Defect pattern detection using a new rule-based approach

Shankar, N. G

2006

Shankar, N. G. (2006). Defect pattern detection using a new rule-based approach. Doctoral thesis, Nanyang Technological University, Singapore.

https://hdl.handle.net/10356/6209

https://doi.org/10.32657/10356/6209

Defect Pattern Detection Using A New Rule-Based Approach

N. G. Shankar

School of Mechanical & Aerospace Engineering

A thesis submitted to the Nanyang Technological University
in fulfillment of the requirement for the degree of
Doctor of Philosophy

2006

ACKNOWLEDGMENTS

I take this opportunity to express my deep sense of gratitude to Dr. Z.W. Zhong, for having recruited me into the Ph.D. program and assigning me this wonderful thesis topic. His support, encouragement and above all constant monitoring and guidance kept me involved in the project for the past two and half-year of my degree course. I consider myself very lucky for having worked with such a great personality and mentor.

I am also grateful to Mr. K. Ramesh for his timely helpful feedback and suggestions. Thanks to my parents, wife, and my son for their constant pestering, is helping me get this thesis written. Many thanks to the Euro Technology Pte Ltd: Mr. Stevan Tan and Mr. Tan Say Hong for their support and assistance.

I would also like to thank Dr. Gerald Seet, Robotics Research Centre, for his support to utilize the facilities, data and assistance.

Table of Contents

Page

Abstract

Automated inspection of semiconductor defect data has become increasingly important over the past several years as a means of quickly understanding and controlling contamination sources and process faults, which impact product yield. To address the issue of too much data and too little time, automation technologies in defect detection and review are being developed by universities, laboratories, industry, and semiconductor equipment suppliers.

In this thesis, a new rule-based approach is proposed to segment defect images. Several segmentation techniques already exist but they often focus on the constraints of a specific application and therefore they lack of generality and flexibility. This limits the use of computer vision in all those tasks where the visual data content and the purpose of the defect analysis are not known a priori. Moreover, the limited generality increases the costs for the design of unsupervised defect image analysis systems.

To overcome these limitations, it is proposed to decompose a general defect image segmentation problem in four levels. The lowest levels have a high degree of generality and are inspired by the perceptual mechanisms of human vision. Their main role is to simplify the input data and to extract the perceptually meaningful information. The proposed process is sufficiently general to be applied to a wide class of applications and input data without the need of human supervision. In order to achieve a higher degree of autonomy, generality and a more flexible solution, the proposed defect image segmentation approach adopts a hierarchical perspective. The visual information in a frame is viewed as being composed of patterns, referred to as objects, which are built from simpler sub-patterns, referred to as regions, which are themselves built from yet simpler primitives corresponding to the pixels in the defect image.

The new rule-based approach to defect pattern segmentation has been evaluated on a number of different input data and applications. The obtained results have confirmed the validity of the approach, in particular its accuracy in defining the segmentation masks and its high degree of automatism.

A new rule-based approach to image segmentation is developed and it achieves the final results through a two-step hierarchical strategy. First, a fully automatic and general-purpose simplification procedure takes place, where the redundant information is simplified, while the information of interest is highlighted. Second, an application dependent process is defined on the simplification results. It integrates all the information and assumptions available for the specific application. It uses accurate and complex models to describe the visual defect information but remains fast and efficient because it is applied on a reduced set of significant regions generated by the simplification step.

List of Publications

1. N.G. Shankar, and Z.W. Zhong, "Rule-based Inspection of Wafer Surface," Proceedings of the WSEAS International Conference on Electronic, Control & Signal Processing, pp. 451-237, Singapore, December 9-12, 2002.

2. N.G. Shankar, N. Ravi, and Z.W. Zhong, "On-Line Defect Detection in Web Offset Printing," Proceedings of the 4th International Conference on Control and Automation (ICCA'03), pp. 794-798, Montreal, 2003.

3. N.G. Shankar, and Z.W. Zhong, "Rule-based Classification of Defects on Semiconductor Wafers, "ICMAT Symposium L, Singapore, December, 2003.

4. N.G. Shankar, Z.W. Zhong, and N. Ravi, "Classification of Defects on Semiconductor Wafers using Priority Rules, "Defects and Diffusion Forum, Volume (230-232), pp. 135-148, 2004.

5. N.G. Shankar, and Z.W. Zhong, "Defect Detection on Semiconductor Wafer Surfaces," Microelectronic Engineering, Vol. 77/3-4 pp 337-346, 2005.

6. Z. W. Zhong, and N.G. Shankar, "A Clustering Technique for Defect Inspection," Proceedings of the 6th WSEAS Int. Conf. On Evolutionary Computing, Lisbon, Portugal, pp. 281-284, June 16-18, 2005.

7. N.G. Shankar, and Z.W. Zhong, "Segmentation of Biological Images using Rule-Based Approach," Biomedical Devices and Instrumentation, ICMAT Symposium C, Singapore, pp.3, July 3-8, 2005.

8. N.G. Shankar, and Z.W. Zhong, "Improved Segmentation of Semiconductor Defects using K-Level 2-D Wavelet Transform," Third Vacuum and Surface Sciences Conference of Asia and Australia (VASSCAA-3), pp.24, Singapore, July 3-8, 2005.

9. N.G. Shankar, and Z.W. Zhong, "A New Rule-Based Clustering Technique for Defect Analysis," Microelectronic Journal, Vol. 36/8, pp. 718-724, 2005.

10. Zhong Z. W. and N.G. Shankar, "A Clustering Technique for Defect Inspection," WSEAS Transactions on Electronics, Vol. 2, Issue 4, October 2005, ISSN 1109-9445, pp.185-188. http://www.wseas.org

11. N.G. Shankar and Z.W. Zhong, "A Rule-Based Computing Approach for the Segmentation of Semiconductor Defects", Microelectronic Journal, Available online 6th September 2005. In press.

12. N.G. Shankar and Z.W. Zhong, "Improved Segmentation of Semiconductor Defects using Area Sieves", Machine Vision and Applications Journal, Available online 9th February 2006. In press.

List of Figures

Page

Page

Page

Page

Page

List of Tables

Chapter 1

Introduction

The work proposed in this thesis finds its place in a broad domain of research referred to as Pattern Recognition [1-3]. In this general context, the problem to solve is how to use a machine for the automatic recognition, description, classification and grouping of defect patterns in digital images. The definition of pattern is somehow tricky. Watanabe [1] defines it as "opposite of a chaos", as "an entity, vaguely defined, that could be given a name". According to the application a different definition of pattern may be given. For example, a defect pattern could be a texture, or a specific object. Defective pattern recognition problems find important applications in several engineering and scientific disciplines such as, biology, medicine, artificial intelligence, and remote sensing and computer vision. This research work particularly interested in computer vision applications such as in semiconductor and printing industries, although the same pattern recognition strategies could be extended to medical or remote sensing domains.

The problem studied in this thesis is referred to as defect image segmentation. Defect pattern detection is an attempt to analyze the visual information contained in digital images by describing what defect is in the image and where it is located [4]. It is a process very similar to vision. Both try to discover from images what is in the world, and where it is located. Both provide us with the necessary information to interact or react to the visual input [5]. Defect image segmentation systems have already improved the efficiency of several industrial automation tasks and it is expected to reduce the need for the presence of a human operator in areas, which are dangerous, unhealthy or simply not accessible. Also, automated inspection of semiconductor wafer defect data has become increasingly important over the past several years as a means of quickly understanding and controlling contamination sources and process faults, which impact product yield.

In spite of almost 50 years of research, a general-purpose machine defect pattern recognizer does not exist yet. Most defect image segmentation techniques used in the past were designed to solve a specific problem [5]. This approach has been successful in implementing unsupervised image analysis solutions in several application domains such as inspection of industrial parts, medical imagery, traffic surveillance and autonomous robotics. These techniques have been tuned to deal with problem's specific constraints.

Thus, researchers obtained optimized solutions in terms of quality of the defect image analysis and minimization of computational requirements. Such approach is tailored to a specific problem. Therefore, its use is limited to that application and cannot be easily extended to others. This limits the use of computer vision in those applications where the visual data content and the purpose of the defect image analysis is not known a priori. Such as object-based coding, automatic organization and retrieval of databases, biometric recognition, remote sensing, autonomous robotics and advanced video surveillance. Moreover, it increases the costs for the design of unsupervised image analysis systems.

Nowadays, the improved computational capability of digital computers justifies the research for more complex techniques with a higher degree of generality. It is increasingly important to find modular solutions that can be adapted to different scenarios and constraints, eliminate the need for new designs and minimize the presence of an expert. Moreover, the definition of an evaluation process that can be effectively used to compare the main properties of the existing defect detection algorithms becomes a critical issue given the number of new approaches that are proposed each year.

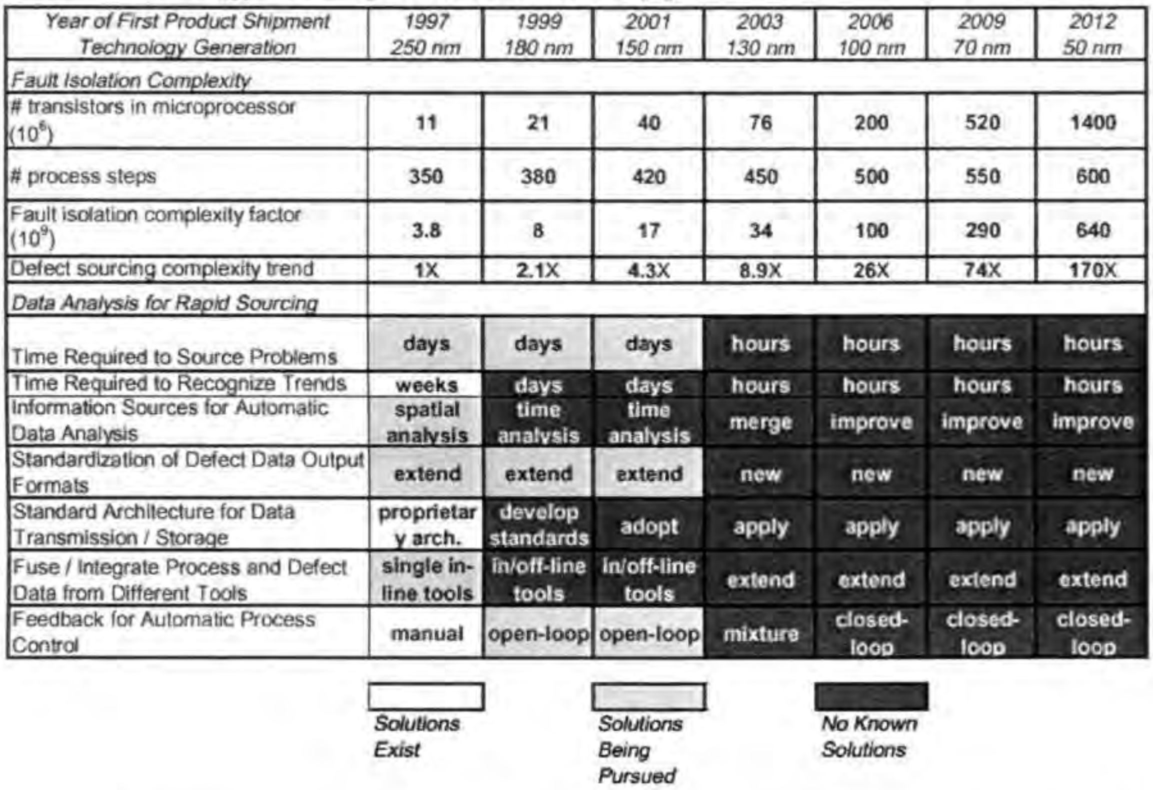
1.1 Background and Motivation

Automated analysis of semiconductor wafer defect data has become increasingly important over the past several years as a means of quickly understanding and controlling contamination sources and process faults, which impact the product, yield. Trends towards larger semiconductor wafer formats and smaller critical dimensions have caused an exponential increase in the volume of visual and parametric defects that must be analyzed and maintained by the semiconductor device manufacturer. This expanse of data has necessitated the development of automation tools for wafer defect analysis. It has been estimated that up to 80% of the yield loss in the production of high-volume, very-large-scale integrated (VLSI) circuits can be attributed to random visual particle and pattern defects [6]. Contamination particles that did not create problems with 1μm design rules can now be categorized as "killer defects" as critical dimensions dip below 0.18μm, which result in improper electrical device functions [7]. The continued trend in semiconductor manufacturing towards higher density devices and larger wafer formats is resulting in a greater need for automated yield analysis tools. For the past few years, the International Technology Roadmap for Semiconductors (ITRS) [8] has highlighted the increasing need for technologies that address key defect detection and characterization requirements.

The need to automate the process of data reduction has been highlighted in the 1997 National Technology Roadmap of Semiconductors as shown in Table 1.1. This table shows the technology requirements information developed by the Defect Reduction Technology Working Group to meet future productivity goals. Under "Fault Isolation Complexity", it is estimated that complexity in defect source will increase by as much as 170 times over the next 15 years as device density and the number of process steps continue to increase.

Defect data is generated from in-line inspection tools, particle monitors, pressure sensors, relative humidity sensors, other wafer processing tools (during dicing) and wafer-mounting devices located throughout the plant. Defect data that arises from various product and process inspection points in the semiconductor assembly will become nearly impossible to maintain by purely manual and only partially automated means. Coordinating and integrating these data sources is becoming necessary to rapidly control device yield as product complexity increases and fabrication costs continue to rise.

Table 1.1: Portion of the Technology Requirements Table for Defect Reduction in the 1997 National Technology Roadmap for Semiconductors [9].



Year of First Product Shipment *Technology Generation*	*1997* *250 nm*	*1999* *180 nm*	*2001* *150 nm*	*2003* *130 nm*	*2006* *100 nm*	*2009* *70 nm*	*2012* *50 nm*
Fault Isolation Complexity							
# transistors in microprocessor (10^6)	11	21	40	76	200	520	1400
# process steps	350	380	420	450	500	550	600
Fault isolation complexity factor (10^9)	3.8	8	17	34	100	290	640
Defect sourcing complexity trend	1X	2.1X	4.3X	8.9X	26X	74X	170X
Data Analysis for Rapid Sourcing							
Time Required to Source Problems	days	days	days	hours	hours	hours	hours
Time Required to Recognize Trends	weeks	days	days	hours	hours	hours	hours
Information Sources for Automatic Data Analysis	spatial analysis	time analysis	time analysis	merge	improve	improve	improve
Standardization of Defect Data Output Formats	extend	extend	extend	new	new	new	new
Standard Architecture for Data Transmission / Storage	proprietary arch.	develop standards	adopt	apply	apply	apply	apply
Fuse / Integrate Process and Defect Data from Different Tools	single in-line tools	in/off-line tools	in/off-line tools	extend	extend	extend	extend
Feedback for Automatic Process Control	manual	open-loop	open-loop	mixture	closed-loop	closed-loop	closed-loop

□ *Solutions Exist*

▨ *Solutions Being Pursued*

■ *No Known Solutions*

The main issues regarding "Data Analysis for Rapid Sourcing" in Table 1.1 is in the reduction of time required to source manufacturing problems and recognizes trends [Chapter 3] in the semiconductor industry. It can currently take from weeks to months to manually analyze and bring together all the data sources necessary to make a determination about yield-impacting events in the line. There are many areas of both research and standardization that need to be concurrently pursued if a high degree of automation is to be achieved.

