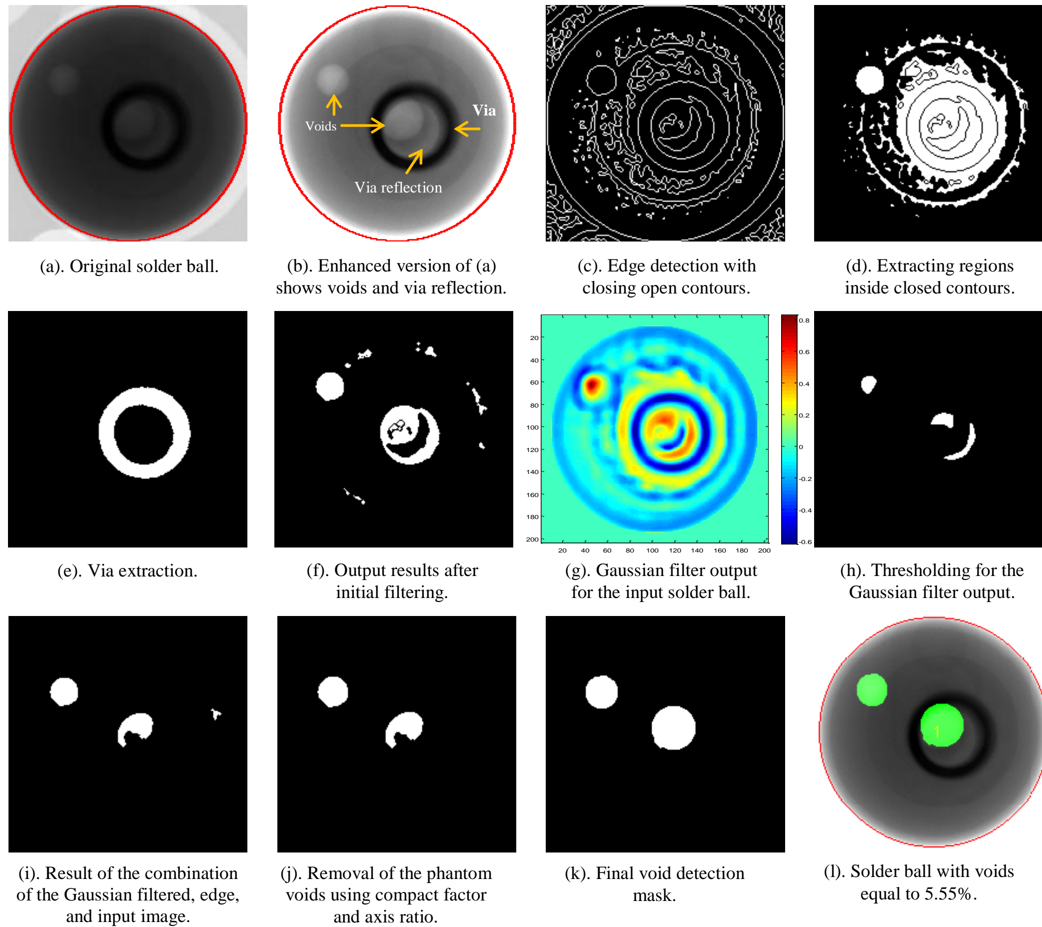
The presence of vias on substrates poses a large problem to manual or automated void detection. The void is often obscured either partially or totally by the via. Some of the new challenges in the void detection are shown in Fig. 4.31. If a solder ball is located near the edge of an image and happens to be positioned over a via, the parallax effect will spread and blur the via image and create additional difficulty in the detection of the via and any voids that are over or touching the via area as shown in Fig. 4.31(c). Reflections are often present inside the vias. The reflections can be caused by metal plating or other reflective effects. The reflections can easily be mistaken for voids because of their brightness and size. The reflections also often overlap with the voids themselves making the separation of the via and the via reflection a challenge for an expert operator or

machine. Based on our previous work [52, 53], we discovered that the challenge posed by the presence of vias in an image is the largest barrier to accurately detecting voids over a wide variety of product types. Phantom voids are often caused by the existence of traces (lines in the background area as shown in Fig. 4.31(c)) or other light-appearing areas in the background of the substrate. These phantom voids distort the total voiding percentage for an individual solder ball.

The new improved method consists of new components in addition to the components mentioned in the previous section. The new components include via extraction, separation between voids and via reflection, and feature extraction and classification of each candidate region in order to filter out all non-void areas. Fig. 4.32(a) shows a solder ball which has voids with some of the aforementioned issues such as: via, via reflection, inconsistent background lighting, and weak edges between the voids and background. An enhanced version of the input solder ball image shown in Fig. 4.32(a), is shown in Fig. 4.32(b) for illustration purposes. Voids and via reflections can be seen clearly in Fig. 4.32(b). Figs. 4.32(c) and (d) show the edge detection, using the LoG filter (Section 4.3.2.2), and the detected closed and filled contours inside the solder ball, respectively. The following subsections give details about the methodologies that are proposed to extract the via regions and to detect the voids in the presence of vias and via reflections.

4.3.4.1. Via extraction

An adaptive thresholding procedure that is based on the gray level distribution inside the solder joint is used to locate the via. The mean value (S_{mean}) and the standard deviation (S_{std}) of the gray level inside the solder ball are first calculated. Voids look brighter than the surrounded region with slightly higher gray level values than the mean inside the considered solder ball, while vias are darker than the neighbor area with a gray



(a). Original solder ball. (b). Enhanced version of (a) shows voids and via reflection. (c). Edge detection with closing open contours. (d). Extracting regions inside closed contours.

(e). Via extraction. (f). Output results after initial filtering. (g). Gaussian filter output for the input solder ball. (h). Thresholding for the Gaussian filter output.

(i). Result of the combination of the Gaussian filtered, edge, and input image. (j). Removal of the phantom voids using compact factor and axis ratio. (k). Final void detection mask. (l). Solder ball with voids equal to 5.55%.

Fig. 4.32. Procedures that show the steps of the proposed method for robust void detection.

level value much lower than the solder ball mean intensity. The presence of the vias increases the standard deviation value inside the solder joint. If the standard deviation (S_{std}) is relatively large (e.g., $S_{std} > 10$ in our implementation), this implies the possible presence of a via in the considered solder ball. Consequently, a threshold value $T_{via} = S_{mean} - S_{std}$ is used to detect candidate via regions that have a mean value less than T_{via} for further processing. The resultant regions are labeled to distinguish between each region. The gray level value, area, and compactness factor of each detected region, represent the main feature parameters that are used in the proposed method to determine which of the detected regions is a via. The largest area, minimum gray level, and a compactness factor greater than 1.5 are used to extract the via from the remaining

regions. The compactness factor is used to differentiate between regular and irregular shapes. As indicated before, the compactness factor is close to 1 for a circular shape and greater than 1 for non-circular shapes. Fig. 4.32(e) shows the extraction of the via that is present in the image shown in Fig. 4.32(a) using the above procedure.

4.3.4.2. Void detection in the presence of via and via reflection

In order to locate actual voids and eliminate false candidate voids, the proposed void detection system makes use of feature parameters to filter out phantom voids, via reflections, and undesired artifacts that are present in the solder balls.

First, feature parameters including the compactness factor and a local threshold parameter that is obtained using adaptive gray level thresholding, are used to filter out phantom voids. These feature parameters are calculated and applied to each extracted candidate void region after the edge detection process. The desired value of the compactness factor for void regions is chosen to be relatively large (≤ 2 in our implementation) in order not to avoid eliminating voids that overlap with other voids, or with via reflections, or with phantom voids. The adaptive gray level threshold value for each candidate void region inside the solder ball, is calculated independently (non-global thresholding). The threshold value is calculated based on the gray level distribution inside and outside each candidate void region. Consequently, the resulting threshold value is adaptive based on the location of the extracted regions with respect to the via. Three different cases of adaptive thresholding are considered: (1). when suspect voids are outside the via, (2). when suspect voids touch or overlap with the via, and (3). when suspect voids are located inside of the via. For each suspect void outside the via, the threshold value is taken to be the mean value of the area within two pixels outside the suspect void contour. For voids that touch or overlap with the via, the threshold value is calculated based on a fraction of the mean of the pixels that are located within two pixels in the vicinity outside of the void’s contour and that are not touching the via. For voids

inside the via, the threshold value is calculated based on a fraction of the gray level mean value of pixels located in the vicinity outside of the void's contour. In the two latter cases, the threshold value is taken to be a fraction of the mean (0.9) rather than the mean value of neighboring pixels due to the fact that the voids are relatively dimmer when they are located inside a via or when they overlap with a via. Any candidate void region that has a mean value less than the adaptive threshold value is filtered out. Then, the compactness factor is used to filter out more phantom voids and keep the regions that are more likely to be voids. Fig. 4.32(f) shows the output result that is obtained after filtering out false candidate void regions based on adaptive thresholding and the compactness factor. For comparison, Fig. 4.32(d) shows the input image before filtering is applied to the extracted regions. From Figs. 4.32(d) and 4.32(f), it can be clearly seen that the employed filtering based on the compactness factor and adaptive thresholding was able to successfully decrease the phantom voids and keep the candidate regions that are more likely to be voids. However, as indicated above, a relatively relaxed value was chosen for the compactness factor in order not to filter out actual overlapping voids. Consequently, as illustrated in Fig. 4.32(f), there are still some phantom voids after this initial filtering operation. Another issue is the overlap between voids and via reflections, which can produce phantom voids due to the similarities between the via reflection and the actual voids, as illustrated in Fig. 4.32(d).

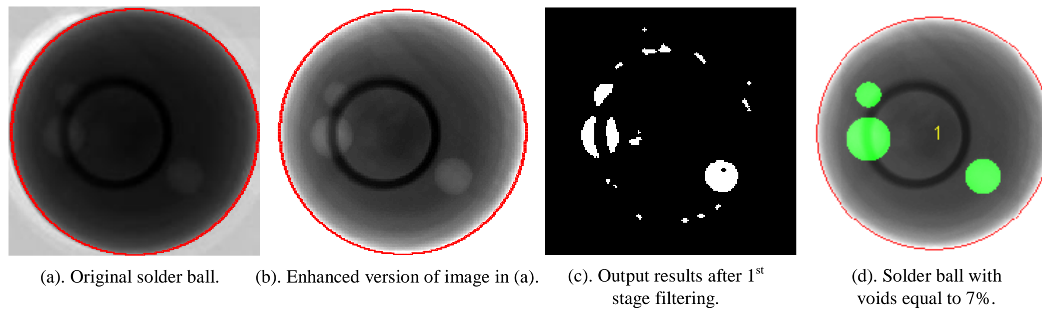
In order to separate between via reflections and voids, the original input image is smoothed using an 11x11 Gaussian lowpass filter with a standard deviation value of 1.5. The resulting Gaussian filtered image contains peaks in regions that correspond to actual voids, where the gray level gradually increases from the void boundaries to the void center. The output of the Gaussian filter, for the input solder ball shown in Fig. 4.32(a), is shown in Fig. 4.32(g). From Fig. 4.32(g), it can be observed that regions with brighter areas in the input image result in a higher intensity value in the Gaussian filter's output.

A void region results in a high intensity value after Gaussian filtering, while a via reflection region results in a relatively small intensity value compared to the void region. In addition, a via reflection region exhibits an irregular shape compared to a void region. A thresholding procedure is used to binarize the output of the Gaussian filter in order to extract the brighter regions. Based on experimental observations, it was found that a threshold value T_{gf} =0.4* max(intensity in Gaussian filter output), is able to separate between voids and via reflections as shown in Fig. 4.32(h). However, the segmentation output using the threshold value T_{gf} results in void regions with smaller areas than the actual void areas. A procedure that helps in growing the extracted small areas gradually without changing the number of regions in the image is used to get areas closer to the actual void areas. This is done by gradually decreasing the threshold value of T_{gf} while ensuring that the number of regions is not changing. For this purpose, the threshold value T_{gf} is decreased gradually, up to a minimum value equal to 0.3* max (intensity in Gaussian filter output), as long as the resulting number of detected void regions is kept unchanged after thresholding with T_{gf}. Then, candidate void regions that have a compactness factor with a value greater than 1.75, or a principle axis ratio less than 0.35 are filtered out to eliminate via reflections and to reduce further phantom voids. While performing the detection using the Gaussian filtered image helps in eliminating phantom voids due to via reflection, some void regions, with relatively low intensities, can be missed. In order to detect these missed void regions, a combination of the Gaussian filtered image, the edge detection image, and the input image is used with more constrained feature parameters. The binarized output of the Gaussian filter (Fig. 4.32(h)) is, first, combined with the region extraction image (Fig. 4.32(d)) by multiplying the two images pixel by pixel. The resulting product image represents a mask that has 1's for overlapping non-zero areas between the two images (binarized Gaussian filter and region

extraction images) and 0's elsewhere. The resulting mask represents the locations of candidate void regions and void-like artifacts, and is applied to the input image to extract these candidate regions. In order to classify the extracted candidate regions into void or phantom void regions, feature parameters including mean value and adaptive thresholding, compactness factor, and principal axes ratio are used as before but with more constrained values (threshold for adaptive threshold increased by 20%, compactness factor decreased to 1.4, and principle axes ratio decreased to 0.3). The resulting detected voids are shown in Fig. 4.32(i) where two voids and one phantom void were detected.

In order to remove further any detected phantom voids that were not eliminated by the previous operations due to similarities with the actual voids, a circularity constraint is imposed and is quantified using the compactness factor and the principle axis ratio. At this point, a compactness factor that is greater than 1.15 is used to filter out phantom voids while allowing voids with close-to-circular shapes in addition to partial or incomplete voids to be kept. The principle axis helps to filter out elongated shapes. Any shape that has a principle axis ratio less than 0.3 is filtered out. The resulting output image using these constraints on the feature parameters is shown in Fig. 4.32(j). From Fig. 4.32(j), it can be clearly seen that additional phantom voids were successfully eliminated as compared to Fig. 4.32(i). Finally, a procedure is used to check if any partial or incomplete voids are touching the via. In this latter case, the diameter of the partial void is estimated and a circle is drawn at the center point of the longest axis of the detected void. Fig. 4.32(k) shows an example for reconstructing a complete void from a partial void using the proposed procedure. Fig. 4.32(l) shows the final void detection result using the input image given in Fig. 4.32(a).