Basic studies reveal that each object has different surface intensity, shape, color, and texture. Each and every time, the user has to teach the system to identify and recognize the objects by some means. Secondly the user has to teach the system with various matrix tolerances and the system should be able to tell the user about the shape, size, and defect type and how to segregate into various classes for future references. As manufacturing processes mature, the occurrence of anomalies arises from particle contamination, mechanical damage and process variations due to improper equipment calibration or miss-out calibration or poor maintenance of the equipment. Equipment with built in defect recognition algorithms had no prior knowledge of newly developed defect events and also the accuracy of detecting equipment slowly degrades due to aging. Since, most of the front-of-line process equipment in the semiconductor assembly accustomed to the defect data supplied by the wafer manufacturer as wafermaps. These wafermaps have no prior knowledge of the defects generated during assembly process. Hence there is a necessity to detect these defects along with wafer manufacturer defect data (wafermap) upstream of the die-bond process in the semiconductor assembly. Also the speed of processing the defective pattern utilizes much of the system operating time.

Defective pattern recognition (wafer die pattern) is accomplished by converting the 2D graphical image into an abstract representation. One of the most efficient ways to describe an object is by drawing the contour outline of the object (wafer die pattern). For human beings, recognition of contours comes very naturally. However, for machines, contours are sets of pixel, which are similar in intensity or color and have not evolved to do similar as human beings. For example, objects, which contain large sets of pixels with similar intensity and colors, may be hard to identify if you had to distinguish many elements in the pattern. On the other hand, an image with distinct, sharply contrasting intensity or color patterns would be easier to recognize.

The traditionally formulated techniques utilized in wafer inspection systems find less accurate, non-uniform recognition of defect patterns in the industry, very slow (off-line inspection because of high resolution required), and difficult in handling multiple defect components before and after dicing of the wafer.

The main goal of this research is to demonstrate suitability and superiority of the custom computing approach for all levels of vision algorithms. Special purpose architectures for vision are usually targeted for different levels. The recognition systems utilized in most of the semiconductor and printing industries facing difficulties in classifying the defective patterns for future reference and to take corrective measures in processes. The goal of this research is also to implement high speed processing to recognize defects from IC chips during assembly after wafer dicing and also in the online offset printing systems.

1.2 Objectives

The objectives of this research project are:

1. To evaluate a new rule-based approach to recognize a wide range of defects.
2. To automate the defect pattern recognition for the first time in semiconductor assembly after wafer dicing (post dicing) and offset printing.
3. To investigate a rule based approach to categorize the defect patterns to its specific structure (bridging, diffusion fault, scratches, chip die, and glassivation void, etc.,)
4. To evaluate a new wafermap with the in-house defects using wafer map editor
5. To evaluate real-time defect detection in web offset printing systems.

1.3 Scope

This thesis investigates a rule-based approach to detect defect patterns and to classify the defect patterns that appear on the semiconductor wafer surfaces. To obtain a general and modular defect pattern detection technique, the proposed rule-based approach adopts a hierarchical perspective. A formal analogy has been drawn between the structure of defect patterns and the symptom of disease in clinical practice. The defect patterns to be recognized are viewed as decision made to a particular disease. In this work, it is proposed to select right set of rules according to the application at hand. Algorithms are developed in support of the rules at each level. Design goals include detection of flaws and correlation of defect features based on co-occurrence matrix. The system is studied in identifying the defects on the wafers after die sawing. Each unique defect structure is defined as an object. Objects are grouped into user-defined categories such as chipping, metallization peel off, silicon dust contamination, etc. after die sawing and micro-crack, scratch, bridging, etc. from the wafer. The study also evaluates the rule-based approach with particular attention to possible applications in segmenting defects in offset printing, and biological images.

1.4 Thesis Layout

The design of an image segmentation technique or more generally the design of a defect pattern recognition system essentially involves three successive steps [6]: data acquisition, data representation and decision-making.

In this thesis, data acquisition corresponds to the extraction of features from the digital input data. This step is discussed in Chapter 3. The data representation step is implemented by organizing these features in classes and by describing them with their representatives. This procedure is based on clustering techniques, which are discussed in Chapter 3.

The decision making step is guided by rules defined on the relationships between the obtained representatives. These rules and relationships are proposed and defined in Chapter 3, 4, 5, and 6.

In greater details this thesis is organized as follows. First in Chapter 2, it briefly describes the need for rule-based system and various computer vision algorithms. Review of various defect pattern detection systems is also discussed in Chapter 2.

The definition of multiple features for the description of visual scenes requires techniques for their classification. The important steps towards an unsupervised classification procedure are discussed. Moreover, a review of several clustering algorithms has been proposed. This analysis is used to justify the choice of the classification algorithm selected for the proposed segmentation technique. This is defined in detail in Chapter 3.

Color features are not the only available description of visual information. Texture features are necessary information to complete the analysis of the image. Chapter 4 discussed the definition of texture features. Also, it has reviewed the major approaches available in literature underlying their advantages and limitations in the scope of defect image segmentation.

Chapter 4, 5, and 6 evaluate analytically and empirically the performances of the proposed image segmentation technique on semiconductor, printing and biological applications. Finally Chapter 7 concludes the achievements of this thesis and suggests its recommendations.

Chapter 2

Literature Review of Pattern Recognition

2.1 Introduction

In the majority of semiconductor assembly, the visual inspection process of the wafer surface depends on manual review by human experts. Since the inspection task requires extreme concentration, the time that an inspector can continue the task is quite limited, and still, it tends to be quite slow and inaccurate. As IC feature sizes shrink, semiconductor processes become more complex, and new defect classes become yield limiters. Contamination particles that did not create problems with 1μm design rules can now be categorized as "killer defects" as critical dimensions dip below 0.18μm, i.e., defects which result in improper electrical device functions [6]. Also the increasing demands for miniaturization, high electrical performance and high I/O pin count has resulted in intensive research and development of advanced IC (integrated circuit) packages [10, 11], which results in much critical inspection task. The continued trend in semiconductor manufacturing towards higher density devices and larger wafer formats is resulting in a greater need for automated yield analysis tools. The increased application of image-based defect detection and review workstations for process monitoring and characterization is generating considerable amounts of data for evaluation by production personnel. This data was necessary to evaluate the state of the manufacturing process and to ultimately improve product yield in a timely manner. For the past few years, the International Technology Roadmap for Semiconductors (ITRS) [12] has highlighted the increasing need for technologies that address key defect detection and characterization requirements.

2.2 The Inspection Task

Defect detection in semiconductor assembly manufacturing has become increasingly important over the past several years as a means of quickly understanding and controlling contamination sources and process faults, which impact the product yield. To address the above issue, automation technologies in defect detection and review are being developed by universities, laboratories, industry, and semiconductor equipment suppliers. Several techniques were adopted such as Automatic Defect Classification (ADC) on the sub-die or defect level and Spatial Signature Analysis (SSA) on the whole-wafer level [13]. One of the few commercially available ADC systems extracts numerous spatial and textural features from the defect region, and applies a set of user defined fuzzy predicates for defining the defect classes.

Defect detection is performed by directly comparing the two complex wavefronts taken from corresponding fields of view from adjacent die on the wafer. Difference images can be computed as either amplitude or phase or as a composite difference. In order to compare two images from different die, the pixel values must be aligned in the x and y directions (spatial registration) and matched in terms of their overall intensity and phase offset (normalization). Spatial registration of the complex images is performed via an automatic registration algorithm described previously [14]. There are some semiconductor inspection applications, however, in which a reference image is either unavailable or is of such poor quality that it cannot be used as a reliable basis for comparison. In the event that the defects need to be redetected at a future date, the lack of a reference image dictates that a non-referential defect detection (NRDD) approach [15] can be employed. For example, if a yield engineer sees a new defect being detected by the inspection system, his first goal will be to determine the source of the defect. One way to determine the source would be to search the historical defect image database to find previously diagnosed cases in which the defect resembles the new one. Before this database can be searched for defects with similar appearance using a technique such as content-based image retrieval [16], the defects within the database must first be redetected. Finding defects without the aid of a reference image is relatively easy for a human but is an extremely challenging task for a computer vision system.

Automated techniques that have been developed to date are highly application-specific in that the image processing techniques used are tuned for the specific object and background that are expected to be present in the image [17]. Several promising well-known techniques such as digital holography [14], digital shearography [18], and semiconductor neural networks system have been developed and reported by several research groups [19] for classification of the feature vectors and the defect classes. Rule-based inspections of semiconductor wafer surface have been reported [20, 21].

Hence, defects arises during IC assembly has to be taken care to improve yield as well as efficiency of the assembly line. The wafer die inspection criteria listed in sections 3.7.3 and 5.2 requires good established systems to detect such defect patterns and further classification. Rule-based approach is currently going through a major evolution and such an approach is tried to detect defects of flaws as small as 1μm on parts up to 8-inch wafer size [71]. Hence, there is a necessity of automated routine to detect defects. The current requests are good quality, save payroll cost, improve throughput, rationalization of consumption and cost reduction.

2.3 Previous Work

The objective of wafer inspection should include image acquisition, defect detection and defect classification. Issues based on defect classification were well explained by Chou et al. [22] and A.K. Jain et al. [23]. In general automatic inspection systems use one or a combination of two approaches: design rule checking or image-to-image reference [24]. A pure design rule system should check and compare every pixel in the digital image under inspection with the corresponding pixel in the reference image; the image to be analyzed is perfectly registered. With this approach, image registration is a major problem.

Optical special filtering [25] can also be used in defect detection on masks and patterns. This method is very fast but a major disadvantage is that small defects cannot be recognized. Wavelet technique [26] has also been involved in wafer inspection. A complete review of the related literature was found in Babian et al. [27], Newman et al. [28], Moganti et al. [29] and Klaus et al, [30]. The above methods need a database of images or some prior knowledge. A self-reference technique to overcome the mentioned difficulties was developed by Dom et al. [31], in which the comparison was made using the repeating cells in the image. Khalaj et al. [32] developed further by extracting the building block of repeating patterns from the acquired image. The defects are detected by comparing the resulting building block with the image and estimate the frequency components using the ESPRIT algorithm [33]. A golden-template self-generating method for patterned wafer image was discussed [34], where a defect-free golden template building block was extracted. Finally, a pixel-to-pixel comparison was all needed to find the defects.

Various parallel processing techniques have been developed and applied to process the pattern of interest. To achieve these objectives a number of neural network methods, such as higher-order neural networks (HONN), were used to recognize position, scale and rotation discussed by Lilly et al. [35]. A mixed signal design approach is used to exploit the massively parallel computational power of the neural network architecture for video motion detection [36]. Highly parallel computation on the pixel level is achieved by using compact analog circuit design for the neuron and synapse cells.

Parallel processing of images per pixel will offer the highest possible speed in functions related to low-level vision. This is indeed the present trend in real-time hardware for digital image processing. Digital signal processors have advantages in immunity to component noise and mismatches, more ready programmability and shorter development times but considerably larger chips than their analog equivalents. The smoothing of output image using an active resistor network was discussed in H. Kobayashi et al. [37].

Land's assumptions and various versions of color constancy algorithms are discussed [38], which reduces the computational complexity of color constancy problem tremendously, perhaps more than other models.

Multiple-view approach to three-dimensional object recognition using Hopfield network was discussed [39]. The surface and vertex correspondence problems using Hopfield network is discussed in which a 3-D object recognition system adopting multiple-view approach calculates the surface matching score between the input image and each object model in the model database (which is in 2-D projections).

A general concept for pattern recognition was described by Widrow [40] which involves in the use of invariance net followed by a trainable classifier. The key ideas of how an invariance net can be trained or designed to produce a set of outputs that are insensitive to translation, rotation, scale change, perspective, etc., of the test pattern using a multi-layer adaptation algorithm was described. Applying this algorithm to wafer inspection systems will lead to insights that will hopefully allow a mathematical analysis.

2.4 Yield Management Today

Yield management systems today are available commercially and developed in-house by semiconductor manufacturers. Although in-house yield management systems have been evolving for many years, the first commercial systems became available around 1995 [41]. The initial focus of these systems has been to track defect data. It has been estimated that up to 80% of the yield loss in the mature production of high volume, very large-scale integrated

(VLSI) circuits can be attributed to random visual particle and pattern defects [6]. The fundamental purpose of a yield management system is to collect and store relevant manufacturing data. These systems must also provide the yield engineer with search and retrieval tools that allow easy access to the data. From a high-level point of view this data is used to improve yield by monitoring yield (e.g., probe yield at electrical test), linking yield loss to defects and parametric data, and finally, linking defects and parametric data to processes and tools. With this knowledge, a recovery action plan can be created and executed. The system was used to confirm that correct actions were taken or not.

For product data, the wafermap file from in-line inspection represents the primary mechanism for associating yield loss with the manufacturing process. The wafermap file generally consists of a list of (unclassified) defect coordinates and an estimate of defect size. The wafer identification (ID), slot ID, lot number, and inspection tool type are also stored in this file, which is then maintained as data records in the database. In the database, images are linked to defects on identified wafers and lots. WIP (wafers in process) data is then used to link process equipment to wafers. The yield engineer therefore depends on lot and wafer information, along with personal experience, to query the manufacturing process to determine (i.e., source) problems in the line.

2.5 Automating the Analysis of Wafer Data

Automating the analysis of wafer defect data to provide informational content is necessary to lay the foundation for knowledge discovery in the manufacturing environment. Defect, wafermap, binmap, bitmap, and statistical monitoring techniques were developed to the extent that they are currently being tested and proven in the field. Automatic Defect Classification (ADC) has been developed to provide automation of the tedious manual inspection processes associated with defect detection and review. Although the human ability to recognize patterns in data exceeds the capabilities of computers in general, effectively designed ADC can provide a more reliable and consistent classification result than can human classification under well-defined conditions. ADC also does not suffer boredom, fatigue, or personal problems.

ADC was initially developed in the early '90s to automate the manual classification of defects during off-line optical microscopy review [42-44]. Since this time, ADC technologies have been extended to include optical in-line defect analysis [45] and SEM off-line review. For in-line ADC, a defect may be classified "on-the-fly", i.e., during the initial wafer scan of the inspection tool, or during a re-visit of the defect after the initial wafer scan. During in-line detection the defect is segmented from the image using a die-to-die comparison or a "golden template" method [22]. During off-line review the defect is re-detected using the specified electronic wafermap coordinates and die-to-die or golden template methods.

Spatial Signature Analysis (SSA) was initially developed in 1994 to address the need to intelligently group, or cluster, wafermap defects together into spatial signatures that can be uniquely assigned to specific manufacturing processes [46-49]. Standard practice has been to apply proximity clustering that results in a single event being represented as many unrelated clusters. SSA performs data reduction by clustering defects together into extended spatial groups and assigning a classification label to the group that reflects a possible manufacturing source. SSA and ADC technologies are now being combined to facilitate intelligent wafer sub-sampling for efficient off-line review and improved ADC classifier performance [50]. SSA technology has also been extended to analyze electrical test binmap data to recognize process-dependent patterns [51].

The basic input to all of the classifier approaches are descriptive features [52, 53]. Features are numerical or syntactic [54] descriptions of the defect object within an image. Numerical features include measurements of the defect image intensity, texture, morphology, color, etc. and are used to distinguish one defect from the next. The methods used for classifying defects, vary greatly although all feature-based. There are two broad categories of classifier in use: rule-based classifiers with a fixed number of pre-defined classes (pre-defined by the system developer) and trainable classifiers that are trained in the field by the end-user.

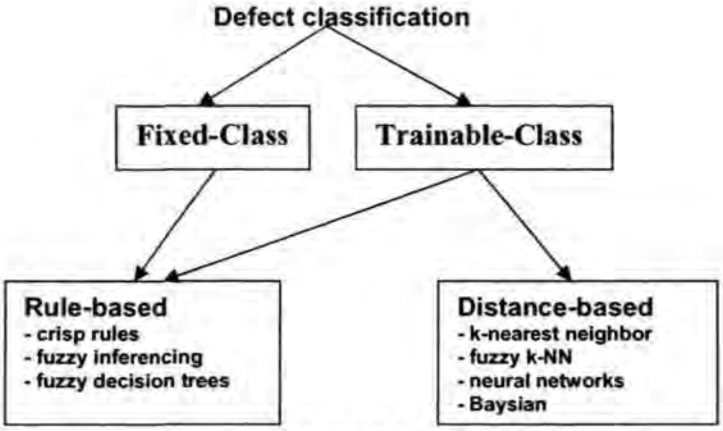


Figure 2.1: Flow diagram representing the differences between various classification systems.

Fixed-class systems have come into popularity for in-line applications since the resolution of these systems is generally less than off-line review microscopes. The reduced sensitivity of the in-line systems results in simple classification schemes that classify defects, for example, by size or brightness. There is no user training of a fixed-class system. The result is ease-of-use. The down side of this approach is that the system cannot easily be trained to accommodate new defect classes that are manufacturer-specific.

A trainable system (e.g., based on distance-based classifiers such k-nearest neighbor [55] or neural networks [56]) can accommodate the wide range of defect types associated with different inspection points in the process, various process layers, or products, but can be cumbersome to train and maintain. The concept of having a classifier system that is ready to use has prompted the extension of the fixed-classifier concept to some off-line review systems but the lack of classification flexibility is considered to be an undesirable limitation by yield engineers. Ultimately there will likely be a fusion of these two approaches that allows the yield engineer to use the system immediately to classify basic categories of defects, while fine-tuning these categories through a training process over time.

2.6 Statistical Process Control

Wafer-based SPC today depends on tracking particles and clusters. A cluster is defined as a group of wafer defects that reside within a specified proximity of each other. Current strategies typically involve removing cluster data and tracking the remaining particle data under the assumption that they are random, uncorrected defects. Field-testing of the advanced clustering capabilities of SSA has revealed that this basic approach can be modified dramatically to reveal much information.

The ability of SSA to locate and analyze this event as one long scratch removes ambiguity from the clustering result. It allows the user to assign process-specific information via the automatic classification procedure that will allow tracking (in an SPC sense) of these types of events to monitor total counts, frequency of occurrence, etc. Care must also be taken in analyzing random events on a wafer. Unless a yield engineer happens to view the wafermap, the error could go un-assessed indefinitely. Using an approach such as SSA results in the separation of wafer events into random and systematic events (both clustered and distributed) that provide a higher level of information about the manufacturing process.

Using this informational content to separate and monitor random defects from systematic distributions from scratches, streaks, double slots, and other clusters, provides the yield engineer a much clearer window into the manufacturing process. The ability to use this level of SPC monitoring to initiate data mining processes should also result in much more reliable automatic decision-making.

As the semiconductor package density continues to increase and critical dimensions become smaller, the size of killer defects will shrink. As manufacturing moves towards the next technology node, 0.18 μm, these killer defects can be on the order of 0.09 μm in size. These defects will be difficult, or even impossible in some cases, to detect using well-established optical microscope-based defect inspection systems. Although new inspection technologies are under development, the importance of electrical testing of devices will play an important role in predicting product yield and determining the health of the manufacturing process. For this case a particular two-dimensional spatial pattern, or signature, of die failure on the wafer is well explained by Tobin [57] under the topic Spatial Signature Analysis (SSA).

2.7 Inspection in Semiconductor Manufacturing

Semiconductor wafer analysis using microscopy techniques falls into three main categories [58]: (i) in-line microscopy for rapid, whole wafer defect detection, (ii) off-line microscopy for defect review and failure analysis, and (iii) microscopy techniques for critical dimensions.

In-line microscopy must keep up with the flow of manufacturing. At an inspection point, of which there are several in the process stream, a fraction of the wafers will be inspected. The fraction of wafers inspected depends on the maturity of the process, the complexity of the product and the particular process step being monitored using this approach requires two to twenty minutes to scan an 8 inch wafer. At a defect sensitivity of 0.25 μm, a throughput is required of approximately 2-GHz, i.e., $2x10^9$ image pixels must be captured, moved through the image processing system and reduced to a small set of descriptive features, e.g., defect location and size, every second. These systems must therefore rely on high-speed and reliable imaging technology. Optical microscopy using charge coupled devices (CCDs) are the systems of the choice for in-line defect detection with CCDs configured in-line scan or time-delay integration (TDI) format.

Off-line microscopy typically involves re-sampling a small fraction of the wafers inspected in-line. Off-line tools therefore do not require the same throughput but do depend on higher image fidelity so that human-level decisions can be made regarding defect type and source. Higher quality images for failure analysis are obtained such as; color CCD sensors, con-focal microscopy and scanning electron microscopy (SEM).

2.7.1 Sensors for Image Acquisition

CCD sensors are used almost exclusively as imaging detectors for optical microscopy. A CCD sensing element is a metal oxide semiconductor (MOS) capacitor that converts light photons into electrons [58]. These sensing capacitors are arrayed in linear and area configurations to provide line-scan and area imaging capabilities [59]. Long wavelength light will tend to pass through the sensing region of the MOS element and be absorbed deep in the substrate below the active area of the sensor. Short wavelength light will tend to be absorbed in the passivation layer above the active area of the device. CCD devices operate by integrating the incident photon energy over a short period of time and then transferring the recorded charge out of the device as an analog signal.

At room temperatures, electron-hole pairs are created in these devices that are sufficient in energy to fill the storage region and cause a contamination of the signal. Applying a positive voltage to the silicon gate, which results initial deficiency prior to integration reduces this thermal effect. The user improves the signal-to-noise ratio by controlling the integration of light to the sensor, i.e., the exposure to light between charge transfer cycles. The allowable time for integration is driven by the required image quality and the application [14].

A longer integration time requires that the inspection surface persist in the field of view of the imaging sensor, or pixel, for a longer period. This proves problematic for a high-speed, high-resolution imaging environment since high resolution and high throughput equates to short integration times. For in-line inspection, the dominant CCD sensor configuration is the TDI sensor [12].

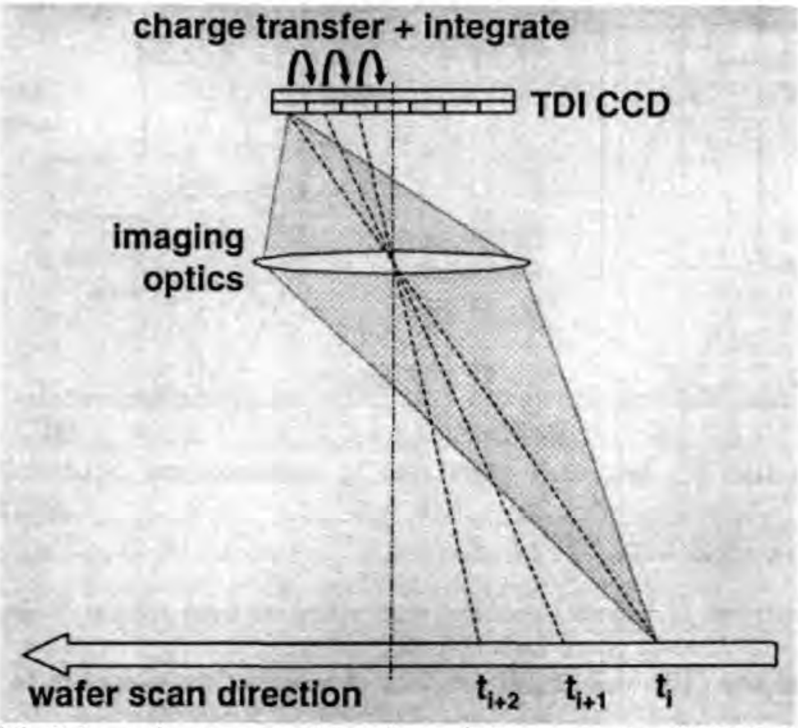


Figure 2.2: Schematic representation of time delay integration CCD device [12].

The TDI is an area array of imaging elements that works as a line-scan integration device. Figure 2.2 shows how the CCD TDI operates. A linear region of the wafer surface is imaged onto the first line of the TDI sensor at time $t_{i.}$ After a predetermined integration time, the wafer has shifted during continuous scan and a new line on the surface is imaged onto the first CCD sensor. Meanwhile, the charge from the first sensor has been shifted to the second CCD line, at time $t_{i+1,}$ and exposed to the same viewing region of the wafer as the previous line. This process continues down the array resulting in an n-fold integration of the imaged line, n being the number of integration lines available on the device. The output of the TDI is a single line of integrated data with improved signal-to-noise ratio (SNR), analogous to the output of a common line-scan CCD device. As the wafer passes under the imaging optics, a semiconductor device image is built up line by line.

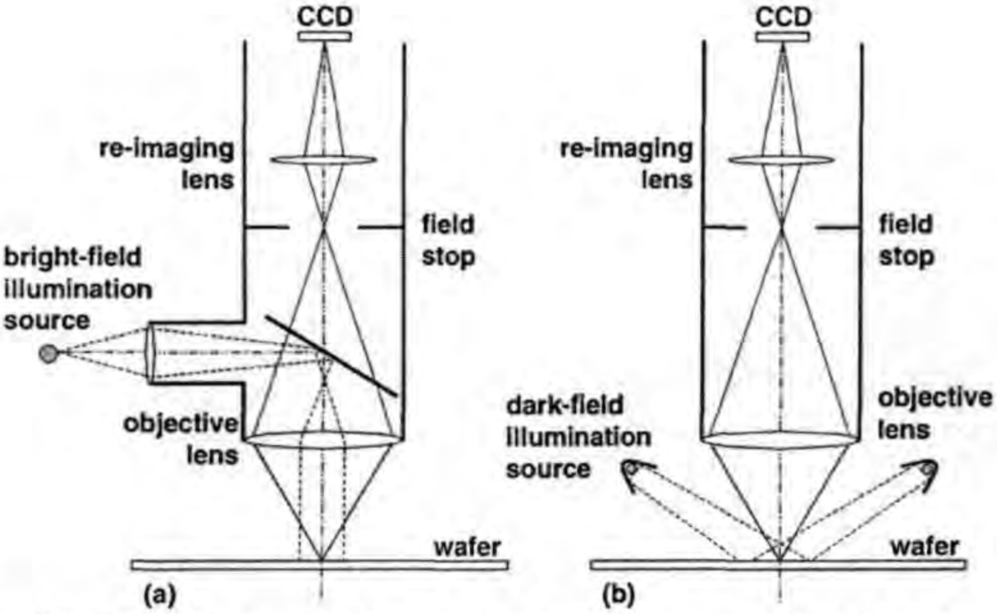


Figure 2.3: Schematic representation of (a) bright field and (b) dark-field imaging configurations [58].

The optical system used to image semiconductor device is designed to enhance various detail of the wafer surface prior to analysis. Bright-field (BF) and dark-field (DF) imaging is used to improve contrast between different materials. Figure 2.3 shows the optical configurations for BF and DF imaging. In BF imaging, the illumination path is coincident with the viewing path. Contrast in a BF image is a strong function of light attenuation and reflection between different materials. Contrast in DF imaging, on the other hand, is a result of indirect illumination such that none of the light rays will be collected by the imaging optic unless scattered by an edge or a textured surface.

An example in Figure 2.4 shows images of a finished semiconductor device at about 5 μm per pixel resolution. The device has a large stain defect in the center of the field of view. Note in the BF image that the die pads contrast highly against the focused surface but the stain is difficult to describe. In the DF image, the stain is smooth relative to the high contrast device. The DF image also tends to highlight edge information. In practice, an imaging system may employ some level of both BF and DF illumination within the system.

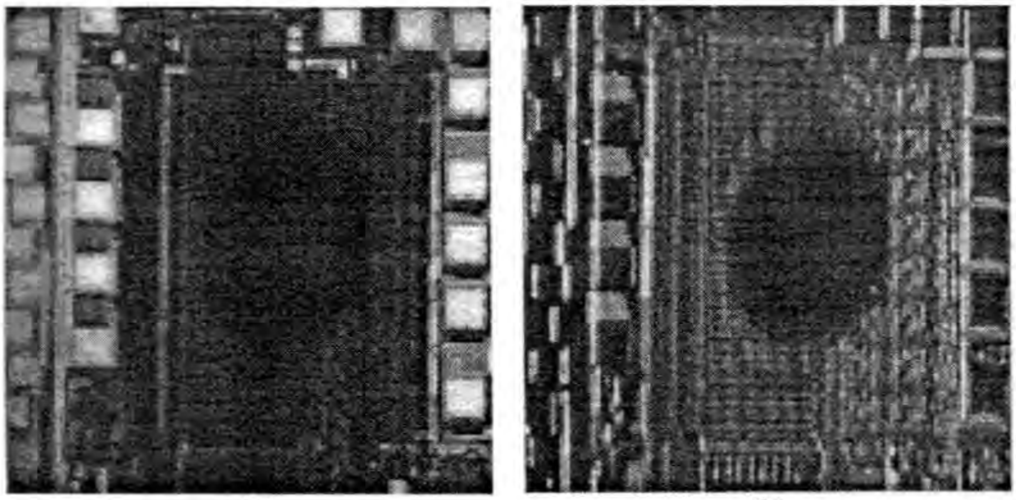


Figure 2.4: Bright-field (a) and dark – field (b) image of a semiconductor device at approximately 5 μm resolution [58].

As the critical dimension of semiconductor devices continues to shrink, defects that were too small to electrically interfere with a circuit are becoming problematic. At today's critical defect dimension size of 0.18-μm, a defect of roughly one-half to one-third this dimension can cause electrical failures through bridging. Defects as small as 80-nm can now cause electrical faults but are not visible by means of optical inspection [60]. At this resolution, scanning electron microscopy (SEM) imaging can provides high defect resolution and contrast.

The primary role of SEM in today's fabrication facility is for off-line review, failure analysis, critical dimension and overlay metrology. SEM is very effective for revealing surface structure, defect morphology and elemental composition, but current tools are slow and only now beginning to provide a high degree of automation. As the critical dimension continues to shrink, advances in SEM are required that will result in the application of SEM to high-speed, in-line applications.

2.7.2 Resolution, Data Rates, and Throughput

To achieve specified detection sensitivity on an inspection tool, it is required that the surface be sampled at the appropriate resolution [58]. For example finding a 0.25-μm defect on an 8-inch diameter wafer is equivalent to searching for an object in the size of a baseball on 58,000 acres of land. Nyquist sampling theory requires a sample size of 0.125-μm. It may be sufficient to detect a defect size of 0.25-μm. The ability to resolve such an object requires 50-nm per sample. This becomes especially critical when automating defect classification.