If any partial voids are separated by the via as shown in Fig. 4.33, a circle that includes both partial voids around the via, is drawn with a center equal to the centroid of a box



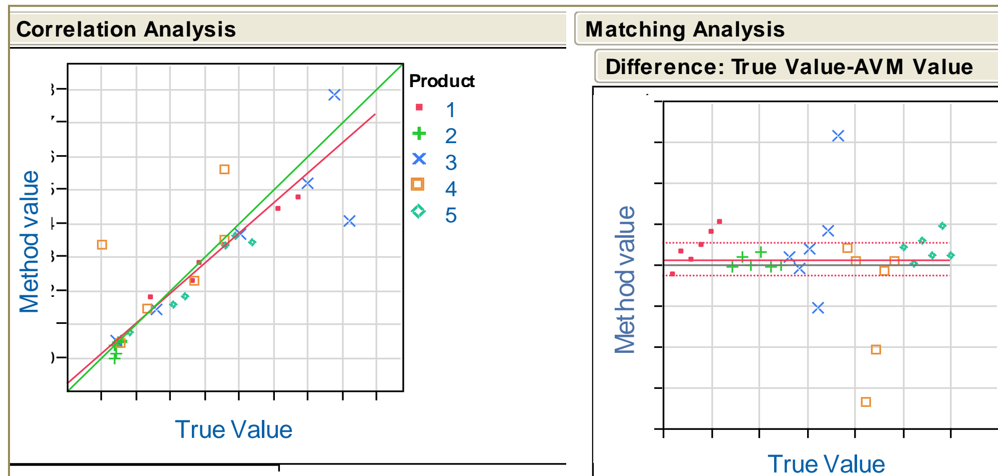
(a). Original solder ball. (b). Enhanced version of image in (a). (c). Output results after 1st stage filtering. (d). Solder ball with voids equal to 7%.

Fig. 4.33. Reconstruction of voids obscured by via.

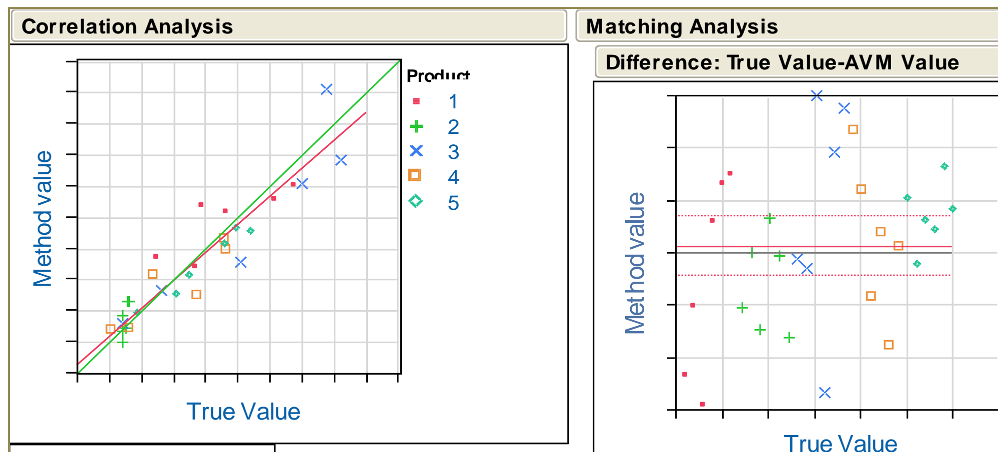
that includes the two voids and with a diameter equal to the longest distance between points in both partial voids. Fig. 4.33 illustrates the robustness of the proposed algorithm in its ability to accurately detect the obscured voids and to decrease phantom voids.

4.3.5. Simulation results for improved void detection method

The proposed method has been extensively tested on a wide variety of products. Each product has a different average void percentage ranging from a low void percentage to a high void percentage. Some of the tested products have issues related to via and other do not have vias-related issues. The proposed method was applied to all tested products without special tuning or tweaking for each product. This makes the proposed method robust and adaptive to the variations and to the different issues that are present in each product. Different x-ray tools were used to test the repeatability and reproducibility of each product in order to validate the robustness of the proposed method. A golden set of images from different products were used to show the performance of the proposed method in different x-ray tools. The matching results range from a correlation squared of 76% on Tool 1 to 86% on Tool 2 as given in Figs. 4.34(a) and (b), respectively, when compared with averaged measurements from experienced operators. The slope of the method and the manual average measurements are statistically equal to 1. The bias between the measurements is statistically equal to 0. These results show that the method performs well across tools. The repeatability and reproducibility of the results from the proposed method have also been proven. Figs. 4.35(a) and (b) show the repeatability



(a). Matching results from Tool 1.



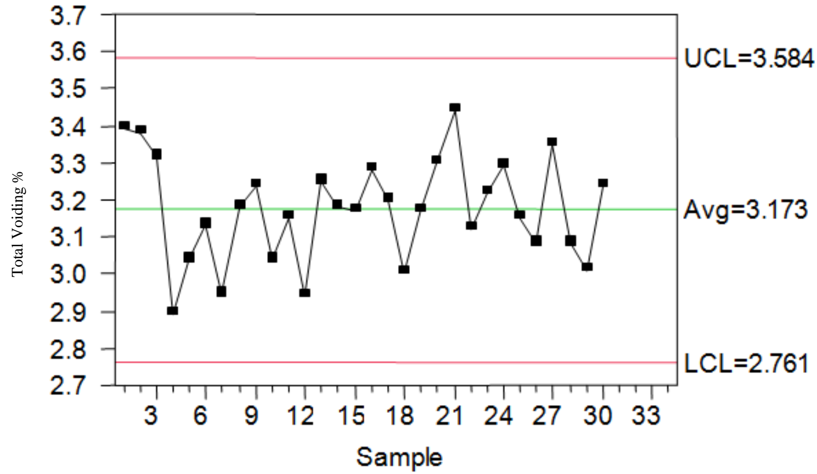
(b). Matching results from Tool 2.

Fig. 4.34. Comparison between the proposed method's void percentage result and result obtained by experienced operators for two different x-ray tools.

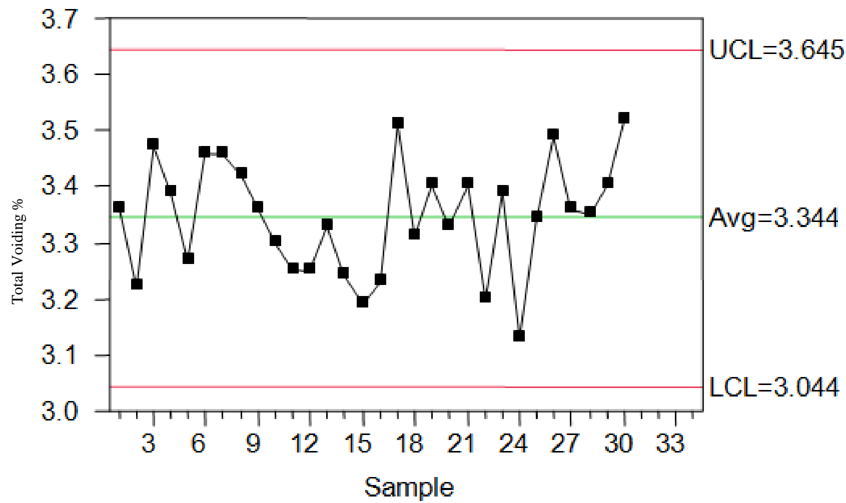
results for the same product taken with two different x-ray tools. The proposed method gives a standard deviation of void percentage equal to 0.5% in Tool 1 and 0.4 % in Tool 2. The differences between the results that are obtained when using Tools 1 and 2, are due to the differences in power, current, parameters setting, x-ray filament age, and other extra factors that contribute to the lighting.

4.3.6. Summary of void detection

A robust automatic void detection scheme was presented in order to allow automated inspection and automated manufacturing quality assessment. The proposed method is fully automated and can benefit the manufacturing process by reducing operator effort



(a). Tool 1 with void standard deviation of 0.5%.



(b). Tool 2 with void standard deviation of 0.4%.

Fig. 4.35. Repeatability results of the proposed method for one of the tested product captured by two different x-ray tools.

and process variability. The algorithm works for all products without specific tuning. The method has been implemented in a standalone PC that is configured to retrieve images automatically from a 2D x-ray system, thus reducing the time to run the method to the time the x-ray takes to produce the images. The proposed method can enable compliance to IPCA610D for cumulative voiding of 25% or less in post SMT solder joints and the new JEDEC guideline JC-14-1 for cumulative voiding in a sample of solder ball measurements not to exceed 15%.

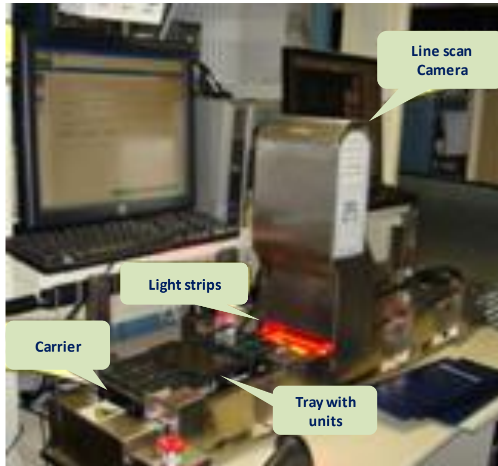


Fig. 4.36. Image acquisition using line scan camera.

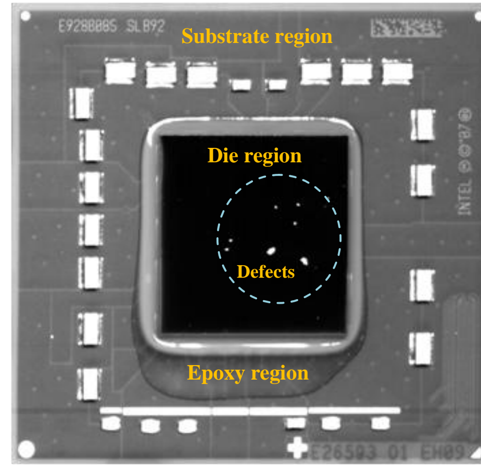
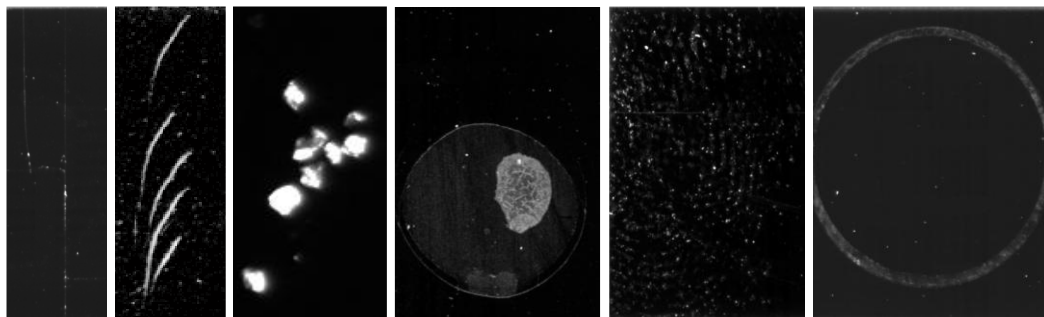


Fig. 4.37. Unit image with different region of interests.



(a) Cracks. (b) Scratches. (c)Foreign materials. (d) Stain. (e) Fingerprint. (f) Suction cup.

Fig. 4.38. Samples of defects in the die area of the semiconductor units.

4.4. Defect Detection and Classification in the Die Area of Semiconductor Units

This section deals with the detection and classification of defects in the die area of semiconductor units. Fig. 4.36 shows the adapted image acquisition system for imaging the semiconductor units. The image acquisition consists of (i). a line scan camera which takes images while the units carrier is moving; (ii). a tray which carries several semiconductor units; (iii). light strips to focus the light on the region that need to be scanned; (iv). a moving carrier that has a speed matching the speed of the line scan camera; and (v). a station controller to help in checking and communicating between these devices. An example of a unit image is shown in Fig. 4.37.

The main goal in these images is to extract the regions of interest, then detect and classify defects inside each one of these regions of interest. The regions of interest (ROIs)

in the considered example are: die region, epoxy region, and substrate region, which are shown in Fig. 4.37. These regions of interest exhibit different types of defects including: crack, scratch, foreign material, stains, fingerprints, suction cup, and other defects as shown in Fig. 4.38. Most of the defects usually occur in the die region; therefore, our main goal in this section is to automatically detect and classify defects inside the die area (ROI) of each unit image.

Related work on defect detection in semiconductor units includes the work performed at Oakridge National Lab (ORNL) as described in [173-179]. The related work at ORNL makes use of feature parameters to represent defects by using a specified known mask. The employed wafer level feature parameters are not directly applicable to the considered semiconductor units (SU) images. In some of the ORNL published schemes, defects are detected by taking the difference between the reference (non-defective) image and a defect image. This latter approach would not work for the considered SU images as there is no valid reference image that can be directly used due to the following issues:

- Light variation: In most SU images, the light distribution is not uniform. Moreover, the light reflection in epoxy, during scanning, is not the same in all cases, which makes the images quality not consistent.
- Misalignment of the SU images.
- The epoxy area varies from one image to another, which can result in the detection of false defects if a subtraction procedure is used to segment defects.
- The die areas vary from unit to another due to noise, size, misalignment, and other factors.