It is known from the semiconductor manufacturing that in order to maintain a throughput of 8-inch wafers at 3 wafers per hour, requires a pixel processing speed of about 2-GHz. This was determined by dividing the wafer area by a Nyquist sample size of 0.125-μm, and further divided by 3 wafers per hour. A series of images must be captured, processed and reported at this sampling rate. This is accomplished by scanning the wafer across the rows of die and subtracting one die from its neighbor to locate subtle differences. The following generic algorithm describes the required steps for defect detection presently used in the semiconductor field:

a) An image is captured from a test region of a die

b) The wafer stepper scans to the same location on a neighboring die and a second image is captured

c) The two images are aligned

d) And subtracted

e) Small alignment residuals and texture anomalies are filtered and removed

f) From the resulting mask the defect location and size features are extracted

g) Defect information is logged in the electronic wafermap file for later reference

h) The stepper moves to the next die chip of the wafer and the process is repeated.

To maintain the throughput suggested above, it is required that several of these steps be completed in parallel. For this reason, parallel pipeline image processing is a leading technology in this area. Pipeline image processing allows for a serial stream of functions to be processed in parallel. A Rule-Based system is one parallel processing technique to overcome the above said processing requirement. For example, an algorithm consisting of image transfer, image subtraction and image filtering has to process in parallel with each functional block running in parallel. As digital signal processing (DSP) devices and field-programmable gate arrays (FPGA) continue to mature, these devices are also being used more often in parallel configurations (e.g., multiple DSPs), and in conjunction with pipelined architectures. The suitability and low cost makes rule-based systems more attractive alternatives for us to use in this project. The details about the applicability and how, Rule-Based systems used in our application are discussed in Chapter 4,5 and 6.

2.8 Overview of Computer Vision

The goal of computer vision is to automatically construct a description of a given scene from an analysis of the sensed pattern of the scene. The sensed pattern can be a single pattern taken from a single camera, multiple views of the scene using multiple cameras or a sequence of images of the same scene taken over a period of time using multiple cameras. The description of the scene consists of the identity and localization (position and orientation of an object) of the objects present in the scene based on the measured physical attributes. Simply stated, computer vision aims at providing visual capabilities to a machine. It includes techniques from image processing, exploratory data analysis, statistical pattern recognition, cognitive science and artificial intelligence.

Most children recognize digits and letters at the age of five. The young child easily recognizes small characters, large characters, hand written and machine printed or rotated.

Pattern recognition is the study of how machines can absorb the environment to distinguish patterns of interest from this background, and make reasonable decisions about the categories of the patterns.

The best pattern recognizers in most instances are humans, yet we do not understand how humans recognize patterns. Since computers were taught to recognize patterns which is the only alternative in close resemblance to human brain. The processing speed of the brain as compared to that of the computer is something like the speed of a snail compared to that of a car. This basic difference leads to the fact that the computer and brain have their own strengths and weakness in applications. Our goal here is to introduce pattern recognition as the best possible way of utilizing available sensors, processors and knowledge to make decisions automatically.

2.8.1 Computer Vision Algorithms

Computer vision algorithms [61] are natural candidates for high performance computing due to their inherent parallelism and intense computational demands. For example, a simple 3×3 convolution on a 512×512 gray scale image at 30 frames per second requires 67.5 million multiplication and 60 million additions to be performed in one second. Special-purpose hardware provides better performance compared to a general-purpose hardware for all the three levels of vision tasks. With recent advances in very large scale integration (VLSI) technology, an application specific integrated circuit (ASIC) can provide the best performance in terms of total execution time. However, long design cycle time, high development cost and inflexibility of a dedicated hardware deter design of ASICs. Simply stated, computer vision aims at providing visual capabilities to a machine. It includes techniques from image processing, exploratory data analysis, statistical pattern recognition, cognitive science and artificial intelligence.

Designing robust and general-purpose computer vision systems is a challenging task. A number of difficult imaging conditions as well as pattern and object complexities are encountered in practice. These non-ideal and confounding conditions arise due to (i) improper lighting, (ii) shadow, (iii) occlusion, (iv) noise in the sensed image and (v) assumptions made in object representation strategies. A typical computer vision system involves a front-end image acquisition and a preprocessor, followed by a pattern interpreter. The back-end deals with interpreting the pattern from the extracted features. One of the main problems in computer vision is to automatically determine a salient set of features that is suitable for describing the pattern explicitly. In the literature many attempts have been made to design machine vision systems that mimic a human vision system. But, as the human vision system is extremely complex and not fully under-stood, these human vision-based models and approaches are not very helpful in designing practical machine vision systems.

2.8.2 Need for Real-time Vision Systems

A system in a given time instant after the presentation of input, an output is produced is called a real-time system. Shin and Ramanathan [62] have identified three major components and their interplay that characterize a real-time system. Loosely speaking, the system output must meet a time deadline since the output is related to the input changes. Brown and Terzopoulos [63] define real-time computer vision systems as follows: Real-time computer vision systems refer to computer analysis of dynamic scenes at rates sufficiently high to affect practical visually-guided control or decision making in everyday situation. Another definition of real-time system is that the response time of the machine vision system may be equal to or faster than the response of the human performing the same task. These definitions lead to an expected processing rate of about 10-30 frames per second. A computer vision system in time-critical applications such as silicon wafer inspection, it is essential that the vision processing to be done at the data acquisition rate (video frame rate of 30 frames/second). For applications based on video data, processing at this rate is an essential requirement. In military applications such as target detection, the need for real-time processing is highly critical [64]. The need for real-time processing is also very important in medical image analysis applications such as vision-guided non-invasive surgery. Similar constraints exist in other applications such as compression / decompression of images in multi-media applications. In order to meet the real-time requirements, a frame of image buffer needs to be processed in roughly 33 milliseconds. For a 512 × 512 gray level image this amounts to a data rate roughly 7.5 MHz.

The vision algorithms described in the conventional methods demand a very high execution time on a general-purpose computing platform compared to the desired real-time response. Using appropriate hardware accelerators the large disparity between the system response time and the actual response time reduces to the desired value [65].

2.8.3 Vision Task Hierarchy

Computer vision tasks can be divided into a three-level hierarchy, namely, low-level, intermediate-level and high-level [66-68]. Low-level vision tasks consist of pixel-based operations such as filtering and edge detection.

The Low-level tasks handle large amount of data (pixels), small neighborhood operations and relatively simple operations. The pixel grouping operations such as segmentation and region labeling are intermediate-level vision tasks. Intermediate-level tasks again handle local data access, but more complex pixel operations. High-level vision tasks are more decision-oriented such as point matching, tree matching and graph matching. These tasks are characterized by non-local data access, non-deterministic algorithms. Several examples of vision tasks belonging to these three-levels are shown in Table 2.1.

Table 2.1: Examples of vision tasks in three-level hierarchy.

Task level	Computational characteristics	Examples
Low	Small neighborhood data access, simple operations, large amount of data	Edge detection, filtering, and image morphology
Intermediate	Small neighborhood, more complex operations	Hough transform, connected component labeling, relaxation
High	Non-local data access, non-deterministic and complex operations	Point matching, tree matching, graph matching, object recognition

2.8.4 Architectures for Vision

In order to meet the high computational needs of vision systems, many special purpose architectures [61] have been proposed in literature. By analyzing several representative problems in computer vision, the following architectural requirements are observed:

i) For low-level vision algorithms, Single Instruction Multiple Data (SIMD) and fine-grained architectures are preferred, but for high-level algorithms, Multiple Instruction Multiple Data (MIMD) and coarse-grained architectures are required.

ii) At lower levels, communication is limited to a local neighborhood, but at higher levels the communication tends to be global.

iii) A typical pattern contains large amount of data. Therefore, a high bandwidth I/O is essential to sustain good performance.

iv) Speeding up only one stage of the vision system will not result in a significant speedup of the overall performance.

v) For good performance, the load on different processors should be balanced.

vi) In a multi-processor system, failure of some processing elements should not result in an overall system failure. Therefore, a graceful degradation should be supported.

vii) Often, a specific processor topology is preferred for an algorithm depending on its communication characteristics. Hence, flexible communication support is essential for mapping many communication patterns. The algorithm mapping should be independent of data size.

2.8.5 Hardware-software Co-design

The system designer has the dilemma of deciding which portions of an algorithm should be run on a special hardware and which should be run on a standard hardware. An application-specific hardware system that could be built without high cost and a high turn-around time would be preferred. But, dedicated systems are always coupled with high development time and large investments otherwise one can use a general-purpose processor at the cost of sacrificing the performance. The designer has to make a cost/performance analysis to arrive at an efficient partition of the problem. This optimization problem is the subject of hardware-software co-design, which refers to the process of simultaneously designing both hardware and software to meet specified performance objectives. In traditional approach, hardware and software partitions are relatively rigid. In co-design the partition is flexible and can be shifted to meet the changing performance criterion. Micheli [69] has described a framework for hardware-software co-design. An embedded system is a system with a mix of general-purpose processors, dedicated hardware as well as software running on one or more processors in addition to sensors to interact with the environment. Hardware-software co-design is an important method for designing embedded systems.

2.9 Automated Inspection

Automated detection and classification systems are becoming commonplace in industrial environments. Anil [2] presents an overview of the many applications and methods being implemented for automated visual inspection, including both in-house and commercial systems. The primary function of these systems is to segment and classify defects, where a defect can be defined as a region in the image that deviates from an expected pattern or specification. Figure 2.5 shows a typical computer vision process for detecting a defect based on the golden template or neighborhood reference method. This technique represents a segmentation of the defect from the expected regions of the image.

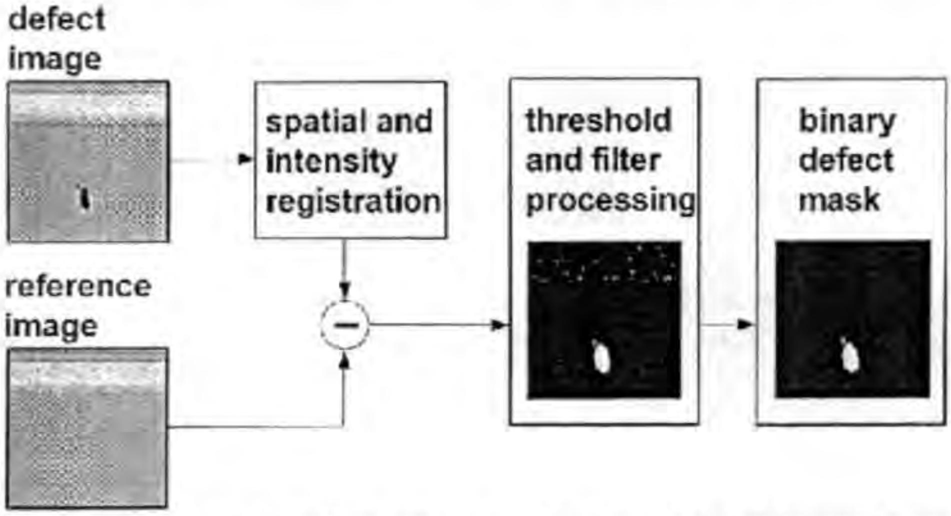

defect image
spatial and intensity registration
threshold and filter processing
binary defect mask
reference image
−

Figure 2.5: A typical computer vision process used to detect a defect between a test and a reference image. The result is a binary defect mask that segments the anomaly from the expected image components [2].

2.10 Automatic Defect Classification

A defect is detected on an in-line tool by comparing the wafer die under test to its neighbor. Figure 2.6 shows the basic steps applied to detecting a wafer anomaly on an in-line inspection tool. The wafer is scanned in serpentine fashion as shown in Fig. 2.6(a). Each die under test is compared with its neighboring die through a process of alignment, subtraction and filtering as shown in Fig. 2.6(b) and Fig. 2.6(c).

The defect data is then recorded in the wafermap by defect number (arbitrarily assigned), x and y location and size prior to database storage as shown in Fig. 2.6(d). Since each die on the wafer contains the same pattern or circuitry, the only difference should be due to local anomalies caused by surface or imbedded particle contamination, or extra or missing pattern.

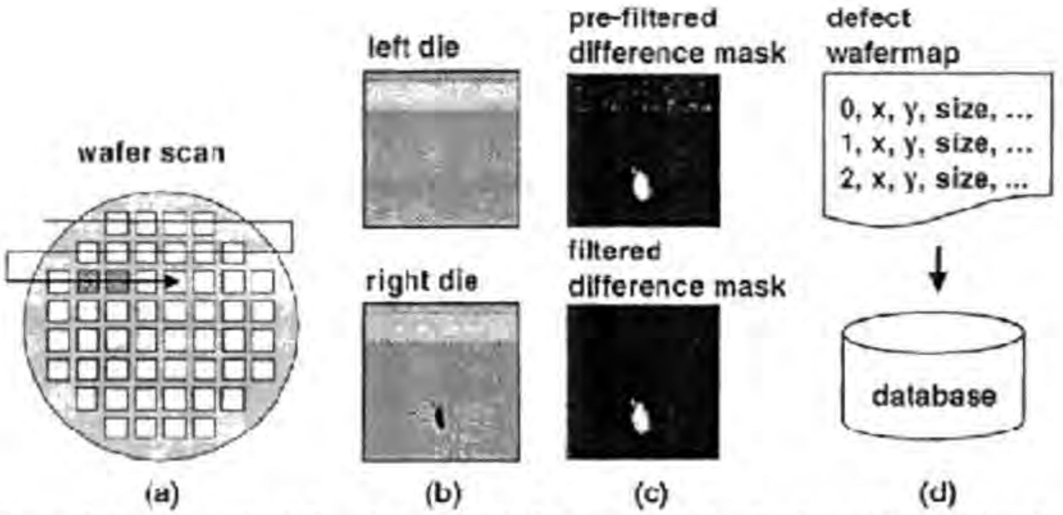


Figure 2.6: Automatic defect detection -Defect segmentation begins with scanning the wafer row-by-row, (a) collecting neighboring die images, (b) comparing and filtering to find anomalies, and (c) reporting and data storage [58].

The two basic categories of defects are the particle and pattern defects. These defects broadly encompass most categories of defects on the wafer but their description and possible manufacturing sources vary widely. The purpose of automatic detection and classification is to quickly isolate defective areas on the wafer, label and categorize the event, determine the source or cause of the defect, and quickly make a tool or process correction to reduce further yield loss.

Limitations:

1.) Defect detection before dicing of wafers i.e., only primary inspection possible.

2.) No final solution to in-line assembly.

3.) The main drawback of ADC is the time consumption and also there is no reference pattern for the whole process i.e., it considers neighboring die as reference for the die under inspection.

2.11 Spatial Signature Analysis

SSA attempts to automatically collect defects on a wafermap that come from a single manufacturing source [70]. A user-trained classifier then assigns a label that identifies the root cause. SSA begins the signature classification process by converting the electronic wafermap file into a gray-scale image where each pixel is assigned an intensity value according to the number of defects in the subtended area. Each pixel represents a first level grouping of the individual defects. Pixels are connected according to their proximity to neighbors to form cluster of defects. Clusters of pixels are connected into multi-element objects (e.g., a multi-element scratch) by means of a unique advanced clustering procedure. Prior to signature classification, objects are grouped into elemental sets depending on their proximity to neighboring clusters and on their morphology. These elemental sets are the result of a "divide and conquer" approach to the SSA problem required to reduce the complexity of signature classification.