Other related work on defect classification includes paper surface defect classification in the paper manufacturing process [180-183], a visual inspection system for assessing printing quality [184], raw textile defects [185, 186], fabric defect classification [187, 188], integrated circuits and wafer pattern defects [189-196], wood defect classification

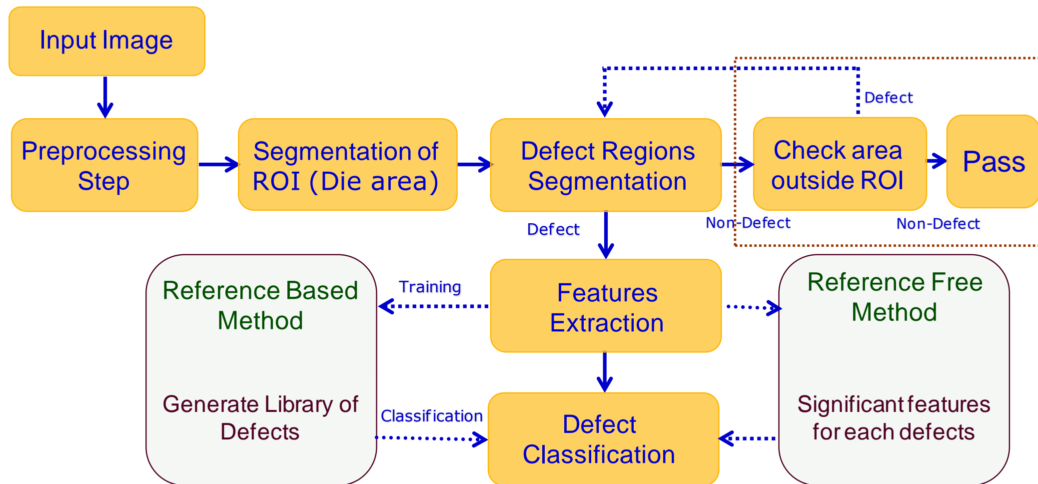


Fig. 4.39. Block diagram of the proposed automatic defect classification scheme in unit images.

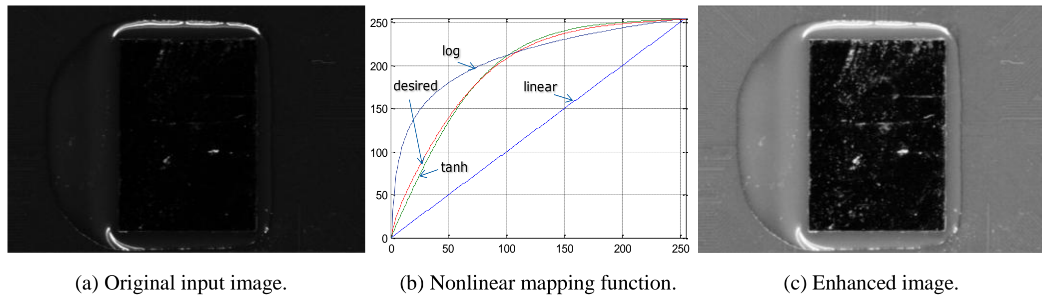
[197-202], and steel/metal defect classification [203-211]. The above methods consider different applications and are not suited nor optimized for the considered problem of automatically detecting and classifying SU defects using the considered SU images.

Fig. 4.39 shows the proposed scheme for defect detection and classification in unit images. The proposed scheme consists of different steps: preprocessing, ROI segmentation, defect detection, feature extraction, and defect classification. The classification step can be used to classify each defect using either a reference-free method or a reference-based method, as described later. Details about the steps of the proposed scheme are provided in the following subsections.

4.4.1. Preprocessing step

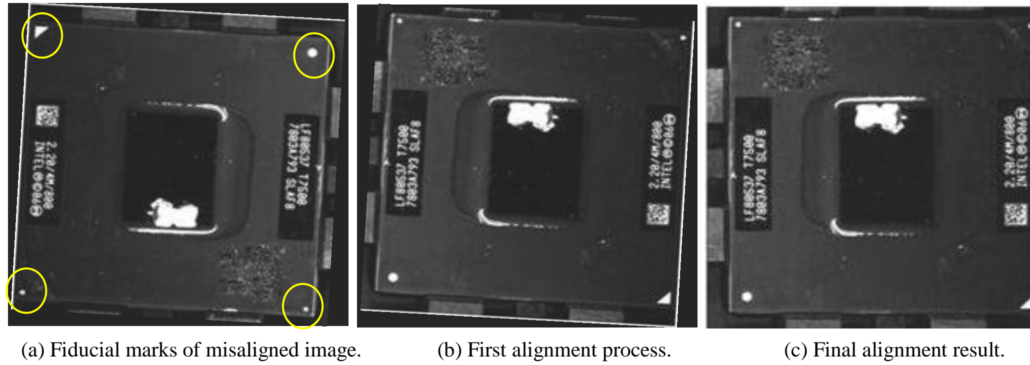
The considered images pose various challenges such as different lighting conditions, noise and artifacts, misalignment, and faded defects. These challenges have to be solved before applying any detection or classification procedures. A preprocessing step is used to eliminate some of these challenges.

Due to the image capturing process using a line scan camera, unit images exhibit light variations due the non-uniform light distribution from the light strips that are attached to the camera system as shown in Fig. 4.36. The lighting variation in the



(a) Original input image. (b) Nonlinear mapping function. (c) Enhanced image.

Fig. 4.40. Light enhancement using nonlinear mapping.



(a) Fiducial marks of misaligned image. (b) First alignment process. (c) Final alignment result.

Fig. 4.41. Image alignment using fiducial marks.

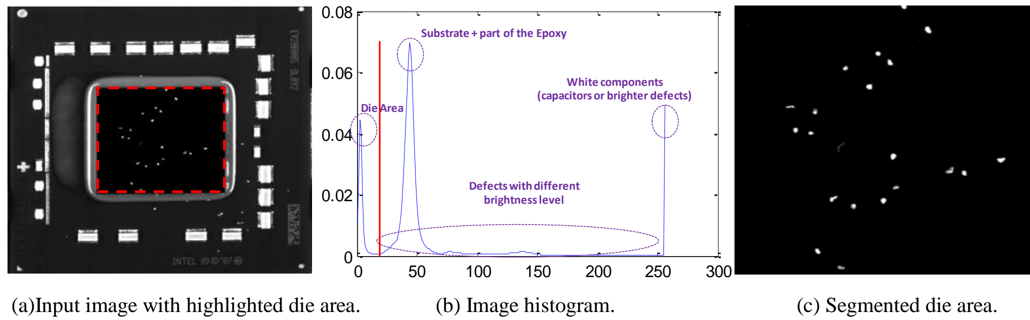
produced images is considered as one of the main challenges in the semiconductor unit images. An example of inconsistent light variation is shown in Fig. 4.40(a) where the image exhibits light variations in addition to an invisible defect. Solving this problem is an important issue before proceeding to the next steps. The problem can be handled by using a smooth and non-linear mapping function to normalize the input gray level image to be between 0 to 255. There are many different ways that can be used to improve the lighting issue such as using normalization (linear mapping), log function, or tanh function. The log function puts more weight on dark regions and less weight on bright regions in the image. The tanh function puts less weight on dark regions and more weight on bright regions as shown in Fig. 4.40(b). A nonlinear light enhanced function was derived by combining both tanh and log functions with different weight to get better light enhancement. A weight with a value equal to 0.3 is used for the log function, while a weight with a value equal to 0.7 is used for the tanh function. Those weight values were selected based on an experiment that was conducted on several images that were scanned

under the same settings of the line scan camera. Fig. 4.40(c) shows the enhanced image using the proposed non-linear mapping function.

During the scanning of the semiconductor units (SUs) using a high resolution camera, the units can be rotated or placed in any orientation. Automatic image alignment is needed to align the scanned images of the semi-conductor units that are being manufactured. This alignment step is necessary to locate the region of interests and to facilitate efficient defect detection and classification. One of our proposed approaches is to use a two-step alignment strategy. In this proposed approach, first, the fiducial marks in the considered scanned image need to be automatically located, segmented, and identified. As shown in Fig. 4.41(a), there are different combinations of fiducial marks located in the four corners of the considered scanned image: two small circles, one large circle, and one triangle fiducial marks. In the first alignment step, the considered image is rotated by an angle $\theta \in \{0^o, 90^o, 180^o, \text{ or } 270^o\}$ so that the triangular fiducial mark is located near the bottom right. The triangular fiducial mark is identified by comparing the compactness factor value of each segmented fiducial mark. The second alignment step ensures that the two small circular fiducial marks are aligned on the same horizontal line. This can be performed by computing the angle from the horizontal axis of the line between the centroids of the two small circular fiducial marks and then rotating the image by this angle. The rotation angle in the second alignment is less than 1 degree in most cases due to the limited deviation of the imaged tray on the moving carrier (Fig. 4.36). Fig. 4.41(c) shows the final alignment result using the proposed method.

4.4.2. ROI segmentation

The main goal is to automatically detect and identify the defects that are located in the regions of interest in the considered semiconductor images. For this purpose, these region of interests (ROIs) need to be automatically detected and segmented for further analysis. Fig. 4.42(b) shows the histogram of the input image shown in Fig. 4.42(a). The



(a)Input image with highlighted die area. (b) Image histogram. (c) Segmented die area.

Fig. 4.42. ROI segmentation using histogram analysis.

histogram shows 3 main clustering regions: (i) the first clustering region exhibits a low gray level value which represents the die area, (ii) the second clustering region exhibits a medium gray level value which represents the epoxy and substrate regions, (iii) the third clustering region exhibits a high gray level value which represents the bright regions in the image such as defects, epoxy reflection, and capacitor components. The region between the second and third clustering regions represents defects with different gray level values.

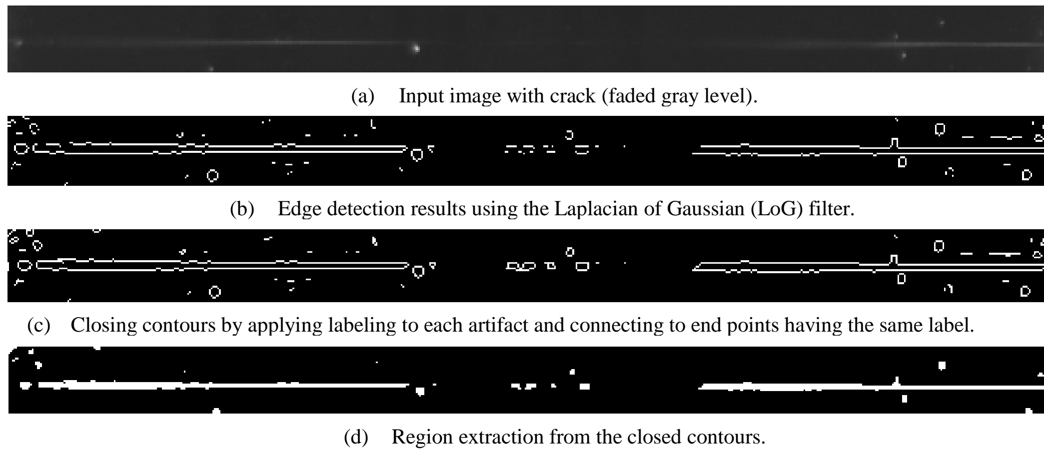
The die area is usually contaminated with most of the defects in the considered semiconductor units; therefore, our focus in this section is to segregate the die area and use it as our main region of interest. The die area corresponds to be first clustering region from the histogram analysis shown in Fig. 4.42(b). An adaptive threshold that lies between the first and second cluster regions need to be estimated in order to segment the die region. To calculate the adaptive threshold, the mean value of each cluster region has to be calculated first. This can be achieved by using one of the clustering approaches [13-18] to locate the centroids of each cluster, and then calculate the mean value of each cluster, which corresponds to the peak of each clustering region in the histogram. In our case, the k-means algorithm [18] is used to estimate the cluster centroids of the three regions in the considered images. Three initial randomly selected centroids are used in the k-means algorithm to locate the actual centroids. The obtained estimated centroids from k-means are used to locate the nearby peaks in the histogram. The segmentation

threshold is then computed to be midway between the first two histogram peaks and is used to segment the die area. Segmentation using the proposed adaptive thresholding scheme produces an image with regions that have gray level values less than the estimated threshold value. This can produce undesired artifacts inside and outside the die area due to non-die areas with low gray level values. However, these artifacts can be easily eliminated due to their small sizes with respect to the die area size. Mathematical morphology operations [164, 165] such as dilation, erosion, opening, and closing can be used to successfully eliminate undesired artifacts. Fig. 4.42(c) shows the final ROI segmentation results for the die area in the SU image shown in Fig. 4.42(a). The proposed algorithm was tested on more than 1000 images that were taken using 8K and 12K line scan cameras under varied lighting conditions, and was found to yield 100% accurate ROI segmentation.

4.4.3. Defect detection

In the proposed defect detection method, once the die area (ROI) is segmented, an edge detection method is first performed to locate the contours of the defects inside the ROI. Edge detection helps in locating the pixels where the gray level difference between neighboring pixels is significant. The output of the edge detection process highlights the regions inside the ROI with significant gray level differences, and each region is represented by its edge contour. The Laplacian of Gaussian (LoG) edge detection method followed by contour closing, as described in Section 4.3.2.2, is used here to locate the edges of defects.

Fig. 4.43(a) shows a portion, inside the die area, which represents a crack defect with faded areas in the middle. Applying the edge detection procedure using the LoG to the image shown in Fig. 4.43(a) produces an image with edges as shown in Fig. 4.43(b). The resulting image has many open contours and these open contours have to be closed first in order to extract the regions for classification. Fig. 4.43(c) shows the results after



(a) Input image with crack (faded gray level).

(b) Edge detection results using the Laplacian of Gaussian (LoG) filter.

(c) Closing contours by applying labeling to each artifact and connecting to end points having the same label.

(d) Region extraction from the closed contours.

Fig. 4.43. Edge detection and contour closing applied to a portion inside the die area.

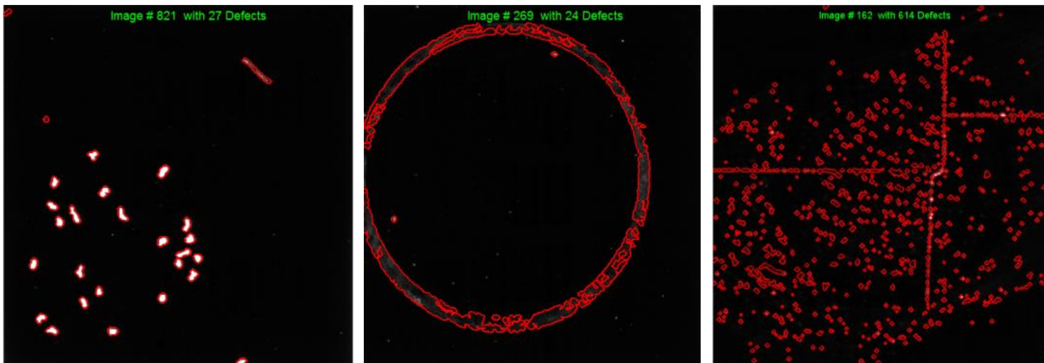
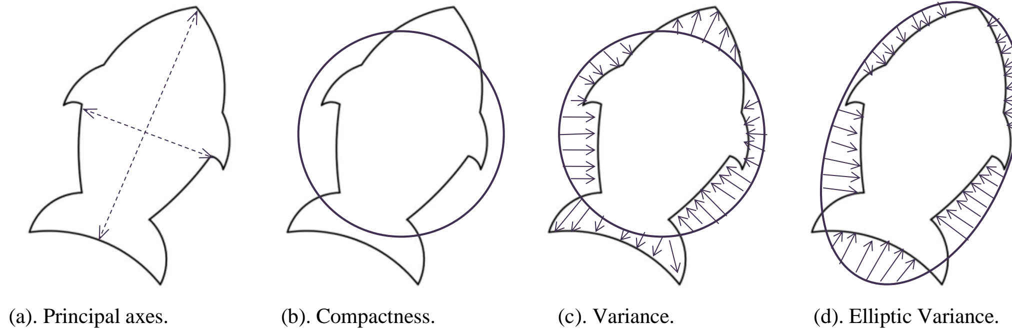


Fig. 4.44. Defect detection using the proposed method.

applying the contour closing procedure. Each region is extracted easily by using the extracted closed contour, which is shown in Fig. 4.43(d). Closing the open contours is a very essential step for the detection and classification of defects, especially crack. Fig. 4.44 gives some examples of defect detection using the proposed method.

4.4.4. Feature extraction

In order to classify each defect, we need to represent each defect with unique signatures. These signatures are unique features that are used to well define each defect. Representing each defect with its significant feature parameters results in a concise representation of the defect using a vector with few parameters. Feature parameters should have characteristics that make them rotation-invariant, shift-invariant, and scale-invariant. In this section, we will give few examples of feature parameters that can be



(a). Principal axes. (b). Compactness. (c). Variance. (d). Elliptic Variance.

Fig. 4.45. Shape-based feature extraction.

used in describing different defects in the considered SU. In the considered SU images, we extracted several shape-based and histogram-based feature parameters for each defect. The combination of shape-based and histogram-based features is important to have a better representation of defects. Some defects have similar gray level values (histogram-based) but they have different shapes and vice versa.

Fig. 4.45 shows some examples of feature extraction using shape-based parameters [212-214]. The principal axes ratio is the ratio of the shape's minor axis to the shape's major axis as shown in Fig. 4.45(a). The major axis of a shape is its longest diameter, a line that runs through the centre of the shape and its ends being at the widest points of the shape. The minor axis is perpendicular to the major axis and is an axis of symmetry. The principal axes ratio is used in distinguishing between different shapes based on the elongation. The compactness factor is computed as $CF = p^2/4\pi A$, where p and A represent, respectively, the perimeter and the area of the considered shape. If the region has a shape close to a circular shape, its compactness factor is close to "1". The variance in Fig. 4.45(c) is the squared standard deviation between the considered shape and a regular circle shape with a center located at the shape centroid and with an area equal to the shape area. The calculation of the elliptic variance in Fig. 4.45(d) is similar to the calculation of the variance in Fig. 4.45(c), except that an elliptical shape is used instead of a circular shape.

The aforementioned feature parameters are important to distinguish between different

defects based on shape analysis. Other features such as curvature, eccentricity, elongation, template matching, and other shape descriptors can be added to the shape-based feature extraction. Histogram-based feature extraction can be represented by mean value, entropy, dispersion, dynamic range, bending energy, kurtosis, and many other feature parameters [173-179, 215, 216].

Feature parameters have different numerical values and units. It is highly recommended to normalize the feature parameters to have the same dynamic range, i.e, [0 1]. Normalization is a very critical step in order to have a balanced representation of each defect during the classification process. In our case, we normalize each feature parameter to have the same dynamic range between 0 and 1. The area of a detected defect can be normalized to be between 0 and 1 by using the die area, and the gray level mean value can be normalized by using the maximum gray level value in the image (e.g., 255 for an 8-bit image).

4.4.5. Defect classification

The extracted feature parameters are used to identify the type of the defects by assigning each defect to a known class. There are two approaches to get an automatic classification: refrence-based method and refrence-free method. In the reference-based classification method, we use information from stored or trained data for the classification. The performance of this method is heavily dependent on the database library that is used for training and classification with different forms of defects. In the reference-free classification method, we don't need to have a library of defects. The classification is based on locating some key signatures (features) that are sufficient to classify the defects. The following subsections describe the proposed reference-based and reference-free classification methods.

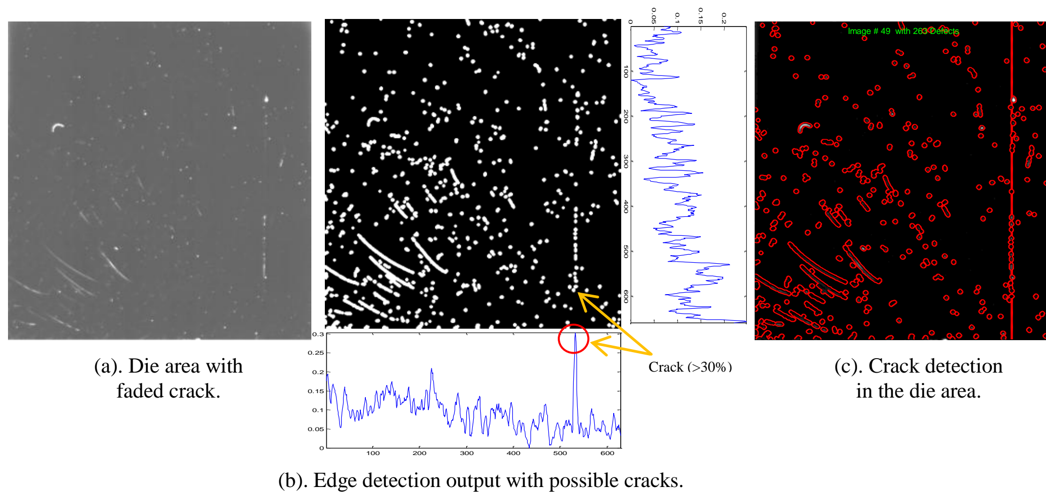
4.4.5.1. Reference-based classification

The reference-based method requires a library of defects that can be used for classification as discussed above. Feature parameters are used to build the database of all desired defects. The library should be constructed and updated so that all types of detected defects can be verified and classified correctly. Moreover, the library structure should be represented by feature vectors, where each vector is represented by feature parameters for different defects with different forms. The library is generated during the training process as shown in Fig. 4.39. In the training process, an operator looks at each detected defect and classifies it manually to build the library of defects. It is highly recommended to have more samples of each defect during the training (learning process) in order to have accurate classification. During the classification process, the feature vector of the considered defect is compared to each feature vector in the stored library of defects by computing a distance between the two feature vectors. The considered defect is classified as the stored defect that results in the minimum distance.

4.4.5.2. Reference-free classification

The reference-based method of Section 4.4.5.1 has many disadvantages including the facts that it is time consuming, and that it requires creating a library of defects, which in turn requires training and storage. The result of the reference-based classification method is heavily dependent on the library of defects which requires a large training data set that covers all possible defects. These issues can be eliminated by using a reference-free classification method.

The main challenge in the reference-free classification method is to find significant feature parameters (descriptors) that well define each defect in order to have an accurate classification. There are three main defects that occur frequently in the die area of the SUs: cracks, foreign materials (FM), and scratches. Examples of cracks, scratches, and FM defects are shown in Fig. 4.38 (a), (b), and (c), respectively. Each one of these



(a). Die area with faded crack.

(c). Crack detection in the die area.

(b). Edge detection output with possible cracks.

Fig. 4.46. Procedure for crack detection.

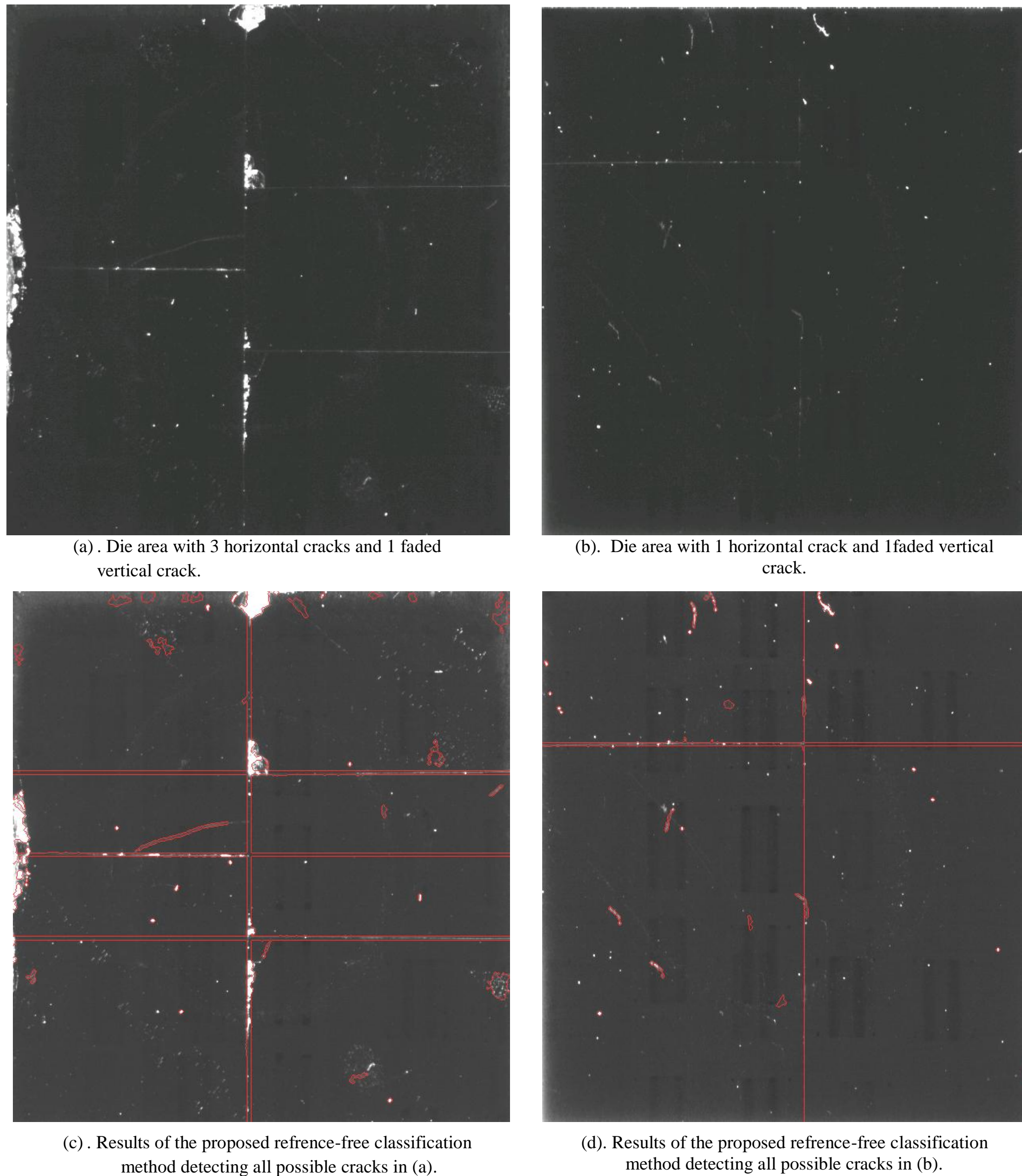
defects has significant feature parameters that qualify it for reference-free classification. For example, the crack defect has significant feature parameters based on the fact that it goes form one edge to another edge of the die, and is very thin compared to other defects. The foreign material defect has some unique properties including having uniformly distributed gray levels, a brighter mean intensity compared to other defects, and a close-to-circular shape. The scratch defect has a bright gray level in the center and the intensity gradually fades by moving away from the center to the edges of the scratch. In addition, the scratch defect has an elongated shape compared to FM. These three main defects are well defined by some feature parameters and can be classified using a reference-free method.

4.4.6. Simulation results

The existence of a crack on a semiconductor unit results in the failure of the unit. Thus, it is very important to check for cracks on the units before sending these to customers. A crack is one of the toughest defects to detect in the die area because it is very thin and it requires a high resolution camera to see it clearly. In addition, a crack does not have a consistent brightness and it exhibits fading in some sections, which makes it very hard to locate. On the other hand, as indicated previously, a crack has

unique and nice properties such as linearity due to the crystal structure of the die (can be horizontal or vertical), and the fact that it goes from one side of the die to the other side. Few assumptions were made to detect the cracks such as alignment in the horizontal or vertical direction, and having a length greater than or equal to 30% of the width or the height of the die. The 30% assumption is used due to the fact that, using the imaging set-up as described at the beginning of Section 4.4, a crack has many fading regions, which makes it hard to locate the whole crack. Fig. 4.46(a) shows the die area with a faded crack. After extracting all the defects in the die area, we check each horizontal and vertical line in the binary image to locate the ones that are greater than or equal to 30% of the width and/or the height of the die area. By checking the thickness and the length of the detected defect, one can identify the crack as shown in Fig. 4.46(b). Fig. 4.46(c) shows the die area with the identified crack in addition to other detected defects.