There are four distinct sets in use with the SSA procedure denoted by global, curvilinear, amorphous, and microstructure. The assumption is made that every connected or distributed object can be categorized into the one of these elemental sets. For example, elongated objects such as scratches are assigned the curvilinear set since they have curvilinear attributes such as elongation, compactness, orientation, etc. These objects tend to be associated with mechanical wafer damage. Tightly clustered objects are placed in the amorphous set and can generally be associated with problems such as insufficient etching, or other systematic sources not related to mechanical damages. Distributed objects such as a ring pattern or a random uniform distribution of particles that are broadly distributed over the wafer surface are grouped into the global set. Microstructure objects define the final set. These objects are composed of a distribution of pixels whose sub-pixel defects are organized in a linear fashion. These pixel-level objects arise from planarization processes such as chemical and mechanical polishing.

Once an object has been assigned to a high-level set and characterized, its features are sent to a classifier where a user-defined label is assigned to the result. For this work, a pair-wise fuzzy k-Nearest Neighbor (kNN) approach has been adapted which uses a unique feature reduction procedure to optimize classifier performance.

Limitations:

1) The main drawback of SSA is that it has no relation with the defect data generated by local anomalies caused during wafer mounting, dicing, imbedded particle contamination, etc.
2) Defect detection before wafer dicing i.e., only primary inspection possible.
3) Detect specific defects in off-line.
4) The wafermap generated by SSA contains only defects identified during primary inspection and hence, no final solution to in-line assembly.

2.12 Detection of Semiconductor defects using a fractal-encoding algorithm

There are some semiconductor inspection applications, however, in which a reference image is either unavailable or is of such poor quality that it cannot be used as a reliable basis for comparison. In the event that the defects need to be redetected at a future date, the lack of a reference image dictates that a non-referential defect detection (NRDD) approach [15] must be employed.

NRDD introduces an algorithm that leverages the concept of image self-similarity. Self-similarity is an image quality that is exploited by fractal image compression schemes in that a sub-region in one area of an image is described by a mathematical transformation of another sub-region within that same image. The process of breaking the image up into different sub-regions and then determining the set of affine transformations that best map one region to another is called *fractal encoding*. The idea behind this NRDD algorithm is that a defect will appear different than the normal image background structure, and, hence, it will be difficult to find an affine transformation that effectively maps another image sub-region to the current defective one. An image sub-region containing a defect is flagged for post-processing as a defective sub-region and it is called as flagged sub-region a *focus-of-attention region* (FAR). The post-processing employed here is the application of active contours, or snakes, to the defect image sub-regions to more finely delineate the defect boundary.

Limitations:

1) This NRDD approach is in theory and in practice it may be applicable to multiple image modalities (e.g. SEM and optical microscope imagery).
2) The performances of the active contours for post-processing highly depend on how close the initial contours are to the actual boundaries of the ROIs.
3) Unfortunately, a priori information about the ROIs is generally not available in semiconductor images. This makes it difficult to employ active contour models in semiconductor image segmentation.
4) Defect detection before wafer dicing i.e., only primary inspection possible.

2.13 Semiconductor wafer defect detection using digital holography

Defect inspection metrology is an integral part of the yield ramp and process monitoring phases of semiconductor manufacturing. High aspect ratio structures have been identified in the ITRS [8] as critical structures where there are no known manufacturable solutions for defect detection. Currently there are three main technologies in use for defect detection: dark field inspection, bright field inspection, and scanning electron microscopy. The physics of all of three of these technologies will limit them in at least one of the requirements necessary to reach the goals of the current technology roadmap. M.A. Schulze et al [14] present case studies of a new inspection technology based on digital holography that addresses this need. Digital holography records the amplitude and phase of the wavefront from the target object directly to a single image acquired by a CCD camera. Using deep ultraviolet laser illumination, digital holography is capable of resolving phase differences corresponding to height differences as small as several nanometers. Thus, the technology is well suited to the task of finding defects on semiconductor wafers. Specifically, digital holography allows improved defect detection on high aspect ratio features, such as improperly etched contacts. In addition, the phase information provided by digital holography allows us to visualize the topology of defects, and even generate three-dimensional images of the wafer surface comparable to scanning electron microscope (SEM) images. In addition, the detection technology provides direct information about the topology and material characteristics than can be used to aid the differentiation of defect severity. The first production ready version of an inspection tool based on this technology, named Fathom, has been in the characterization and evaluation process.

Limitations:

1) Digital holography allows improved defect detection on high aspect ratio features, such as improperly etched contacts.
2) Detect specific defects before dicing i.e., only primary inspection possible.
3) Detect specific defects in off-line and no automated defect inspection system available practically.

2.14 Background of Development

To reduce the size of equipment in which a computer is used for, the chip mounting technology has been changed into the flip-chip technology (highly integrated packaging) by which IC chips are directly mounted on a substrate, such as "Flip Chip On Board (FCOB)" or "Flip Chip On Glass (FCOG)".

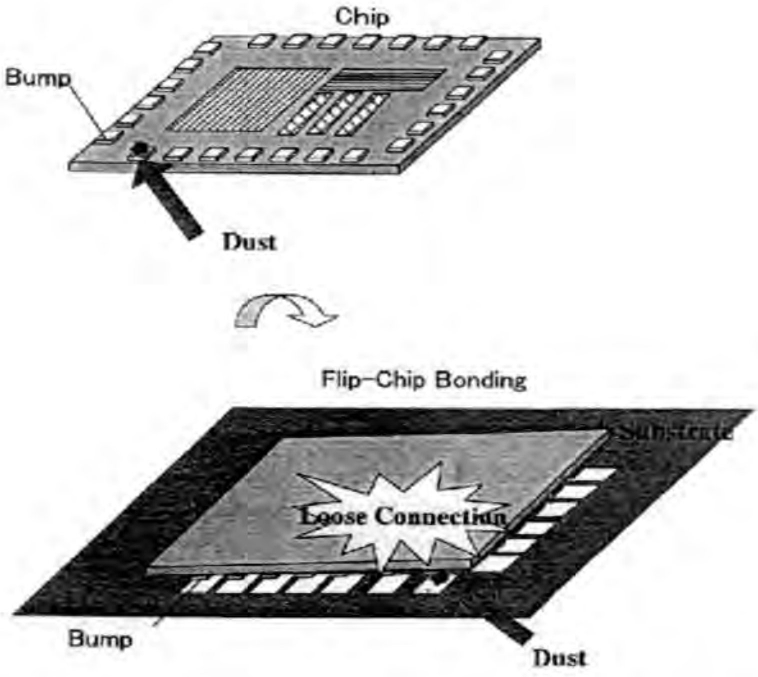


Figure 2.7: Portion of IC chip surface affected due to dust particles [72].

Once an electrical test is carried out, there is a necessity of surface inspection, because non-conductivity or cracks may occur due to dust, scratches and defects on a bond pad in the mounting process. Alternatively employing a skilled technician rather than inspection systems leads to limitation on resolution of human eyes, personal error of measurement and also increase in cost.

Figure 2.7 shows the portion of the IC chip surface affected due to dust particles in a typical flip-chip bonding and it leads to loose interconnection. The Semiconductor wafer manufacturers have a large variety of tools available to provide wafer measurements ranging from in-line whole-wafer optical inspection for defect detection, to off-line scanning electron microscopy (SEM), energy dispersive x-ray spectroscopy (EDX), and focused ion beam (FIB) for defect analysis and fault isolation.

The wafer is also electrically tested to isolate failed die for memory and cache, and to provide bin mapping for die sort. The result is the collection of a wide variety of data and processes. To achieve stringent productivity goals, this process must be fast and reliable. Methods such as automatic defect classification (ADC), spatial signature analysis (SSA), statistical process control (SPC), and data mining are but a few of the technologies being developed and tested in the fabrication environment today as a means of adding informational content to the raw measured data. This information is required to automate the yield analysis process. The evolution of these technologies along with emerging issues will be discussed. Hence, there is a necessity of automated routine to detect defects. The current requests are good quality, save payroll cost, improve throughput, rationalization of consumption and cost reduction.

To address the above issue, automation technologies in defect detection and review are being developed by universities, laboratories, industry, and semiconductor equipment suppliers. Several techniques were adopted such as automatic defect classification (ADC), dark field imaging, digital holography on the sub-die or defect level, and spatial signature Analysis (SSA) on the whole-wafer level. The main draw back of all these techniques is that they are all adopted for primary inspection i.e., before wafer dicing also they are not providing final solution to in-line assembly such as, the defect data generated by local anomalies caused during wafer mounting, dicing, imbedded particle contamination, etc.

The schematic block diagram in Figure 2.8 shows that there is no proper in-line defect detection system after wafer dicing. All the inspection systems available up-to-date is to detect defects before wafer dicing. Figure 2.8 also shows the sub-schematic (green line) block diagram of the proposed rule-based defect detection system, which is to detect defects after wafer dicing (before die-bonding).

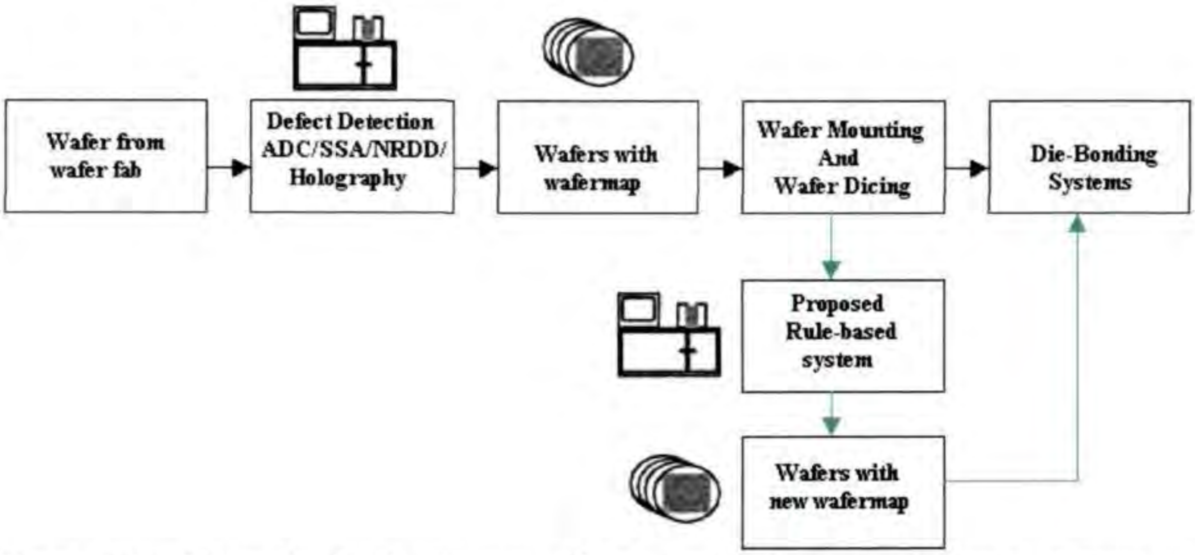

Wafer from wafer fab
Defect Detection ADC/SSA/NRDD/ Holography
Wafers with wafermap
Wafer Mounting And Wafer Dicing
Die-Bonding Systems
Proposed Rule-based system
Wafers with new wafermap

Figure 2.8: Schematic block diagram of the application of rule-based systems in semiconductor assembly.

Hence, defects arises during IC assembly has to be detected in order to improve yield as well as efficiency of the assembly line. The wafer die inspection criteria listed in sections 3.7.3 and 5.2 also requires good established systems to detect such defect patterns and further classification. Rule-based approach is currently going through a major evolution and such an approach is tried out to detect defects of flaws as small as 1μm on parts up to 8-inch wafer size [71].

2.15 What is Pattern Recognition?

Automatic machine recognition, description, classification and grouping of patterns are important problems in a variety of engineering and scientific disciplines in artificial intelligence, biology, computer vision, medicine, marketing, psychology and remote sensing. For example, a pattern could be a finger print image, a handwritten word, a human face, or an object. Given a pattern, its recognition may consist of one of the following two tasks [2].

i) Supervised classification in which the input pattern is identified as a member of a predefined class,

ii) Unsupervised classification in which the pattern is assigned to an unknown class.

Demands on automatic pattern recognition systems are rising enormously due to the availability of large databases and stringent performance requirements (speed, accuracy and cost). In many of the emerging applications, no single approach for classification is "optimal", and multiple methods and approaches have to be used.

The design of a pattern recognition system essentially involves three aspects [2]:

i) data acquisition and preprocessing, ii) data representation and iii) decision-making. Learning from a set of examples (training set) is an important and desired attribute of most pattern recognition systems. The four well-known approaches for pattern recognition are i) template matching, ii) statistical classification, iii) syntactic or structural matching and iv) neural networks.

i) Template Matching

A template (typically, a 2D shape) or a prototype of the pattern to be recognized is available. The pattern to be recognized is matched against the stored template while taking into account all allowable translation, rotation and scale changes. It would fail if the patterns were distorted due to the imaging process, viewpoint change, or large intra class variations among the patterns.

ii) Statistical Approach

In this approach each pattern is represented in terms of **d** features or measurements and is viewed as a point in a **d**–dimensional space. The goal is to choose those features that allow pattern vectors belonging to different categories to occupy compact and disjoint regions in a **d**–dimensional feature space. Given a set of training patterns from each class, the objective is to establish decision boundaries in the feature space, which separate patterns belonging to different classes such boundaries can be constructed using, a mean squared error criterion.

iii) Syntactic Approach

In syntactic pattern recognition, a formal analogy is drawn between the structure of patterns and the syntax of a language. The patterns are viewed as sentences belonging to a language and the sentences are generated according to a grammar. Sub pattern or primitives are viewed as the alphabet of the language. This paradigm has been used in situations where the patterns have a definite structure, which can be captured in terms of a set of rules, such as ECG waveforms, textured image and shape analysis of contours. Implementation of syntactic approach leads to many difficulties of noisy patterns and the inference of the grammar from training data.

iv) Neural Networks

Neural network can be viewed as massively parallel computing systems consisting of an extremely large number of simple processors with many interconnections. The main characteristics of neural networks are that they have the ability to complex nonlinear input and output relationships, use sequential training procedures and adapt themselves to the data. The learning process involves updating network architecture and connection weights so that a network can efficiently perform a specific classification / clustering task. The increasing popularity of neural network models to solve pattern recognition problems has been primarily due to their seemingly low dependence on domain-specific knowledge and due to the availability of efficient learning algorithm for practitioners to use.

Neural network also provides nonlinear algorithms for feature extraction and classification. In addition, existing feature extraction and classification algorithms can also be mapped on neural network architectures for efficient hardware implementation.

2.16 Summary

The objective of this Chapter has been to find out any alternative for the defect detection process in industrial applications. The Chapter is organized into two parts. The first part summarizes the available inspection systems and the various criteria pertaining to the visual world. These elements are the necessary guidelines to justify and understand the construction of new visual segmentation approach.

The second part (sections 2.10 to 2.13) has discussed a few approaches for defect pattern detection: from the simplest detection to the more accurate region-based approaches. In this context a new rule-based detection approach is proposed that provides various advantages over the other approaches. These two groups are used at different levels of analysis in the segmentation method proposed in Chapter 3.