Some statistics were collected to show the performance of the crack classification. The reference-free classification method was tested on 261 images with cracks. From the obtained statistics, there were 3 misses out of 261, which gives a 99% classification rate. The 3 misses were due to the fading and low visibility of the cracks. Fig. 4.47 shows different examples of crack detection. The image shown in Fig. 4.47(a) has 3 horizontal cracks and one faded vertical crack. The image in Fig. 4.47(b) shows the die area with one horizontal crack and 1 vertical crack (barely visible). The result of the proposed reference-free classification method for the images in Figs. 4.47 (a) and (b) are shown in Figs. 4.47(c) and (d), respectively. The proposed method was able to successfully classify all the possible cracks in the given images as shown in Figs. 4.47(c) and (d). The images in Fig. 4.47 correspond to enhanced versions of the original images for illustration purposes; however, the algorithm is applied directly on the original images without any enhancement.



(a) . Die area with 3 horizontal cracks and 1 faded vertical crack.

(b). Die area with 1 horizontal crack and 1faded vertical crack.

(c) . Results of the proposed refrence-free classification method detecting all possible cracks in (a).

(d). Results of the proposed refrence-free classification method detecting all possible cracks in (b).

Fig. 4.47. Examples showing the performance of the proposed refrence-free classification method in classifying cracks inside the die area of semiconductor units.

In this section, we proposed a scheme for automatic defect detection in the die area of semiconductor units. The proposed detection scheme can be combined with a reference-based or reference-free classification method. Performance results of the proposed method were presented for crack detection and classification, and show a high correct classification rate for cracks. The proposed method can be easily extended to classify other defects.

4.5. Summary

A robust and automatic defect detection methodology is very desirable in many industrial applications. This helps significantly in reducing the cost, decreasing time consumed during manual investigation, improving the quality, and producing more reliable and accurate results. In this chapter, three automated defect detection and classification methods were proposed for three different applications to automatically detect and classify defects in semiconductor unit images. The first proposed scheme is for segmenting and classifying each solder ball of the processor sockets as defective (Non-Wet) or non-defective. The method gives a 96% detection rate and saves 89% of the operator time. The second proposed scheme is used for detecting voids inside solder balls in different boards. The method produces a high correlation rate and matching with the ground truth data. The third proposed scheme is used to detect different defects in the die area of the semiconductor unit images such as cracks, scratches, foreign materials, fingerprints, and stains. The proposed three methods, which were discussed in this chapter, result in a higher accuracy compared to the results obtained by using the existing high cost state-of-the-art machines, and are inexpensive to implement.

CHAPTER 5: CONCLUSION

This research work contributes to image segmentation, analysis, and classification and their applications in natural, texture, biomedical, and industrial images. Issues related to level-set, multi-region and texture image segmentation, cell evolution analysis in biomedical images, and defects detection and classification in semiconductor unit images were investigated in this thesis. This chapter summarizes the major contributions of this work and discusses the possible future extensions.

5.1. Contributions

This work proposed robust and noise resilient approaches that resulted in a significantly improved multi-region and texture image segmentation and classification, and their applications in biomedical and industrial images. The main contributions of the presented work can be summarized as follows:

- Multi-region and texture image segmentation:

  A noise resilient multi-region and texture image segmentation based on a level-set framework with constraints was proposed. The proposed method is less sensitive to initializations and exhibits faster convergence as compared to existing multi-region level-set segmentation schemes. Simulation results using synthetic, texture, medical, and natural images were presented to show the robustness of the proposed method in segmenting multi-region images with any number of regions and under different initializations.

- Cell Evolution Analysis:

  Two Cell Evolution Analysis schemes were proposed for cancer cell images. Cell migration analysis was performed by tracking the cell cluster region at different time points. The cell cluster region is segmented based on two different proposed segmentation schemes: i) Piecewise image segmentation: weighted level-set and two step segmentation methods; and ii)Texture image segmentation using tensor vector and

Bayesian formulation with a fast *à* trous wavelet filter for preprocessed texture images. Both of the proposed schemes were successfully tested on different cancer cell images with low contrast, cell overlap, and noise. The second scheme is more accurate than the first scheme, especially when the cell image contains different artifacts and noise. Cell proliferation and dispersion analysis was performed by using our proposed individual cell segmentation and counting schemes. The proposed individual cell segmentation is devised based on the distribution of gray level values of the background and the gray level distribution analysis inside individual cells. An adaptive and robust segmentation threshold was calculated based on gray level distributions to extract individual cells' centroids for counting.

The developed Cell Evolution Analysis system was tested in two different labs from two leading companies that conduct work in cell migration analysis and drug discovery. The reports [82] from these two companies have strong recommendations and show that the developed CEA system produces accurate results compared to manual results by expert operators. In addition, it is fully automated, which helps in saving time and cost. The proposed method is included in a filed US patent [80, 83].

- Non-Wet defect detection and classification in processor sockets:

Non-Wet defects occur in solder joints in processor sockets and can cause motherboard failures. In the proposed non-wet detection method, an automatic defect detection algorithm was presented to accurately locate the non-wet solder joints in processor sockets. The conducted performance evaluation and resulting statistics show that the proposed algorithm provides a detection rate of 95.8%. This is 21% to 53% better performance than commonly used state-of-the-art inspection machines. The proposed algorithm has the potential of resulting in a detection rate of 100% with an operator in the loop due to the identification of joints that are adjacent to the missed defective ones. Moreover, the proposed scheme reduces the false positive rate, false negative rate, and

the inspection time and requires only a relatively inexpensive 2D x-ray imaging device, which makes it cost effective and manufacturing friendly.

- Voids detection and classification in solder joints and solder balls:

  Void defects occur in solder joints and solder balls and can cause incorrect scrapping and rework in addition to board failures. In the proposed void detection method, a scheme was presented in order to allow automated inspection and automated manufacturing quality assessment. The proposed method is resilient to noise, via, via-reflection, artifacts, and different challenges on solder joints. Moreover, the method is fully automated and can benefit the manufacturing process by reducing operators' effort and process variability. The proposed method shows high performance in terms of detection rate compared to the results obtained from the ground truth data from a 3D CT scan.

- Automatic defects detection and classification in the die area

  For the automatic defect detection and classification in semiconductor units, we proposed a scheme for the detection of defects in the die area. The die area was segmented based on an adaptive and noise resilient proposed segmentation method. The proposed die area segmentation is robust to lighting, alignment, noise, and artifacts. The defects inside the die area were detected using a modified edge detection and contour closing scheme. The proposed defect detection scheme in the die area can be designed based on reference-based or reference-free classification methods. Simulation results show that the proposed method gives a high classification rate of 99% for detecting crack defects.

5.2. Future Work

The image segmentation and classification area has a lot of interesting applications. The work presented in this thesis can be extended and optimized for different applications. Future possible directions of the presented work include the following:

- Multi-region and texture image segmentation:

  Possible improvements to the proposed multi-region and texture segmentation method include detecting the number of regions automatically to improve the performance of the method, and developing an automatic initialization for different regions in order to significantly decrease the sensitivity to random initialization.

- Cell Evolution Analysis:

  Future possible work in this area includes individual 2D cell tracking, 3D cell tracking, differentiation between live and dead cells, and designing mathematical prediction models of migration analysis.

- Non-Wet defect detection and classification in processor sockets:

  Future directions in non-wet defect detection include investigating the applicability of the proposed algorithm to multiple processor sockets and ball grid arrays, and locating non-wets for irregular structures of solder joints of processor sockets. Other future work include considering new features beside the angular deviations of the joints' centroids, and characterizing other types of defects in solder joints.

- Voids detection and classification in solder joints and solder balls:

  More investigations can be conducted to improve the methodology of void detection including the detection of voids for irregular sizes of solder balls and solder joints in the same image, differentiating between overlapped voids based on light contrast, and automatic joint labeling.

- Automatic defects detection and classification in the die area

  Possible future directions in the area of defects' detection in the die area of semiconductor units include extensions of the current method to classify other types of defects such as stains, fingerprints, water drops, and foreign materials using either a reference-free or reference-based classification.

REFERENCES

[1] H. S. Wu, J. Barba, and J. Gil, "Iterative thresholding for segmentation of cells from noisy Images," Journal of Microscopy, vol. 197, pp. 296-304, 2000.

[2] C. Zikuan, T. Yang, C. Xin, and C. Griffis, "Wavelet-based adaptive thresholding method for image segmentation," Optical Engineering, vol. 40, no. 5, pp. 868-74, 2001.

[3] H. Shah-Hosseini and R. Safabakhsh, "Automatic multilevel thresholding for image segmentation by the growing time adaptive self-organizing map," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 24, no. 10, pp. 1388-93, 2002.

[4] O. J. Tobias and R. Seara, "Image segmentation by histogram thresholding using fuzzy sets," IEEE Transactions on Image Processing, vol. 11, no. 12, pp. 1457-65, 2002.

[5] X.-Q. Chen, Y.-H. Hu, and Y.-R. Huang, "Image thresholding segmentation based on two-dimensional MCC," Journal of Infrared and Millimeter Waves, vol. 24, no. 5, pp. 397-400, 2005.

[6] A. Z. Arifin and A. Asano, "Image segmentation by histogram thresholding using hierarchical cluster analysis," Pattern Recognition Letters, vol. 27, no. 13, pp. 1515-21, 2006.

[7] J. Canny, "A computational approach to edge detection," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. PAMI-8, no. 6, pp. 679-698, 1986.

[8] J. S. Lim, "Two-Dimensional signal and image processing," Englewood Cliffs, pp. 478-488., NJ: Prentice Hall, 1990.

[9] J. R. Parker, "Algorithms for image processing and computer vision," pp. 23-29, New York: John Wiley & Sons, Inc., 1997.

[10] Rafael C. Gonzalez and R. E. Woods, Digital Image Processing: Prentice Hall, 2003.

[11] S. Tabbone, "Detecting junctions using properties of the Laplacian of Gaussian detector," Proceedings of the 12th IAPR International Conference on Pattern Recognition, pp. 52-6, 2002.

[12] S. A. Tabbone, L. Alonso, and D. Ziou, "Behavior of the Laplacian of Gaussian extrema," Journal of Mathematical Imaging and Vision, vol. 23, no. 1, pp. 107-128, 2005.

[13] D. L. Yang, J. H. Chang, M. C. Hong, and J. S. Liu, "An efficient K-means-based clustering algorithm," Proceedings of the 1st Asia-Pacific Conference on Intelligent Agent Technology Systems, Methodologies, and Tools. pp. 269-73, 1999.

[14] T. Kanungo, D. M. Mount, N. S. Netanyahu, C. D. Piatko, R. Silverman, and A. Y. Wu, “An efficient K-means clustering algorithm: analysis and implementation,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 24, no. 7, pp. 881-92, 2002.

[15] S. S. Shahapurkar and M. K. Sundareshan, “A novel K-means hierarchical clustering algorithm for efficient information extraction from large data sets,” International Conference on Information and Knowledge Engineering - IKE, pp. 390-6, 2003.

[16] J. Chen, R. K. H. Ching, and Y. Lin, “An extended study of the K-means algorithm for data clustering and its applications,” Journal of the Operational Research Society, vol. 55, no. 9, pp. 976-87, 2004.

[17] H. Ming-Chuan, W. Jungpin, C. Jin-Hua, and Y. Don-Lin, “An efficient K-means clustering algorithm using simple partitioning,” Journal of Information Science and Engineering, vol. 21, no. 6, pp. 1157-77, 2005.

[18] G. A. F. Seber, Multivariate Observations, Hoboken, NJ: John Wiley & Sons, Inc, 1984.

[19] T. H. Le, S. W. Jung, K. S. Choi, and S. J. Ko, “Image segmentation based on modified graph-cut algorithm,” Electronics Letters, vol. 46, no. 16, pp. 1121-3, 2010.

[20] W.-l. Yang and L. Guo, “Image segmentation method based on watersheds and graph theory,” Computer Engineering and Applications, vol. 43, no. 7, pp. 28-44, 2007.

[21] K. Fukuda, T. Takiguchi, and Y. Ariki, “Graph cuts segmentation by using local texture features of multiresolution analysis,” IEICE Transactions on Information and Systems, vol. E92-D, no. 7, pp. 1453-61, 2009.

[22] Z. Junhua, W. Yuanyuan, and S. Xinling, “An improved graph cut segmentation method for cervical lymph nodes on sonograms and its relationship with node's shape assessment,” Computerized Medical Imaging and Graphics, vol. 33, no. 8, pp. 602-7, 2009.

[23] K. T. Park, J. H. Lee, and Y. S. Moon, “Unsupervised foreground segmentation using background elimination and graph cut techniques,” Electronics Letters, vol. 45, no. 20, pp. 1025-7, 2009.

[24] C. In Gook and P. Kyu Ho, “Range image segmentation using multiple Markov random fields,” IEICE Transactions on Information and Systems, vol. E77-D, no. 3, pp. 306-16, 1994.

[25] K. Held, E. R. Kops, B. J. Krause, W. M. Wells, III, R. Kikinis, and H. W. Muller-Gartner, “Markov random field segmentation of brain MR images,” IEEE Transactions on Medical Imaging, vol. 16, no. 6, pp. 878-86, 1997.

[26] G. Poggi and R. P. Ragozini, “Image segmentation by tree-structured Markov random fields,” IEEE Signal Processing Letters, vol. 6, no. 7, pp. 155-7, 1999.

[27] E. Cesmeli and D. Wang, “Texture segmentation using Gaussian-Markov random fields and neural oscillator networks,” IEEE Transactions on Neural Networks, vol. 12, no. 2, pp. 394-404, 2001.

[28] Y. Xiangyu and L. Jun, “Unsupervised texture segmentation with one-step mean shift and boundary Markov random fields,” Pattern Recognition Letters, vol. 22, no. 10, pp. 1073-81, 2001.

[29] D. E. Melas and S. P. Wilson, “Double Markov random fields and Bayesian image segmentation,” IEEE Transactions on Signal Processing, vol. 50, no. 2, pp. 357-65, 2002.