This Chapter also focuses on the review of various defect pattern detection systems in the literature. Section 2.15 has discussed a few approaches for defect pattern detection: from the simplest detection to the more accurate region-based approaches. In this context the proposed rule-based approach provides various advantages over the other approaches. This Chapter also discussed the major methods (ADC/SSA/NRDD/Holography) for defect detection in semiconductor industry.

Chapter 3

A New Rule-Based Defect Pattern Detection

3.1 Introduction

Defect pattern segmentation is an attempt to perform an analysis of the visual information contained in the error image by describing what is present in the scene and where it is located. This description makes defect pattern segmentation a process very similar to human vision way of segmenting defect pattern. Both try to discover from images what is present in the world, and where it is located. Also it provides information to decide what actions to make to interact or react to the visual input.

Most defect pattern segmentation techniques that have been proposed in the past are designed to solve a specific problem. This strategy has been successful in proposing unsupervised image analysis solutions in several application domains such as inspection of industrial parts, medical imagery, traffic surveillance and autonomous robotics. The techniques have been tuned to deal with the specific constraints of the problem at hand. Thus, optimized solutions have been obtained both in terms of quality of the error image analysis and minimization of computational requirements.

However, such a specialized approach limits the generality of the techniques and the extension of the available systems to new applications. Modified approaches have been proposed to extend specific solutions to more broad application domains. Yet, the modified approaches usually need to be tuned manually in order to be adapted to a new scenario, and often the tuning process is complex and requires an expert user. In general, the adaptation of these algorithms to other applications requires considerable efforts and often a radical change of the adopted strategy.

This specialization factor increases the costs for the design of unsupervised defect pattern analysis systems. But above all, it limits the use of Computer Vision in all those applications where the visual data content and the purpose of the video analysis were not known a priori. This is the case for object-based video coding, video or image editing, autonomous robotics and advanced video surveillance.

Today, the improved computational performances of digital computers justify the research of more complex techniques with a higher degree of generality. It is more and more important to define modular solutions able to easily adapt to different scenarios and constraints, without the need of new designing efforts and minimizing the interaction with an expert user.

This Chapter proposed a new defect pattern segmentation scheme that has been conceived to achieve a high degree of flexibility. The adopted strategy takes into account some of the known properties of the human visual system (HVS), which remains the optimum image analysis system in several applications.

According to this method, it is proposed to decompose the general defect pattern segmentation problem in several levels. The lowest levels have a high degree of generality. Their main role is to simplify the input data and to extract the perceptually meaningful information. The proposed process is sufficiently general to be applied on a wide class of applications and input data. The highest levels combine the information generated by the lowest levels according to the specific criteria of the application at hand.

The objective of this Chapter is to define the low levels and the high levels rules and the associated algorithmic solutions that provide a perceptually equivalent simplification of the input data. The final goal of this thesis is to demonstrate, on different application domains, that it is possible to easily integrate different high level rules on the existing more general set in order to adapt the algorithm to new applications.

3.2 Defect Pattern Detection Process

The defect detection system works on the principle of comparing two images while identifying the faults. One image is called the Golden Master (GM), while the other image is the actual image under inspection. To have both images compared, they must be in the same position. The GM is stored in the referred lookup table, and the image for comparison is acquired on a regular basis for checking purposes. While acquiring images, there is always some displacement in the images. This can be due to many reasons like accuracy in synchronizing camera system to triggering pulses, sudden jerks, and vibration of the camera etc. This is assumed to happen and defect detection system should take these changes into consideration.

Defect pattern detection system can be separated into major steps as mentioned below.

- Golden Master (GM) Selection
- Matching of the GM and acquired Image
- Defect Pattern Segmentation Process

3.2.1 Golden Master (GM) Image Selection

The GM image selection is performed to identify the Region of Interest (ROI) in an image. Normally the ROI serves as valuable data used to match the GM with the actual print image, using correlation. Also using a single frame for comparison to the actual image would end up with more "defects", which may not be defects. There are many conditions, which can prevent two identical images acquired during two different points of times. Some of the critical factors, which play a major role in the selection of GM images, are illumination during the acquiring of images, accuracy in converting the analog video signal from CCD camera to digital data, external factors like heat, dust etc. This would reflect changes in the pixel values at given x, y positions for two identical over-lapped images. A one to one comparison of pixel values would only lead to unknown errors. A built-in tolerance system is used to overcome changes in the GM image when two identical images acquired at two different points of time. Hence, the GM image is further subdivided into two phases namely

- ROI detection in a input image
- GM image selection

3.2.1.1 ROI Detection

ROI detection for this detection system uses the principle of edge detection [73] in the images. The algorithm adopted utilizes the following two methods.

a) Method 1:

In this method at the given pixel position the algorithm finds the difference in intensities for red, green and blue pixels in x and y directions. The difference in intensity for a red pixel is given as,

$$\Delta r_x = I_r(x,y) - I_r(x+1,y) \tag{3.1}$$

$$\Delta r_y = I_r(x,y) - I_r(x,y+1) \tag{3.2}$$

where $I_r(x,y)$ is the pixel intensity in the R plane at a given x,y position. Δr_x is the x direction pixel intensity difference in the R plane and Δr_y is the y direction pixel intensity difference in the R plane. Similarly, the differences in intensity for green and blue pixels to the corresponding plane are given as.

$$\Delta g_x = I_g(x,y) - I_g(x+1,y) \tag{3.3}$$

$$\Delta g_y = I_g(x,y) - I_g(x,y+1) \tag{3.4}$$

$$\Delta b_x = I_b(x,y) - I_b(x+1,y) \tag{3.5}$$

$$\Delta b_y = I_b(x,y) - I_b(x,y+1) \tag{3.6}$$

The intensity in each color plane is given as,

$$R_{mag} = \Delta r_x + \Delta r_y \tag{3.7}$$

$$G_{mag} = \Delta g_x + \Delta g_y \tag{3.8}$$

$$B_{mag} = \Delta b_x + \Delta b_y \tag{3.9}$$

Then the total intensity of all color planes is expressed as,

$$Magnitude = R_{mag}{}^2 + G_{mag}{}^2 + B_{mag}{}^2 \tag{3.10}$$

where *Magnitude* is the total intensity of all color planes. If *Magnitude* is more than the threshold* (which is predefined) then that pixel is marked as an edge.

The input image and its corresponding edge detected image are shown in Figures 3.1 and 3.2 respectively. A windowing algorithm is then executed on this edge-detected image to find the ROI.

* The threshold value is detected dynamically based on the image characteristics.

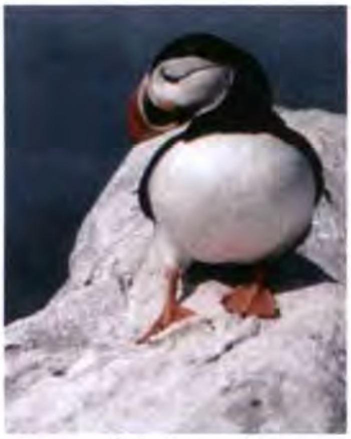


Figure 3.1: Input image.

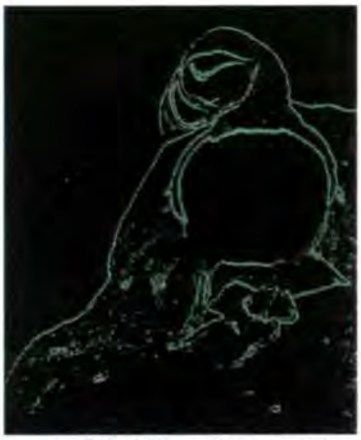


Figure 3.2: Edge detected image.

b) Method 2:

From the x and y resolution of the CCD image, the system uses a windowing algorithm by moving a predefined window (W) of smaller size in x and y directions of the input image as shown in Fig. 3.3.

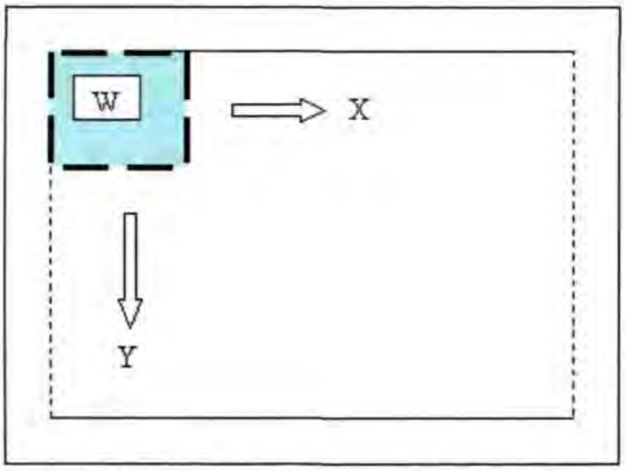


Figure 3.3: x and y directions of the input image.

The choice of the window size plays a critical role in the ROI selection. It is quite possible in an image with mixed images and regions a window selected should be larger in size, due to high intensity variations. In case of images with less contrast the window size should be smaller. Smaller window size also offers additional advantage in terms of computation. In the experiments I started on a normal sized window of 50×50 magnitude. For images of poor contrast the size was reduced to 25×25. The default ROI window selection in most of the automated systems depends on the quality of images and contrast parameters. Most of the time it is started with larger window size and reduce the size if required.

The predefined window movement is confined to an area, a little smaller than the original dimension of the acquired image, since it is not important about the edges. This movement is shown in Figure 3.3 as arrow mark in both x and y directions. The algorithm written is to have an overlapping movement in both x and y directions and these overlapping movement can be changed.

For each position on the window, for example from x_1, y_1 to x_2, y_2, calculate the gradient difference in both x and y directions using simple filters as shown in Fig. 3.4.

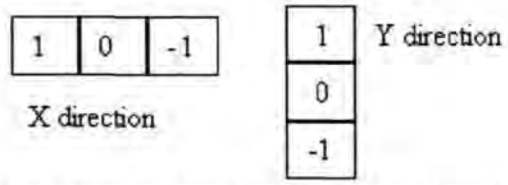


Figure 3.4: Filters used to calculate the gradient difference.

For a given window location of x_1, y_1 and x_2, y_2 the gradient difference in x direction is calculated using the loop equation and it is expressed as,

```
$for(i = y_1; i <= y_2; i++)$
{
$for(j = x_1; j <= x_2; j++)$
{
$dr_x = abs(I_r(j-1,i) - I_r(j+1,i));$
} }                                                    (3.11)
```

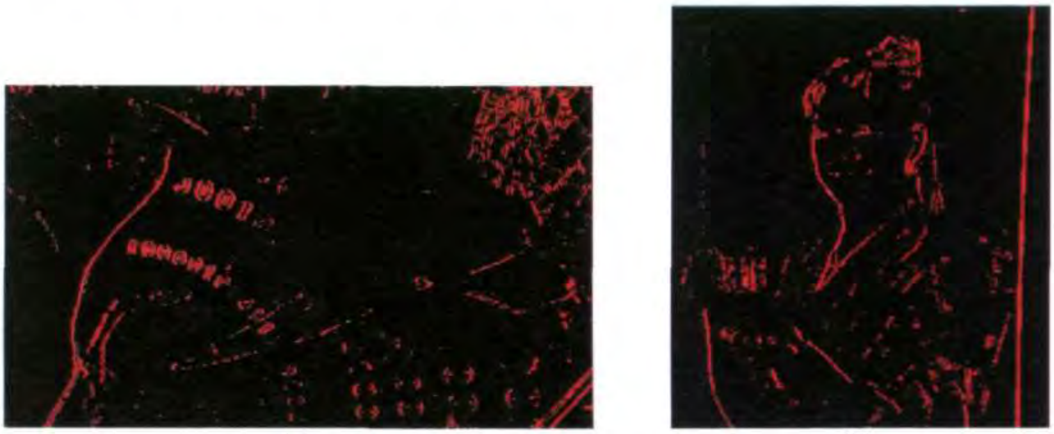
where $I_r(i-1, j)$ is the pixel intensity in the R plane at a given $(i-1, j)$ position and dr_x is the x direction pixel intensity difference in the R plane.

The same loop equation is repeated in the y direction for the given window location and equation (3.11) becomes,

```
$for(i = x_1; i <= x_2; i++)$
{
$for(j = y_1; j <= y_2; j++)$
{
$dr_y = abs(I_r(i, j-1) - I_r(i, j+1));$
}}                                                     (3.12)
```

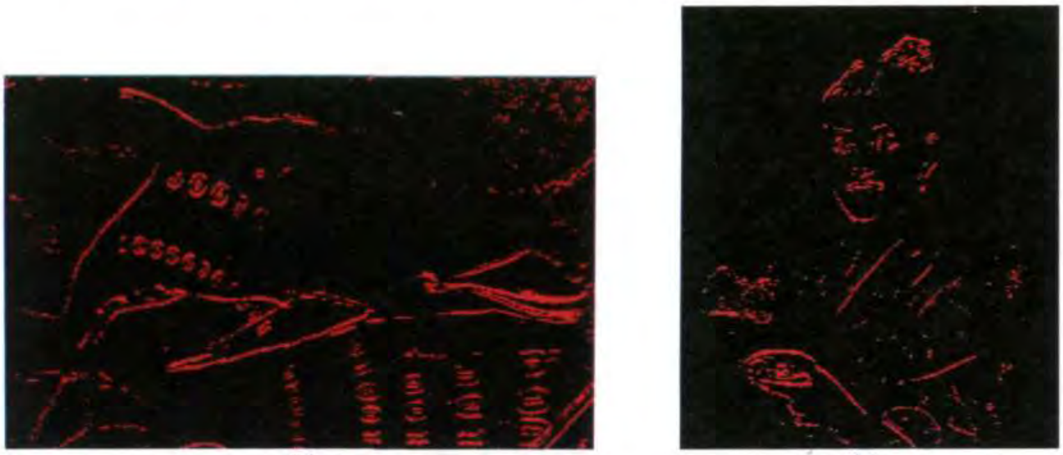
where $I_r(i, j-1)$ is the pixel intensity in the R plane at a given $(i, j-1)$ position and dr_y is the y direction pixel intensity difference in the R plane. The input image in Fig. 3.1 and its corresponding edge detected images using x and y direction filters are shown in Figures 3.5 and 3.6 respectively.

The intensities calculated in x and y directions for a window position in equations (3.11) and (3.12) are added together. The gradient difference is accumulated in both the directions, and this process is repeated for all window jumps. The window position, where the maximum gradient difference achieved is selected as ROI. The position where the maximum gradient difference achieved is stored for further analysis during defect identification.



(a) (b)

Figure 3.5: Edge detected images using the x direction filter.



(a) (b)

Figure 3.6: Edge detected images using the y direction filter.

c) ROI Analysis

In this thesis the methods adopted here depend heavily on the quality of the image. For this reason pre-analysis will be done before ROI detection. There are various possible combinations available for ROI selection.

- Single ROI area.
- Multiple ROI regions. In this, the image can be split into regions and ROI for these regions can be identified. This can facilitate better correlation to the actual image.

Also, the ROI detection algorithms are flexible enough to narrow the window dimension W, if the image contains less structural information. It is due to the fact that the factors explained in this section make the process of selecting GM image very critical.

3.2.1.2 GM Image Selection

The algorithm employed here builds an image made of intensity averages of multiple images namely I_{avg}. The averaging process implemented uses a set of 10 images for this purpose. The choice of 10 images is to test the algorithm implemented and it is not a hard coded value. Most of the time the number of images used for averaging process depends on many factors like lighting, image quality, frame grabber used, synchronization etc. The number of images required for averaging process is lesser for images with sharp contrast than the images of poor contrast.

In real world image processing the success and failure starts and ends with lighting. Lighting plays a very crucial role in image processing applications and particularly in defect detection systems. The algorithm also builds a tolerance image with respect to the averaged image. The reason why an average image as well as tolerance image was constructed is due to the following reason.

It is quite possible that single pixel or a group of pixel under study may be above or below the averaged image pixel or group of pixel intensity. In such situations a tolerance image would help us to clearly pinpoint whether pixel or pixels under study can be classified as errors or not. The tolerance images also help to prevent classifying areas as faults when there is fault in at the point of consideration. Other wise it may end with situations where there are faults detected when it does not exist. It is due to the above reason the tolerance image serves as GM image.

The tolerance image gives the maximum and minimum deviations of a pixel value namely $I_{\min}$ and $I_{\max}$ with respect to the average pixel. Also, the tolerance image in combination with the average image, serves as the GM image for further defect detection. The algorithm to build a tolerance image with respect to the average image is as follows:

Step 1: Let us acquire N (ten) images from the camera.

Step 2: Loop each pixel across the N (ten) images.

Step 3: For a given pixel position 5,5 accumulate the image intensities for the N images and find the average intensity value in each plane

$$I_{RSum} = I_{R1} + I_{R2} + \dots\dots + I_{R10} \tag{3.13}$$

where, I_{R1} is the R plane intensity of pixel (5, 5) in image 1 and so on.

$$I_{RAvg} = I_{RSun} / 10 \tag{3.14}$$

Step 4: Repeat the step 3 for the G and B Plane.

Step 5: The reference image pixel will be made from the averaged values of the color planes.

Step 6: In the averaging process calculate the standard deviation at the given pixel location from the values across the N images. It will form the tolerance values for the pixels in the reference image.

3.2.2 Matching of the GM and Images Using Subtractive Correlation

The next process is to match the GM image to the actual image because the actual image is not always in synchronous with the GM image. There is always some shift in the actual images. Before doing any comparison both the images should be overlapped properly. For this a correlation technique is used to overlap both GM and actual images. The following steps explain the overlapping process.

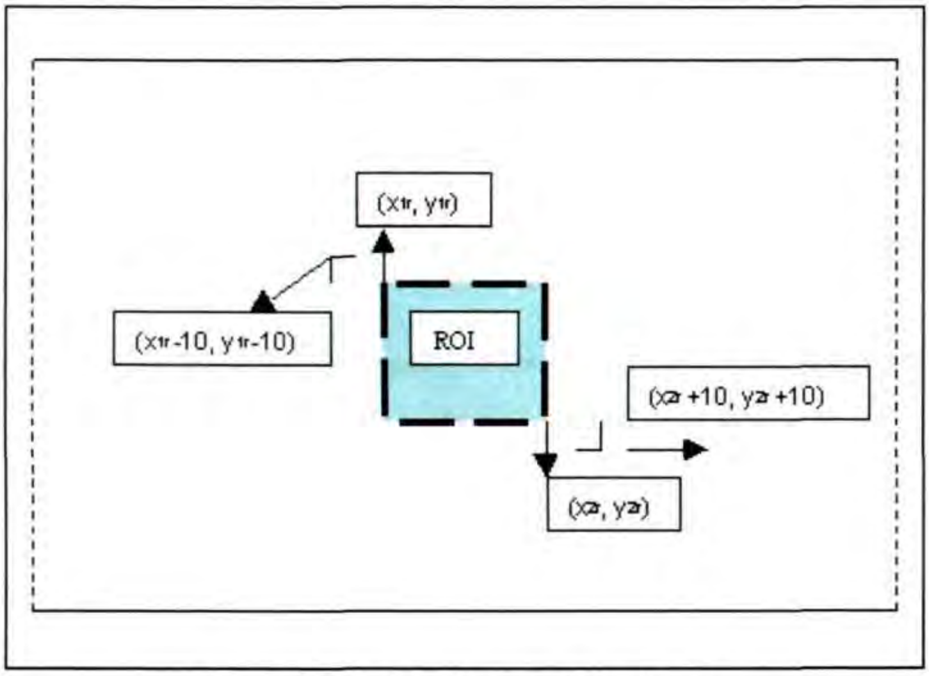


Figure 3.7: ROI in the actual live image.

Step 1: The ROI coordinates (x_{1r}, y_{1r}) and (x_{2r}, y_{2r}) are used for the analysis. The ROI window identified with the above coordinates is moved ± 10 pixels both in x and y directions in the actual live image I_{live} as shown in Fig. 3.7.

Note: The overlapping pixels selections of 10×10 in x, y direction were attempted based on the camera system and synchronization employed in the experiment. In the case of real-time print defect detection systems the camera movement in x direction is through the use of linear traverse and y direction is through the synchronization of the print cylinder rotations and trigger of flash. 100 % it is not possible to acquire the same image at the same position at all times. It is due to the above reason in correlation one has to bring this tolerance into consideration before attempting any image comparison. The x direction movement need not be same as the y direction since most print systems employ precise edge guiding. This prevents any wide move movement of print material in x directions. The correlation ROI window movement identical in x and y directions by n pixels can be altered for the reasons stated above. The choice of 10 pixels was able to widely cover huge synchronization problems. On highly fine tuned system the same can be reduced.

Step 2: For each position starting from $(x_{1r}-10, y_{1r}-10)$ the pixels within the window area are subtracted to the pixels in the ROI region of the GM image I_{avg}. For all window movements the pixels in the GM ROI would be the same and only the pixels in the actual image I_{live} would change for each window movement.

Step 3: The subtracted pixel values are accumulated for each window position and this step is repeated till ± 10 pixels movement in x and y directions.

Step 4: A typical loop for a given location x_1, y_1 and x_2, y_2 is calculated using correlation subtraction and it is expressed as,

```
$for(i = x_1, k = x_{1r}; i <= x_2; i++, k++)$
{
$for(j = y_1, i = y_{1r}; j <= y_2; j++, i++)$
{
$dr_{xy} = abs(I_{ref}(k,l) - I_{live}(i,j));$
} }                                                    (3.15)
```

where $I_{refr}(k,l)$ is the GM image pixel intensity in the R plane at a given (k,l) position. $I_{liver}(i,j)$ is the live image pixel intensity in the R plane at a given (i,j) position and dr_{xy} is the pixel intensity difference in the R plane in x and y directions.

In this process the window position where the pixel difference is minimum, identifies the position where the actual live image and GM image are in synchronization. The x and y coordinates of this position in the actual live image identified as x_1', y_1' are compared to the values x_{1r}, y_{1r} of the GM image to find the exact shift in terms of pixel values.

For example: let us assume that x_{1r}, y_{1r} and x_{2r}, y_{2r} for the ROI in the GM image are 50, 50 and 100, 100 assuming a window dimension of 50×50. Now the movement of a window in the actual live image would be 50 ± 10 in the x direction and 50 ± 10 in the y

direction i.e., in the x direction one has to start from 40 to 60 and in the y direction the same way. Assuming that if it achieves the minimum difference at position 45, 45 i.e., the image for comparison has been shifted 5 pixels left and 5 pixels up with respect to the GM image ROI position. This shift in x and y directions is called very critical defect detection. If comparing pixels at 10, 10 in the GM image then the corresponding pixels in the actual image for defect detection would be from position 5, 5. At certain instant the shift in actual image is not exactly in pixels, which means that sub pixel correlation has to be attempted to bring the images in synchronization.

3.2.3 Defect Pattern Segmentation Process

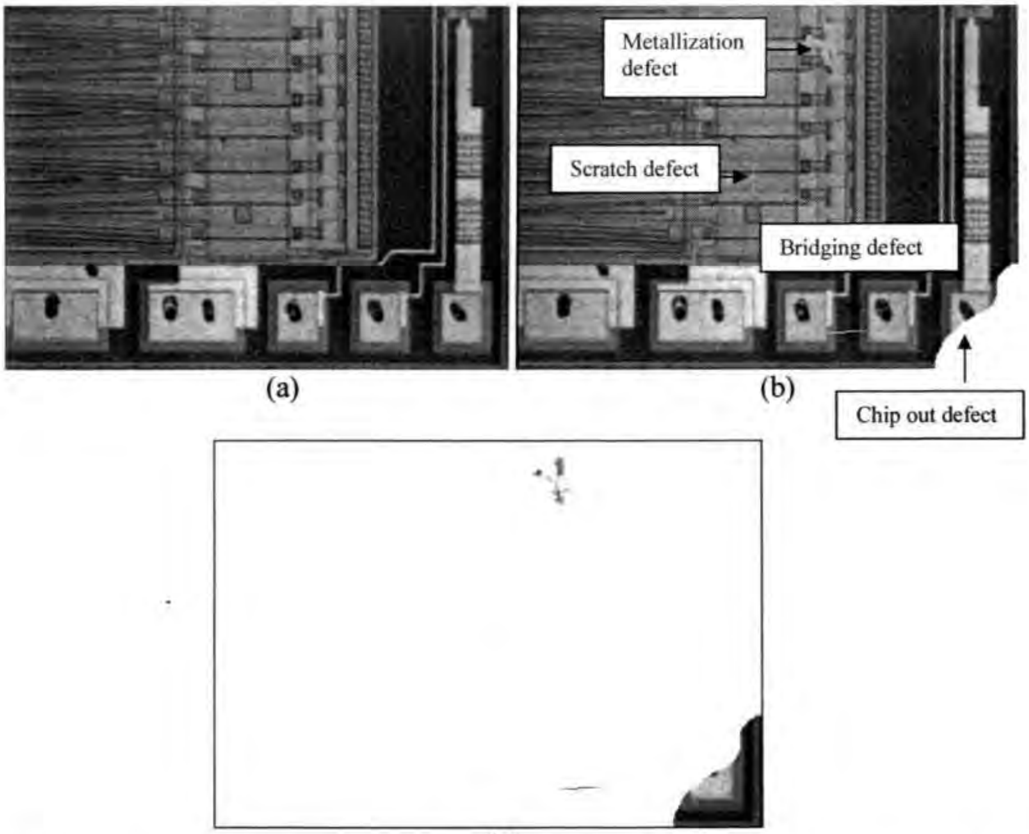


Figure 3.8: (a) Golden Master (GM) die image, (b) inspected die image, and (c) error image.

The Golden Master (GM) die image in Fig. 3.8 (a) is matched to the inspected die image in Fig. 3.8 (b), and the resultant image is the error image as shown in Fig. 3.8 (c). Figure 3.8 (c) contains defects such as chip-out, bridging and scratch. Each pixel of the frame is described with a number of descriptors or features.

The proposed hierarchical defect pattern segmentation approach is summarized in Fig. 3.9. In this scheme the input is represented as a frame of error image, such as the one obtained in Fig. 3.8 (c). The first level or diagnostic level performs an automatic classification of the pixels in the error image according to their properties. This level generates regions with homogeneous properties. These regions represent perceptually non-divisible areas or primitives of the error image content. They are used in the following levels as the basic bricks to build more complex structures such as objects.

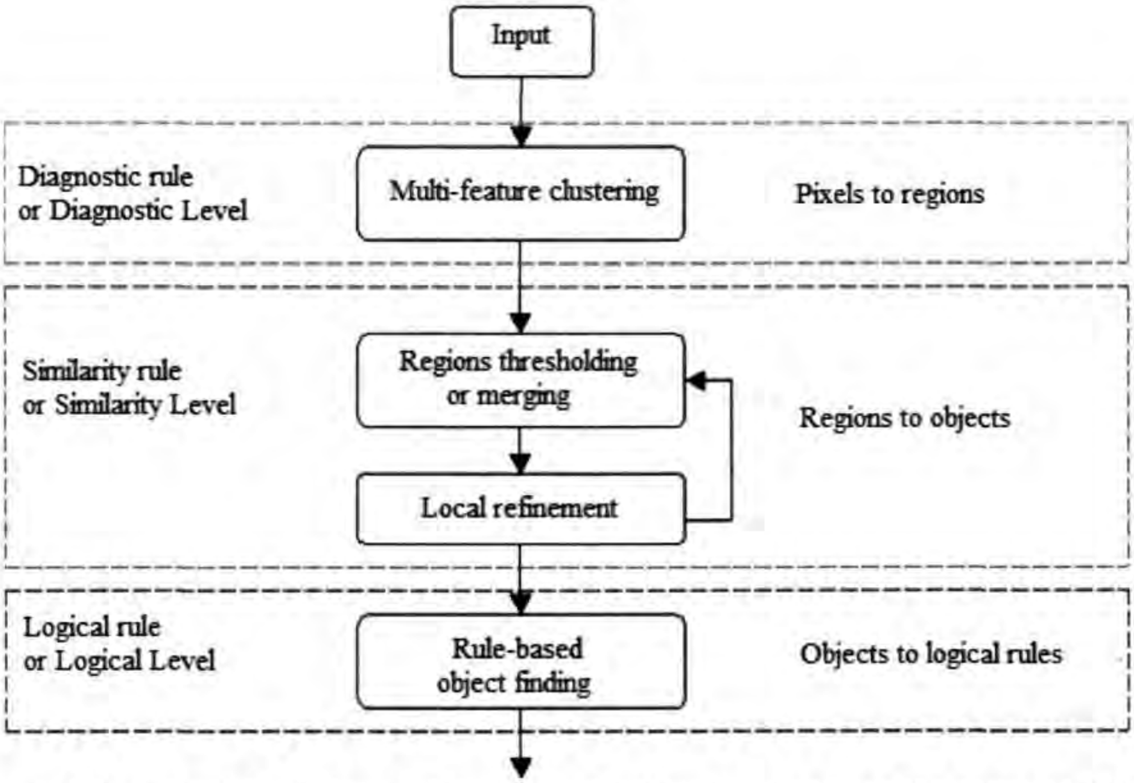


Figure 3.9: A simplified scheme of the hierarchical levels composing the proposed defect segmentation algorithm.

The second level or similarity level that guide to a region merging process. The most similar regions are merged together according to similarity rule to progressively form objects. In this process the merging step is followed by a refinement step using priority rules. This improves the contour accuracy and the measure of the mutual similarity between the regions. This measure is the core of the merging process.

Finally the last level or logical level corresponds to a rule-based object finding procedure. Its results are used to declare the analysis of successive frames of the error image. An object, which is found along the error image, is referred to as the defect pattern in a particular inspected die image.

3.3 Similarities with the clinical diagnosis

In order to explain in more detail the levels that compose the proposed scheme, a comparison with the clinical and visual language is established. This comparison has the main objective to provide an intuitive way for understanding the technique. It intends also to highlight the generality of the approach.

A first analogy can be defined between the symptoms of the patient and the pixels of an error image. The symptoms of the patient are the atomic elements of the clinical diagnosis. If considered alone, a simple symptom has little meaning. Yet, when grouped together, with scan and test results constitute a better understanding of the disease. Not all the symptoms represent a valid disease. Nevertheless, all valid diagnosis of disease obeys to precise rule referred to as diagnostic rule.