[30] D. Huawu and D. A. Clausi, “Unsupervised image segmentation using a simple MRF model with a new implementation scheme,” Proceedings of the 17th International Conference on Pattern Recognition. pp. 691-4, 2004.

[31] Y. Boykov and V. Kolmogorov, “Computing geodesics and minimal surfaces via graph cuts,” Proceedings of the IEEE International Conference on Computer Vision. pp. 26-33, 2003.

[32] D. Mumford and J. Shah, “Boundary detection by minimizing functionals. I,” Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR). pp. 22-6, 1985.

[33] M. Rousson, T. Brox, and R. Deriche, “Active unsupervised texture segmentation on a diffusion based feature space,” Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 699-704, 2003.

[34] C. Lorenz, I. C. Carlsen, T. M. Buzug, C. Fassnacht, and J. Weese, “A multi-scale line filter with automatic scale selection based on the Hessian matrix for medical image segmentation,” First International Conference on Scale-Space Theory in Computer Vision, pp. 152-63, 1997.

[35] S. M. Pizer, P. T. Fletcher, S. Joshi, A. Thall, J. Z. Chen, Y. Fridman, D. S. Fritsch, A. G. Gash, J. M. Glotzer, M. R. Jiroutek, C. Lu, K. E. Muller, G. Tracton, P. Yushkevich, and E. L. Chaney, “Deformable M-Reps for 3D medical image segmentation,” International Journal of Computer Vision, vol. 55, no. 2-3, pp. 85-106, 2003.

[36] L. Hua, A. Elmoataz, J. Fadili, S. Ruan, and B. Romaniuk, “3D medical image segmentation approach based on multi-label front propagation,” IEEE International Conference on Image Processing (ICIP), pp. 2925-8, 2004.

[37] V. Bevilacqua, G. Mastronanii, and M. Marinelli, “A neural network approach to medical image segmentation and three-dimensional reconstruction,” International Conference on Intelligent Computing (ICIC), pp. 22-31, 2006.

[38] Z. Yan, B. J. Matuszewski, L. K. Shark, and C. J. Moore, “Medical image segmentation using new hybrid level-set method,” IEEE International Conference on BioMedical Visualization: Information Visualization in Medical and Biomedical Informatics (MEDIVIS). pp. 71-6, 2008.

[39] Z. Zhou, “Medical image segmentation based on a 3D-MRF,” International Conference on Biomedical Engineering and Informatics (BMEI). pp. 137-41, 2008.

[40] O. Bernard and D. Friboulet, “Fast medical image segmentation through an approximation of narrow-band B-spline level-set and multiresolution,” IEEE International Symposium on Biomedical Imaging: From Nano to Macro (ISBI). pp. 45-8, 2009.

[41] T. Heimann and H. P. Meinzer, “Statistical shape models for 3D medical image segmentation: a review,” Medical Image Analysis, vol. 13, no. 4, pp. 543-63, 2009.

[42] Y. C. J. Hu, M. Grossberg, and G. Mageras, “Volumetric medical image segmentation with probabilistic conditional random fields framework,” International Conference on Image Processing, Computer Vision, Pattern Recognition (IPCV), pp. 48-53, 2009.

[43] A. Kehoe, G. A. Parker, and A. LeBlanc, “Image processing for automatic defect detection in industrial radiographic images,” IEEE International Conference on Image Processing and its Applications, pp. 202-6, 2002.

[44] P. Geveaux, J. Miteran, S. Kohler, F. Truchetet, and E. Renier, “A lighting characterisation by a reliable method. Application to defect detection by artificial vision in industrial field,” Proceedings of the 24th Annual Conference of the IEEE Industrial Electronics Society (IECON). pp. 1242-5, 1998.

[45] D. R. Patek, J. S. Goddard, T. Karnowski, D. Lamond, and T. A. Hawkins, “Rule-based inspection of printed green ceramic tape,” Proc. SPIE - Int. Soc. Opt. Eng, vol. 3306, pp. 88-99, 1998.

[46] H. Sari-Sarraf and J. S. Goddard, “Vision system for on-loom fabric inspection,” IEEE Annual Textile, Fiber and Film Industry Technical Conference, pp. 8-1, 1998.

[47] H. Sari-Sarraf and J. S. Goddard, Jr., “Vision system for on-loom fabric inspection,” IEEE Transactions on Industry Applications, vol. 35, no. 6, pp. 1252-9, 1999.

[48] P. Bourgeat, F. Meriaudeau, P. Gorria, and K. W. Tobin, “Content-based segmentation of patterned wafer for automatic threshold determination,” Proc. SPIE - Int. Soc. Opt. Eng., vol. 5011, pp. 183-9, 2003.

[49] T. M. Wolbank and P. E. Macheiner, “Detection of airgap asymmetry in induction machines with different pole pair number using the current step response,” Proceedings of the IEEE International Electric Machines and Drives Conference (IEMDC). pp. 1177-1182, 2007.

[50] A. F. Said, B. L. Bennett, L. J. Karam, and J. Pettinato, “A versatile automated defect detection and classification system for assembly, test semi-conductor manufacturing,” in International SEMATECH Manufacturing Initiative conference (ISMI), Austin, texas, USA, 2009.

[51] A. F. Said, B. L. Bennett, L. J. Karam, and J. Pettanato, “Automated detection and classification of non-wet solder joints,” accepted and to appear in the IEEE Transactions on Automation Science and Engineering, 2010.

[52] A. F. Said, B. L. Bennett, L. J. Karam, and J. Pettinato, “Robust automated void detection in solder balls and joints,” IPC APEX EXPO™, Las Vegas, Nevada, USA, 2010.

[53] A. F. Said, B. L. Bennett, L. J. Karam, and J. Pettinato, “Robust automatic void detection in solder balls,” IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), pp. 1650-3, Dallas, Texas, USA, 2010.

[54] A. F. Said, B. L. Bennett, F. Toth, L. J. Karam, and J. Pettinato, “Non-Wet solder joint detection in processor sockets and BGA assemblies,” IEEE Electronic Components and Technology Conference (ECTC), pp. 1147-1153, Las Vegas, NV, USA, 2010.

[55] I. C. Chang and H. Chung-Lin, “The model-based human body motion analysis system,” Image and Vision Computing, vol. 18, no. 14, pp. 1067-83, 2000.

[56] Q. Long, W. Yichen, L. Le, and S. Heung-Yeung, “Constrained planar motion analysis by decomposition,” Image and Vision Computing, vol. 22, no. 5, pp. 379-89, 2004.

[57] M. Nicolescu and G. Medioni, “A voting-based computational framework for visual motion analysis and interpretation,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 27, no. 5, pp. 739-52, 2005.

[58] R. Poppe, “Vision-based human motion analysis: an overview,” Computer Vision and Image Understanding, vol. 108, no. 1-2, pp. 4-18, 2007.

[59] B. Lovern, R. Evans, L. Jones, S. Evans, L. Stroud, and C. Holt, “Functional assessment of the shoulder complex using 3D motion analysis techniques,” Journal of Biomechanics, vol. 41, pp. S415, 2008.

[60] M. N. Nyan, F. E. H. Tay, and M. Z. E. Mah, “Application of motion analysis system in pre-impact fall detection,” Journal of Biomechanics, vol. 41, no. 10, pp. 2297-304, 2008.

[61] F. Bumbaca and K. C. Smith, “Design and implementation of a colour vision model for computer vision applications,” Computer Vision, Graphics, and Image Processing, vol. 39, no. 2, pp. 226-45, 1987.

[62] A. Rosenfeld, “Image analysis and computer vision: 1987,” Computer Vision, Graphics, and Image Processing, vol. 42, no. 2, pp. 234-93, 1988.

[63] A. Rosenfeld, “Image analysis and computer vision: 1988,” Computer Vision, Graphics, and Image Processing, vol. 46, no. 2, pp. 196-264, 1989.

[64] A. Rosenfeld, “Image analysis and computer vision: 1989,” Computer Vision, Graphics, and Image Processing, vol. 50, no. 2, pp. 188-240, 1990.

[65] D. G. Joshi, Y. V. Rao, S. Kar, V. Kumar, and R. Kumar, “Computer-vision-based approach to personal identification using finger crease pattern,” Pattern Recognition, vol. 31, no. 1, pp. 15-22, 1998.

[66] P. K. Saha, “Tensor scale: a local morphometric parameter with applications to computer vision and image processing,” Computer Vision and Image Understanding, vol. 99, no. 3, pp. 384-413, 2005.

[67] P. Ram Bhuwan, T. Juming, L. Frank, and G. Mikhaylenko, “A computer vision method to locate cold spots in foods in microwave sterilization processes,” Pattern Recognition, vol. 40, no. 12, pp. 3667-76, 2007.

[68] Z. Junmei and N. Ruola, “Image denoising based on wavelets and multifractals for singularity detection,” IEEE Transactions on Image Processing, vol. 14, no. 10, pp. 1435-47, 2005.

[69] Z. Ming and B. K. Gunturk, “Multiresolution bilateral filtering for image denoising,” IEEE Transactions on Image Processing, vol. 17, no. 12, pp. 2324-33, 2008.

[70] A. Barbu, “Training an active random field for real-time image denoising,” IEEE Transactions on Image Processing, vol. 18, no. 11, pp. 2451-62, 2009.

[71] I. Firoiu, C. Nafornita, J. M. Boucher, and A. Isar, “Image denoising using a new implementation of the hyperanalytic wavelet transform,” IEEE Transactions on Instrumentation and Measurement, vol. 58, no. 8, pp. 2410-16, 2009.

[72] A. Bugeau, M. Bertalmio, V. Caselles, and G. Sapiro, “A comprehensive framework for image inpainting,” IEEE Transactions on Image Processing, vol. 19, no. 10, pp. 2634-45, 2010.

[73] M. Bertalmio, L. Vese, G. Sapiro, and S. Osher, “Simultaneous structure and texture image inpainting,” IEEE Transactions on Image Processing, vol. 12, no. 8, pp. 882-9, 2003.

[74] S. D. Rane, G. Sapiro, and M. Bertalmio, “Structure and texture filling-in of missing image blocks in wireless transmission and compression applications,” IEEE Transactions on Image Processing, vol. 12, no. 3, pp. 296-303, 2003.

[75] C. Yen-Liang, H. Ching-Tang, and H. Chih-Hsu, “Progressive image inpainting based on wavelet transform,” IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences, vol. E88-A, no. 10, pp. 2826-34, 2005.

[76] Z. Yue-ting, W. Yu-shun, T. Shih, and N. Tang, "Patch-guided facial image inpainting by shape propagation," Journal of Zhejiang University Science A, vol. 10, no. 2, pp. 232-8, 2009.

[77] A. F. Said and L. J. Karam, "Multi-Region texture image segmentation based on constrained level-set evolution functions," in the Proceedings of the IEEE 13th Digital Signal Processing Workshop, pp. 664-668, 2009.

[78] A. F. Said and L. J. Karam, "Cell migration analysis using a statistical level-set segmentation on a wavelet-based structure tensor feature space," in IEEE International Symposium on Signal Processing and Information Technology, pp. 473-478, 2007.

[79] A. F. Said, L. J. Karam, M. E. Berens, Z. Lacroix, and R. A. Renaut, "Migration and proliferation analysis for bladder cancer cells," in IEEE International Symposium on Biomedical Imaging: Macro to Nano, Arlington, VA, USA, pp. 320-323, 2007.

[80] A. F. Said and L. J. Karam, "Automatic cell migration and proliferation analysis," AZTE Case#M8-006 and US Patent filed, April 2009.

[81] A. F. Said, L. J. Karam, and S. R. Gehler, "Use of Muscale CMA*cfz* automated image analysis software to accurately quantitate cell migration," in the 16th Annual Conference & Exhibition in Advancing the Science of Drug Discovery, Phoenix, AZ, USA, 2010.

[82] www.muscale.com.

[83] http://www.faqs.org/patents/app/20100080439.

[84] G. Song and T. D. Bui, "Image segmentation and selective smoothing by using Mumford-Shah model," IEEE Transactions on Image Processing, vol. 14, no. 10, pp. 1537-49, 2005.

[85] P. Zhilkin and M. Alexander, "Multi-phase image segmentation using level sets," in Proc. SPIE, pp. 69143-1, Vol. 6914, San Diego, CA, USA, 2008.

[86] M. Jeon, M. Alexander, N. Pizzi, and W. Pedrycz, "Unsupervised hierarchical multi-scale image segmentation level set, wavelet and additive splitting operator," in Proceeding of NAFIPS, pp. 664-668, Vol.2, Alberta, Canada, 2004.

[87] M. Jeon, M. Alexander, W. Pedrycz, and N. Pizzi, "Unsupervised hierarchical image segmentation with level set and additive operator splitting," Pattern Recognition Letters, vol. 26, no. 10, pp. 1461-9, 2005.

[88] T. Brox and J. Weickert, "Level set based image segmentation with multiple regions," in 26th DAGM Symposium on Pattern Recognition, Tubingen, Germany, pp. 415-23, 2004.

[89] T. Brox, and J. Weickert, "Level set segmentation with multiple regions," IEEE Transactions on Image Processing, vol. 15, no. 10, pp. 3213-3218, 2006.

[90] Z. Hong-Kai, T. Chan, B. Merriman, and S. Osher, “A variational level set approach to multiphase motion,” J. Comput. Phys., vol. 127, no. 1, pp. 179-195, 1996.

[91] P. Nikos and D. Rachid, “Coupled geodesic active regions for image segmentation: a level set approach,” in Proceedings of the 6th European Conference on Computer Vision-Part II, ECCV, pp. 224-240, Dublin, IRLANDE, 2000.

[92] S. Christophe, F. Laure Blanc, raud, A. Gilles, and Z. Josiane, “A level set model for image classification,” Int. J. Comput. Vision, vol. 40, no. 3, pp. 187-197, 2000.

[93] L. A. Vese, and T. F. Chan, “A multiphase level set framework for image segmentation using the Mumford and Shah model,” International Journal of Computer Vision, vol. 50, no. 3, pp. 271-93, 2002.

[94] B. Sandberg, S. H. Kang, and T. F. Chan, “Unsupervised feature balancing multi-phase segmentation,”, UCLA CAM report, 08-02, Tech. Rep., 2008.

[95] H. Lin and S. Osher, “Solving the Chan-Vese model by a multiphase level set algorithm based on the topological derivative,” in Proc. of Scale Space and Variational Methods in Computer Vision, SSVM, pp. 777-88, Ischia, Italy, 2007.

[96] I. Yanovsky, P. M. Thompson, S. Osher, L. Vese, and A. D. Leow, “Multiphase segmentation of deformation using logarithmic priors,” in IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 3359-64, Minneapolis, MN, USA, 2007.

[97] J. Yoon Mo, K. Sung Ha, and S. Jianhong, “Multiphase image segmentation via Modica-Mortola phase transition,” Journal on Applied Mathematics, SIAM '07, vol. 67, no. 5, pp. 1213-1232, 2007.

[98] G. Chung and L. A. Vese, “Energy minimization based segmentation and denoising using a multilayer level set approach,” Proceedings of the 5th International Workshop on Energy Minimization Methods in Computer Vision and Pattern Recognition (EMMCVPR), pp. 439-55, 2005.

[99] A. R. Mansouri, A. Mitiche, and C. Vazquez, “Image partioning by level set multiregion competition,” in IEEE International Conference on Image Processing (ICIP), pp. 2721-4, 2004.

[100] T. F. Chan, and L. A. Vese, “Active contours without edges,” IEEE Transactions on Image Processing, vol. 10, no. 2, pp. 266-77, 2001.

[101] D. M. Strong, J. Aujol, and T. F. Chan, “Scale recognition, regularization parameter selection, and Meyer's G norm in total variation regularization,” Journal of Multiscale Modeling & Simulation, vol. 5, no. 1, pp. 273-303, 2006.

[102] D. Mumford and J. Shah, “Boundary detection by minimizing functionals. I,” in IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), San Francisco, CA, USA, pp. 22-6, 1985.

[103] A. R. Forsyth, “Calculus of Variations,” pp. 17-20 and 29, New York: Dover Publications, 1960.

[104] R. Xia, W. Liu, Y. Wang, and X. Wu, “Fast initialization of level set method and an improvement to Chan-Vese model,” Proceedings of the IEEE Fourth International Conference on Computer and Information Technology (CIT) pp. 18-23, 2004.

[105] J. Jianguo, G. Yanrong, Z. Shu, and L. Hong, “Segmentation of knee joints based on improved multiphase Chan-Vese model,” IEEE 2nd International Conference on Bioinformatics and Biomedical Engineering (ICBBE), pp. 2418-22, 2008.

[106] B. Merriman, J. K. Bence, and S. J. Osher, “Motion of multiple junctions: a level set approach,” Journal of Computational Physics, vol. 112, no. 2, pp. 334-63, 1994.

[107] Z. Song Chun and A. Yuille, “Region competition: unifying snakes, region growing, and Bayes/MDL for multiband image segmentation,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 18, no. 9, pp. 884-900, 1996.

[108] R. Kimmel, “Fast edge integration,” Geometric Level Set Methods in Imaging, Vision, and Graphics, pp. 59-77: Springer New York, 2003.

[109] R. Malladi, J. A. Sethian, and B. C. Vemuri, “Shape modeling with front propagation: a level set approach,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 17, no. 2, pp. 158-75, 1995.

[110] S. Chen, N. A. Sochen, and Y. Y. Zeevi, “Integrated active contours for texture segmentation,” IEEE Transactions on Image Processing, vol. 15, no. 6, pp. 1633-1646, 2006.

[111] G. Gerig, O. Kubler, R. Kikinis, and F. A. Jolesz, “Nonlinear anisotropic filtering of MRI data,” IEEE Transactions on Medical Imaging, vol. 11, no. 2, pp. 221-32, 1992.

[112] V. I. Tsurkov, “An analytical model of edge protection under noise suppression by anisotropic diffusion,” Journal of Computer and Systems Sciences International, vol. 39, no. 3, pp. 437-440, 2000.

[113] D. Cremers, M. Rousson, and R. Deriche, “A review of statistical approaches to level set segmentation integrating color, texture, motion and shape,” International Journal of Computer Vision, vol. 72, no. 2, pp. 195-215, 2007.

[114] J. Bigun, G. H. Granlund, and J. Wiklund, “Multidimensional orientation estimation with applications to texture analysis and optical flow,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 13, no. 8, pp. 775-790, 1991.

[115] M. Rousson, T. Brox, and R. Deriche, “Active unsupervised texture segmentation on a diffusion based feature space,” in Proceedings of the IEEE Computer Vision and Pattern Recognition (CVPR), Madison, WI, USA, pp. 699-704, vol.2, 2003.

[116] J. Weickert, and T. Brox, "Diffusion and regularization of vector and matrix valued images, " Technical report, Saarland University, Saarland, Germany, 2002.

[117] M. Rousson and R. Deriche, "A variational framework for active and adaptative segmentation of vector valued images," in Proceedings of the Workshop on Motion and Video Computing, pp. 56-61, Orlando, FL, USA, 2002.

[118] C. J. Fong, D. M. Sutkowski, J. M. Kozlowski, and C. Lee, "Utilization of the Boyden chamber to further characterize in vitro migration and invasion of benign and malignant human prostatic epithelial cells," Invasion Metastasis, vol. 12, pp. 264-274, 1992.

[119] A. Wells, "Tumor invasion: role of growth factor-induced cell motility.," Adv Cancer Res. , vol. 78, pp. 31-101, 2000.

[120] M. Bittner, P. Meltzer, Y. Chen, Y. Jiang, E. Seftor, M. Hendrix, M. Radmacher, R. Simon, Z. Yakhini, A. Ben-Dor, N. Sampas, E. Dougherty, E. Wang, F. Marincola, C. Gooden, J. Lueders, A. Glatfelter, P. Pollock, J. Carpten, E. Gillanders, D. Leja, K. Dietrich, C. Beaudry, M. Berens, D. Alberts, V. Sondak, N. Hayward, and J. Trent, "Molecular classification of cutaneous malignant melanoma by gene expression profiling," Nature, vol. 406, pp. 536-540, 2000.

[121] Michael E. Berens, Monique D. Rief, Melinda A. Loo, and Alf Giese, " The role of extracellular matrix in human astrocytoma migration and proliferation studied in a microliter scale assay," Clinical and Experimental Metastasis, vol. 12, pp. 405-415, 1994.

[122] S. P. Langdon, Cancer Cell Culture: Methods and Protocols, pp. 219-224, Totowa, NJ: Humana Press, 2004.

[123] N. N. Kachouie and P. Fieguth, "A narrow-band level-set method with dynamic velocity for neural stem cell cluster segmentation," Image Analysis and Recognition (ICIAR), vol. 3656, pp. 1006-13, 2005.

[124] O. Debeir, P. Van Ham, R. Kiss, and C. Decaestecker, "Tracking of migrating cells under phase-contrast video microscopy with combined mean-shift processes," IEEE Transactions on Medical Imaging, vol. 24, no. 6, pp. 697-711, 2005.

[125] E. Espinoza, G. Martinez, J. G. Frerichs, and T. Scheper, "Cell cluster segmentation based on global and local thresholding for in-situ microscopy," 3rd IEEE International Symposium on Biomedical Imaging: Macro to Nano, pp. 542-5, 2006.

[126] F. Bunyak, K. Palaniappan, S. K. Nath, T. L. Baskin, and D. Gang, "Quantitative cell motility for in vitro wound healing using level set-based active contour tracking," 3rd IEEE International Symposium on Biomedical Imaging: Macro to Nano, pp. 1040-3, 2006.

[127] K. Li, E. D. Miller, L. E. Weiss, P. G. Campbell, and T. Kanade, "Online tracking of migrating and proliferating cells imaged with phase-contrast microscopy,"

Proceedings of the Computer Vision and Pattern Recognition Workshop (CVPRW), pp. 65-65, 2006.

[128] Y. Fuxing, M. A. Mackey, F. Ianzini, G. Gallardo, and M. Sonka, "Cell segmentation, tracking, and mitosis detection using temporal context," Proceedings of Medical Image Computing and Computer-Assisted Intervention-MICCAI, Part I (Lecture Notes in Computer Science vol. 3749). pp. 302-9, 2005.

[129] N. N. Kachouie, L. J. Lee, and P. Fieguth, "A probabilistic living cell segmentation model," Proceedings of the IEEE International Conference on Image Processing, ICIP. pp. 1137-1140, 2005.

[130] R. B. Nawrocki, M. Polette, C. Gilles, C. Clavel, K. Strumane, M. Matos, J. M. Zahm, R. F.Van, N. Bonnet, and P. Birembaut, "Quantitative cell dispersion analysis: New test to measure tumor cell aggressiveness," International Journal of Cancer, vol. 93, no. 5, pp. 644-652, 2001.

[131] P. Brodatz, Textures: a Photographic Album for Artists and Designers, Dover, New York, 1966.

[132] M. Rousson, T. Brox, and R. Deriche, "Active unsupervised texture segmentation on a diffusion based feature space," IEEE Conference on Computer Vision and Pattern Recognition. pp. 699-704, 2003.

[133] P. Perona and J. Malik, "Scale-space and edge detection using anisotropic diffusion," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 12, no. 7, pp. 629-39, 1990.

[134] J. Starck, F. Murtagh, and P. Bijaoui, "Multiresolution support applied to image filtering and restoration," Graphical Models and Image Processing, vol. 57, no. 5, pp. 420-31, 1995.

[135] M. Rousson and R. Deriche, "A variational framework for active and adaptive segmentation of vector valued images," Proceedings of the Workshop on Motion and Video Computing, pp. 56-61, 2002.

[136] S. Mallat and W. L. Hwang, "Singularity detection and processing with wavelets," IEEE Transactions on Information Theory, vol. 38, no. 2, pp. 617-643, 1992.

[137] C. R. Jung, "Unsupervised multiscale segmentation of color images," Pattern Recognition Letters, vol. 28, no. 4, pp. 523-33, 2007.

[138] J. C. Olivo-Marin, "Extraction of spots in biological images using multiscale products," Pattern Recognition, vol. 35, no. 9, pp. 1989-96, 2002.

[139] H. Jeong, T. Kim, H. Hwang, H. Choi, H. Park, and H. Choi, "Comparison of thresholding methods for breast tumor cell segmentation," Proceedings of the 7th International Workshop on Enterprise Networking and Computing in Healthcare Industry(HEALTHCOM), pp. 392-5, 2005.

[140] M. Berens and W. McDonough, "Performance of Muscale CEA Automated Analysis Software for Cell Migration Quantification," TGen technical report, www.muscale.com.

[141] M. E. Berens and C. Beaudry, "Radial monolayer cell migration assay," Methods in Molecular Medicine in Cancer Cell Culture, vol. 88, pp. 219-224, 2004.

[142] www.creative-sci.com.

[143] S. Rooks, B. Benhabib, and K. C. Smith, "Development of an inspection process for ball-grid-array technology using scanned-beam X-ray laminography," Proceedings of the Technical National Electronic Packaging and Production Conference, NEPCON West'94, pp. 277-89, 1994.

[144] T. Sumimoto, T. Maruyamay, Y. Azuma, S. Goto, M. Mondo, N. Furukawa, and S. Okada, "Detection of defects at BGA solder joints by using X-ray imaging," IEEE International Conference on Industrial Technology. pp. 238-41, 2002.

[145] T. Sumimoto, T. Maruyama, Y. Azuma, S. Goto, M. Mondou, N. Furukawa, and S. Okada, "Shape measurement of BGA for analysis of defects by X-ray imaging," Proc. SPIE - Int. Soc. Opt. Eng., vol. 5253, pp. 361-5, 2003.

[146] T. Sumimoto, T. Maruyama, Y. Azuma, S. Goto, M. Mondou, N. Furukawa, and S. Okada, "Development of image analysis for detection of defects of BGA by using X-ray images," Proceedings of the 20th IEEE Instrumentation Technology Conference. pp. 1131-6, 2003.

[147] M. Ji-Quan, K. Fan-Hui, M. Pei-Jun, and S. Xiao-Hong, "Detection of defects at BGA solder joints by using X-ray imaging," International Conference on Machine Learning and Cybernetics. pp. 5139-43, 2005.

[148] K. W. Ko, Y. J. Roh, H. S. Cho, and H. C. Kim, "A neural network approach to the inspection of ball grid array solder joints on printed circuit boards," Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks (IJCNN), pp. 233-8, 2000.

[149] V. Sankaran, A. R. Kalukin, and R. P. Kraft, "Improvements to X-ray laminography for automated inspection of solder joints," IEEE Transactions on Components, Packaging & Manufacturing Technology, Part C (Manufacturing), vol. 21, pp. 148-54, 1998.

[150] Y. J. Roh, K. W. Ko, H. S. Cho, H. C. Kim, H. N. Joo, and S. K. Kim, "Inspection of ball grid array (BGA) solder joints using X-ray cross-sectional images," Proc. SPIE - Int. Soc. Opt. Eng., vol. 3836, pp. 168-78, 1999.

[151] Z. P. Xiong, H. P. Sze, and K. H. Chua, "Bump non-wet issue in large-die flip chip package with eutectic Sn/Pb solder bump and SOP substrate pad," Proceedings of the IEEE 6th Electronics Packaging Technology Conference (EPTC), pp. 438-443, 2004.

[152] C. C. Ser, M. T. Yeow, C. C. Tai, L. Samuel, Y. H. Wai, and K. C. Chek, "Assembly challenges of high density large fine pitch lead-free flip chip package," Proceedings of the Electronic Components and Technology Conference. pp. 10-15, 2006.

[153] L. Xingsheng and L. Guo-Quan, "Effects of solder joint shape and height on thermal fatigue lifetime," IEEE Transactions on Components and Packaging Technologies, vol. 26, no. 2, pp. 455-65, 2003.

[154] A. Teramoto, M. Yamada, T. Murakoshi, M. Tsuzaka, and H. Fujita, "High speed oblique CT system for solder bump inspection," 33rd Annual Conference of the IEEE Industrial Electronics (IECON), pp. 2689-93, 2007.

[155] C. Neubauer, "Intelligent X-ray inspection for quality control of solder joints," IEEE Transactions on Components, Packaging and Manufacturing Technology, Part C (Manufacturing), vol. 20, no. 2, pp. 111-20, 1997.

[156] A. R. Kalukin, V. Sankaran, B. Chartrand, D. L. Millard, R. P. Kraft, and M. J. Embrechts, "An improved method for inspection of solder joints using X-ray laminography and X-ray microtomography," IEEE/CPMT International Electronics Manufacturing Technology Symposium (IEMT),pp. 438-45, 1996.

[157] G. Acciani, G. Brunetti, and G. Fornarelli, "Application of neural networks in optical inspection and classification of solder joints in surface mount technology," IEEE Transactions on Industrial Informatics, vol. 2, no. 3, pp. 200-9, 2006.

[158] R. J. Coyle, A. Holliday, P. P. Solan, C. Yao, H. A. Cyker, J. C. Manock, R. Bond, R. E. Stenerson, R. G. Furrow, M. V. Occhipinti, and S. A. Gahr, "Solder joint attachment reliability and assembly quality of a molded ball grid array socket," Proceedings of the 51st Electronic Components and Technology Conference, pp. 1219-26, 2001.

[159] A. R. Kalukin and V. Sankaran, "Three-dimensional visualization of multilayered assemblies using X-ray laminography," IEEE Trans. Compon. Packag. Manuf. Technol. vol. 20, pp.361-366, 2002.

[160] M. Chen, C. Tai, Y. Huang, and I. Wu, "Failure analysis of BGA package by a TDR approach," Proceedings of the 4th International Symposium on Electronic Materials and Packaging, pp. 112-16, 2003.

[161] T. D. Moore, D. Vanderstraeten, and P. M. Forssell, "Three-dimensional X-ray laminography as a tool for detection and characterization of BGA package defects," IEEE Transactions on Components and Packaging Technologies, vol. 25, no. 2, pp. 224-9, 2002.

[162] T. Y. Ong, Z. Samad, and M. M. Ratnam, "Solder joint inspection with multi-angle imaging and an artificial neural network," International Journal of Advanced Manufacturing Technology, vol. 38, no. 5-6, pp. 455-462, 2008.

[163] J. S. Lim, "Two-Dimensional signal and image processing," pp. 469-476, Englewood Cliffs, NJ: Prentice Hall, 1990.

[164] B. van den Rein, and B. van Richard, “Methods for fast morphological image transforms using bitmapped binary images,” Graphical Models and Image Processing (CVGIP), vol. 54, no. 3, pp. 252-258, 1992.

[165] R. M. Haralick, and L. G. Shapiro, "Computer and Robot Vision," Addison-Wesley, 1992.

[166] D. J. Dwyer, M. I. Smith, J. L. Dale, and J. P. Heather, “Real-time implementation of image alignment and fusion,” Proc. SPIE - Int. Soc. Opt. Eng., vol. 5612, pp. 85-93, 2004.

[167] W. Duan, F. Kuester, J. L. Gaudiot, and O. Hammami, “Automatic object and image alignment using Fourier descriptors,” Image and Vision Computing, vol. 26, no. 9, pp. 1196-206, 2008.

[168] R. Aspandiar, “Voids in solder joints,” Surface Mount Technology Association Journal, vol. 19, no. 4, pp. 28-36, 2006.

[169] R. Prasad, “BGA voids and their sources in SMT assemblies,” Surface Mount Technology Magazine, SMT, 2003.

[170] Q. Yu, T. Shibutani, Y. Kobayashi, and M. Shiratori, “The effect of voids on thermal reliability of BGA lead free solder joint and reliability detecting standard,” Proceedings of the tenth Intersociety Conference on Thermal and Thermomechanical Phenomena in Electronics Systems, pp. 1024-1030, 2006.

[171] V. Tvergaard and C. Niordson, “Non-Local plasticity effect on interaction of different size voids,” Int. J. Plast., vol. 20, pp. 107-120.

[172] http://homepages.inf.ed.ac.uk/rbf/HIPR2/log.htm.

[173] S. S. Gleason, K. W. Tobin, and T. P. Karnowski, “An integrated spatial signature analysis and automatic defect classification system.” Proceedings of the Electrochemical Society Symposium on Diagnostic Techniques for Semiconductor Materials and Devices, pp.204-11, 1997.

[174] S. S. Gleason, K. W. Tobin, T. P. Karnowski, and F. Lakhani, “Rapid yield learning through optical defect and electrical test analysis,” Proceedings of the SPIE - The International Society for Optical Engineering, vol. 3332, pp. 232-42, 1998.

[175] T. P. Karnowski, K. W. Tobin, S. S. Gleason, and F. Lakhani, “Application of spatial signature analysis to electrical test data: validation study,” Proceedings of the SPIE - The International Society for Optical Engineering, vol. 3677, pp. 530-41, 1999.

[176] K. W. Tobin, S. S. Gleason, T. P. Karnowski, S. L. Cohen, and F. Lakhani, “Automatic classification of spatial signatures on semiconductor wafermaps,” Proceedings of the SPIE - The International Society for Optical Engineering. vol. 3050, pp. 434-44, 1997.

[177] K. W. Tobin, T. P. Karnowski, L. F. Arrowood, and F. Lakhani, "Field test results of an automated image retrieval system." Proceedings of the IEEE International Symposium on Semiconductor Manufacturing Conference, pp. 167-174, 2001.

[178] K. W. Tobin, T. P. Karnowski, and F. Lakhani, "The use of historical defect imagery for yield learning." Proceedings of the IEEE International Symposium on Semiconductor Manufacturing Conference, pp. 18-25, 2000.

[179] K. W. Tobin, T. P. Karnowski, and F. Lakhani, "Survey of semiconductor data management systems technology," Proceedings of SPIE - The International Society for Optical Engineering, vol. 3998, pp. 248-257, 2000.

[180] J. Iivarinen and A. Visa, "An adaptive texture and shape based defect classification." Proceedings of the Fourteenth International Conference on Pattern Recognition, vol. 1, pp. 117-22, 1998.

[181] I. Kunttu, L. Lepistö, J. Rauhamaa, and and A. Visa, "Classification method for defect images based on association and clustering," Proceedings of the SPIE - The International Society for Optical Engineering, vol. 5098, pp. 19-27, 2003.

[182] K. Iivari, L. Leena, R. Juhani, and V. Ari, "Multiscale fourier descriptor for shape classification," Proceedings of the 12th International Conference on Image Analysis and Processing, pp. 536-41, 2003.

[183] R. Rautkorpi and J. Iivarinen, "Shape-based co-occurrence matrices for defect classification," Proceedings of the 14th Scandinavian Conference on Image Analysis (SCIA), pp. 588-97, 2005.

[184] T. Iga, T. Tanaka, J. i. Hayashi, and S. Hata, "Visual defects classification system using co-occurrence histogram image." Proceedings of the SICE Annual Conference, pp. 598-603, 2007.

[185] V. Murino, M. Bicego, and I. A. Rossi, "Statistical classification of raw textile defects." Proceedings of the 17th International Conference on Pattern Recognition, vol.4, pp. 311-14, 2004.

[186] Y. Xuezhi, G. Jun, P. Grantham, and Y. Nelson, "Textile defect classification using discriminative wavelet frames," Proceedings of the 2005 International Conference on Information Acquisition, v 2005, pp. 54-58, 2005.

[187] Y. Xuezhi, G. Pang, and N. Yung, "Fabric defect classification using wavelet frames and minimum classification error training," IEEE Industry Applications Conference, vol.1, pp. 290-6, 2002.

[188] X. Yang, G. Pang, and N. Yung, "Robust fabric defect detection and classification using multiple adaptive wavelets," IEE Proceedings-Vision, Image and Signal Processing, vol. 152, no. 6, pp. 715-23, 2005.

[189] F. Hanying, Y. Jun, and R. F. Pease, "Self inspection of integrated circuits pattern defects using support vector machines," Journal of Vacuum Science & Technology B: Microelectronics and Nanometer Structures, vol. 23, no. 6, pp. 3085-3089, 2005.

[190] M. R. Driels and D. J. Nolan, "Automatic defect classification of printed wiring board solder joints," IEEE Transactions on Components, Hybrids, and Manufacturing Technology, vol. 13, no. 2, pp. 331-340, 1990.

[191] L. Breaux and B. Singh, "Automatic defect classification system for patterned semiconductor wafers," IEEE/UCS/SEMI International Symposium on Semiconductor Manufacturing, pp. 68-73, 1995.

[192] M. H. Bennett, J. F. Garvin, J. B. Hightower, D. Coldren, and M. Reddy, "In-line automatic defect classification [wafer fabrication]," Proceedings of the IEEE International Symposium on Semiconductor Manufacturing Conference, pp. 29-30, 1997.

[193] K. Kameyama, Y. Kosugi, T. Okahashi, and M. Izumita, "Automatic defect classification in visual inspection of semiconductors using neural networks," IEICE Transactions on Information and Systems, vol. E81-D, no. 11, pp. 1261-71, 1998.

[194] A. Skumanich, J. Boyle, and D. Brown, "Advanced interconnect process development utilizing wafer inspection with "on-the-fly" automatic defect classification," Proceedings of the IEEE/SEMI Advanced Semiconductor Manufacturing Conference and Workshop (ASMC), pp. 259-64, 1999.

[195] J. G. Miller, L. Lee, M. Pham, D. Pham, M. Khaja, and K. Hennessey, "Implementing a fully automatic macro defect detection and classification system in a high-production semiconductor fab," Journal of Microlithography, Microfabrication, and Microsystems, vol. 2, no. 1, pp. 76-80, 2003.

[196] H. L. Sang, I. K. Hong, I. C. Nam, H. J. Yu, S. C. Ki, and S. J. Chung, "Automatic defect classification using boosting," Proceedings of the Fourth International Conference on Machine Learning and Applications (ICMLA). pp. 357-362, 2005.

[197] A. J. Koivo and C. W. Kim, "Classification of surface defects on wood boards," Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics, pp. 1431-6, 1986.

[198] C. W. Kim and A. J. Koivo, "Hierarchical classification of surface defects on dusty wood boards," Pattern Recognition Letters, vol. 15, no. 7, pp. 713-721, July 1994.

[199] H. Kauppinen and O. Silven, "The effect of illumination variations on color-based wood defect classification," Proceedings of the 13th International Conference on Pattern Recognition, vol. 3, pp. 828-32, 1996.

[200] P. A. Estevez, M. Fernandez, R. J. Alcock, and M. S. Packianather, "Selection of features for the classification of wood board defects," IEE Ninth International Conference on Artificial Neural Networks (ICANN), pp. 347-52, 1999.

[201] D. T. Pham and R. J. Alcock, "Automated visual inspection of wood boards: selection of features for defect classification by a neural network," Journal of Process Mechanical Engineering, vol. 213, no. E4, pp. 231-45, 1999.

[202] M. Hongbo, L. Li, Y. Lei, Z. Mingming, and Q. Dawei, "Detection and classification of wood defects by ANN," IEEE International Conference on Mechatronics and Automation, pp. 2235-40, 2006.

[203] P. Caleb and M. Steuer, "Classification of surface defects on hot rolled steel using adaptive learning methods," Proceedings of the Fourth International Conference on Knowledge-Based Intelligent Engineering Systems and Allied Technologies, pp. 103-8, 2000.

[204] R. O. Falessi, M. P. Pensini, G. Rovigatti, and S. Taraglio, "HIPERCLASS: high performance industrial inspection and defect classification in steel industry," Proceedings of the International Conference and Exhibition on High-Performance Computing and Networking. pp. 21-30, 1997.

[205] L. Kangsheng, Z. Haidong, and D. Dongming, "New approach to classification of surface defects in steel plate based on fuzzy neural networks," Proceedings of the SPIE - The International Society for Optical Engineering, vol. 4929, pp. 447-56, 2003.

[206] F. Pernkopf, "Detection of surface defects on raw steel blocks using Bayesian network classifiers," Journal of Pattern Analysis and Applications, vol. 7, no. 3, pp. 333-42, 2004.

[207] R. Rinn, M. Becker, R. Foehr, and F. Luecking, "Steel mill defect detection and classification at 3000 ft/min using mainstream technology," Proceedings of the SPIE - The International Society for Optical Engineering, vol. 3303, pp. 20-6, 1998.

[208] P. Rizzo, I. Bartoli, A. Marzani, and F. L. di Scalea, "Defect classification in pipes by neural networks using multiple guided ultrasonic wave features extracted after wavelet processing," Transactions of the ASME. Journal of Pressure Vessel Technology, vol. 127, no. 3, pp. 294-303, 2005.

[209] F. Trieber, "On-line automatic defect detection and surface roughness measurement of steel strip," Journal of Iron and Steel Engineer, vol. 66, no. 9, pp. 26-33, 1989.

[210] X. Zhao, K. Lai, and D. Dai, "An improved BP algorithm and its application in classification of surface defects of steel plate," Journal of Iron and Steel Research International, vol. 14, no. 2, pp. 52-5, 2007.

[211] Z. Xu, M. Pietikainen, and T. Ojala, "Defect classification by texture in steel surface inspection." International Conference on Quality Control by Artificial Vision, pp. 179-84, 1997.

[212] H. S. M. Coxeter, Introduction to Geometry, 2nd ed., New York: Wiley, 1969.

[213] L. D. F. Costa and R. M. Cesar Jr, Shape Analysis and Classification: Theory and Practice, 2nd ed.: CRC Press, 2009.

[214] I. L. Dryden and K. V. Mardia, Statistical Shape Analysis, 1st ed.: Wiley, 1998.

[215] D. N. Joanes and C. A. Gill, “Comparing measures of sample skewness and kurtosis,” Journal of the Royal Statistical Society: Series D (The Statistician), vol. 47, no. 1, pp. 183-189, 1998.

[216] A. G. Froedesen, O. Skjeggestad, and H. Tøfte, “Probability and Statistics in Particle Physics,” Bergen Universitetsforlaget, Oslo, 1979.