In visual language the atomic elements are the pixels of the error image. These may be considered in our comparison with the clinical language, the visual symptoms. Each pixel is associated with a number of features, for example the lightness, hue and chroma (LHC) predictors. These features are used to recognize the different characteristics of the pixels: thus to distinguish one visual symptom from the others. In analogy with a simple symptom of the disease, a single pixel may not provide us with much information. Instead, a group of pixels may represent a region or an object in the scene. In our comparison with the clinical language, these regions may be referred to as visual disease.

If pixels are grouped together randomly, it may not find significant regions. So it is necessary to define the diagnostic rules of the visual language to group pixels in valid regions. In clinical practice, scan and test results are compared with template to conclude the nature of the disease. The rules that normalize this process define the similarity of the clinical language. In analogy with this process, the visual screening (regions) should be grouped together according to similarity rules in order to form visual test results (objects) with a higher semantical content.

Table 3.1: The corresponding elements between the clinical and visual languages.

Clinical Language	**Visual Language**
Symptoms	Defective pixels
Screening	Region
Test Results	Set of objects
Decision	Object

Table 3.1 summarizes the corresponding elements of the clinical and visual languages. This comparison highlights four levels: the symptom, the diagnostic, the similarity and the logic. Similarly, the defect image segmentation scheme proposed in this work is organized in four levels. The schematic representation of its structure is displayed in Fig. 3.10.

Note that each level finds correspondence with a precise element of the clinical practice. It is sometimes convenient to limit the analysis process on a subset of the input data. According to the application, it will focus the analysis on the central area or on the peripheral areas of the error image. This focus process has the effect of discarding less important information, improving the performances of the analysis at least from a computational point of view. Note that this process is similar to the behavior of the clinical practice, which is able to focus the interpretation process on a limited area of the visual field.

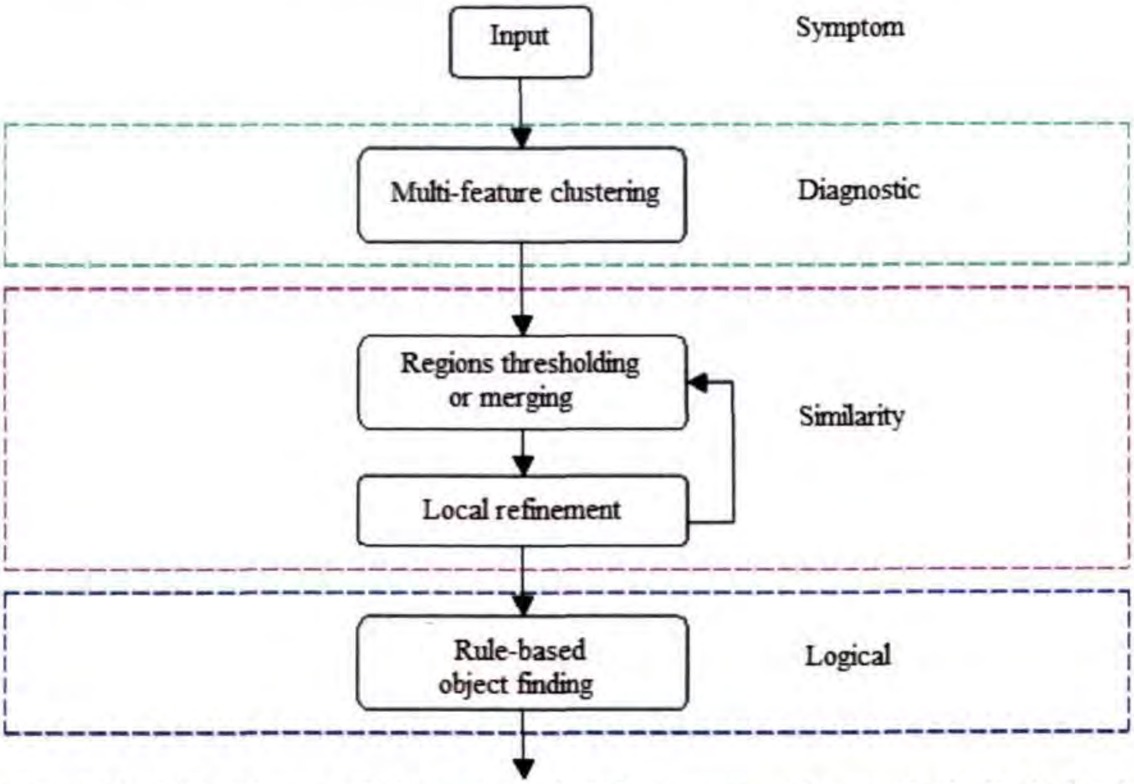


Figure 3.10: The scheme of the proposed defect segmentation method. Each level is associated to the corresponding element in the clinical practice.

The first level of the algorithm extracts the absolute error image according to a specific visual symptom (Fig 3.8 (c)). The second level of the algorithm classifies the pixels into uniform classes. Uniformity is defined according to the pixels properties of the error image. Techniques have been discussed in the following sections that may perform this task.

The third level of the technique analyses the region generated by the second level and decides, according to certain diagnostic rules, when some adjacent regions need to be merged together. This linking process is followed by a recursive local refinement procedure. This step locally refines those regions whose global properties have changed after the merging process. At this level, diagnostic rules may be defined. Those derived from the properties of the human visual system are application independent and will be discussed in Section 3.5. The result of this level is the link of perceptually uniform regions into objects. This logic correspondence between the objects of the visual content improves the analysis of the visual information. The proposed correspondence technique is described in Section 3.6.

The proposed approach aims at providing a general framework where it is easy to integrate high-level rules defined according to the application. The lowest levels, designed to simulate the human visual perception, would remain the same. The idea is to offer a sufficiently general approach to be easily adapted to each new application. This approach offers the ambitious possibility to build a set of high level rules that could be evaluated and selected automatically according to the input data and the system priorities.

3.4 The Symptom

The purpose of a symptom is to code information. The relevant aspect in this process is that what information should be coded. In visual languages, one of the most important information to be coded is certainly the defect information. The model [74] that is used to recover the visual symptom from the test image is shown in Fig 3.11. In the definition of a visual symptom, a low level analysis of the input error data is necessary. This is the reason why it is preferred to evaluate the simpler pattern matching models in Section 3.2.

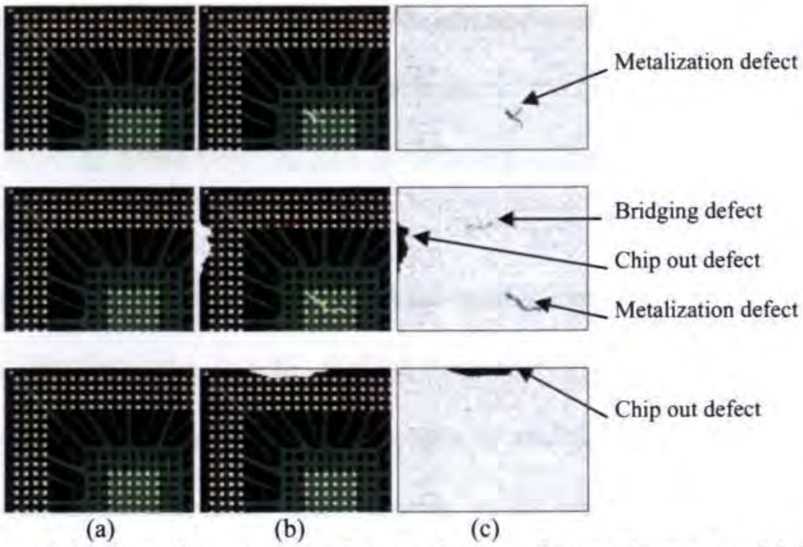


Figure 3.11: Real data used for experiments (a) reference image, (b) test image, and (c) visual error data or visual symptom.

3.5 Diagnostic Rule or Diagnostic Level

In clinical practice, the symptoms of the disease are screened according to specific diagnostic rules. The correct screening procedure encodes more information than the simple symptoms. Similarly, in visual language, the information is not perceived in terms of isolated pixels, but in terms of areas or group of pixels that appear to be perceptually uniform. Diagnostic rule proposed in this thesis will control the association of pixels in perceptually uniform regions.

3.5.1 Feature classification

In the previous sections it has been discussed how it is possible to describe the visual information in terms of perceptive features, e.g. features that predict perceptive mechanisms of human vision. The result of this description is that each pixel in a digital image is associated with a number of values characterizing its color, and texture properties. These values are useful in several computer vision and pattern recognition problems. However, the amount of data generated in extracting all these features is high. Processing all this information becomes heavy even for the computational power of modern processors.

Moreover, since natural images are well approximated by two-dimensional piece-wise functions, most of this information is likely to be redundant and could be better represented by grouping all the pixels that are described by similar features into homogeneous classes. In this way, the description of the properties of a class is sufficient to characterize the properties of all the pixels associated to it.

This section describes different techniques that are used to classify general features in homogeneous sets. In this thesis, the specific goal of these techniques is to obtain from a digital image, a piece-wise constant image (segmentation mask), where all the pixels belonging to the same group (hereafter also referred to as cluster) share the same characteristics. This process reveals the organization of patterns in the feature space and helps in deriving useful conclusions about their similarities and differences. It represents the first step towards an interpretation of the image content. Moreover it reduces the computational efforts required for further analysis.

3.5.2 Fundamental issues in feature classification

Feature classification techniques benefit today from many years of research performed on different domains. Many efficient techniques and algorithms already exist that have been specialized to deal with the constraints of specific problems. However, engineering an efficient clustering procedure for a specific application remains a complex task. It requires the attentive analysis of several issues. The following subsections will discuss the major issues that were taken into account in this work and that should be carefully evaluated whenever a feature classification problems need to be solved.

Features have to be selected so as to encode all the useful information concerning the task of interest. In the framework of computer vision and pattern recognition, many features are available. Some have already been discussed in Chapter 2, but according to the acquisition system or the application scenario, other features may be preferred. As a general comment, the choice and the number of the features to be used in a classification procedure vary according to the application. Feature selection is still an important topic of research [76, 77], and merits the highest attention in the definition of any effective clustering scheme.

3.5.3 Proximity measures

Once the features have been selected, it is necessary to measure how similar or dissimilar two feature vectors are. This is possible by defining a proximity measure. The definition of a proximity measure, in the example of Fig. 3.12, indicates if the point A is closer to the point O than the point B. Moreover, if a threshold is defined, it is possible to assess if the point B is sufficiently similar to the point O. In the example of Fig. 3.12, a threshold may be represented by the circle around the point O with radius equal to the threshold t. All the points inside the circle will obviously have a Euclidean distance to O smaller than t. In the case N features are defined, an N-dimensional feature space is obtained that may be difficult to represent graphically.

Proximity measures are not computed only between single points as in the example of Fig. 3.12. It is often necessary to compute the proximity between a point and a set of points or between two sets of points. Note that the choice of the proximity measure should take into account the nature of the selected features.

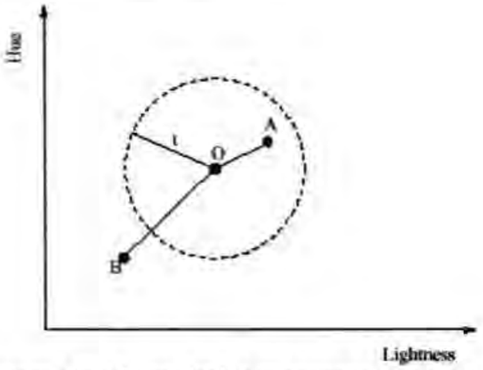


Figure 3.12: A graphical example of proximity measures in a two-dimensional feature space.

One of the most critical step in any classification technique is the definition of the right proximity measure to be used for comparing the elements of the input data I. Proximity measures express the similarity or the dissimilarity between different entities. In particular, for an image clustering procedure, it is important to measure three kinds of proximities:

- Point-to-Point: the proximity between two vectors of features describing the properties of two points (pixels) in the image.
- Point to Set: the proximity between a point and a set of points.
- Set-to-Set: the proximity between two sets of points.

In order to correctly measure the proximity between two entities as listed above, the proximity measures need to take into account the nature of the features describing each entity. This is important so as to give the correct weight to each element of the feature vector. Also, different proximity measures may be chosen according to the properties of the desired cluster. There are many definitions of clusters exist.

The choice of the proximity measure, implicitly defines the properties of the obtained cluster. Finally, it is important to take into account the range of each feature. A feature may assume values from a continuous range or from a finite discrete set. When the finite discrete set has only two values, the feature is referenced to as binary feature. When features with different ranges need to be combined, it is often necessary to perform a normalization procedure.

3.5.4 Definition of clustering criterion

According to the distribution of the data in the feature space and the type of clusters that are expected to underlie the data set, different clustering criteria may be defined. In general these criteria are implicitly defined by the adopted proximity measure. However, the definition of additional cost functions or other rules may guide the procedure so as to obtain compact clusters instead of linear clusters.

3.5.4.1 A mathematical definition of clustering

In this section, a mathematical definition of clustering is proposed together with a block diagram describing the information flow of a general image clustering procedure from the input data to the final classification. The introduced mathematical notations and the scheme will be referenced in the following sections where state of the art clustering procedures are discussed. The discussion aims at highlighting the differences between the possible approaches to image classification. It will justify the choices proposed by the author in this Chapter.

Clustering has been used in the past in several application domains. However, in this work, our concern is limited to computer vision and pattern recognition problems. In these domains the input data is represented by a digital image. This is defined as a $(H \times W)$ grid of pixels, where H and W represent the height and the width of the grid.

When the input digital image undergoes a feature extraction procedure the result is that each pixel of the input image is associated to an N-dimensional feature vector. Here, N represents the number of features extracted. As it is described by the scheme of Fig. 3.13, the clustering procedure is applied on the N-dimensional data generated by the feature extraction procedure and not directly on the input image.

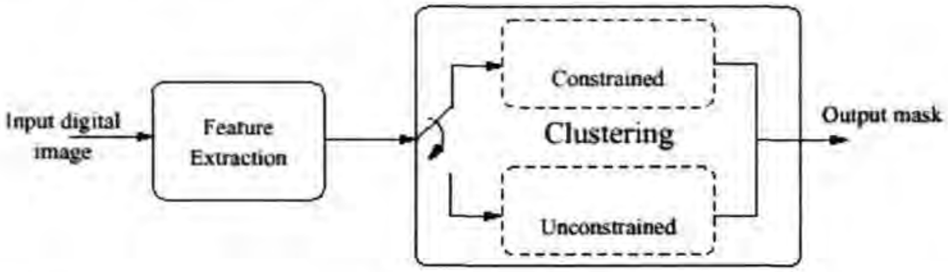


Figure 3.13: Block diagram describing the information flow of a general image clustering procedure from the input data to the final classification [76].

Note that the data resulted from the feature extraction procedure belong to an N-dimensional feature space, but at the same time it is constrained to the $(H \times W)$ grid of the input image. For this reason, the input to an image classification system is supposed in the form of a $(H \times W)$ matrix I, of N-dimensional feature vectors.

In this thesis, an M-clustering of I is defined as the partition of the $(H \times W)$ elements of I into M sets (clusters), $C_1,.........,C_M$. Each cluster contains vectors $\vec{P}$ belonging to I, so that the following conditions are met:

$$\begin{gathered} C_i \neq \theta, i = 1, \ldots\ldots, M \\ \cup_{i=1}^{M} C_i = 1 \\ C_i \cap C_j = \theta, i \neq j, i, j = 1, \ldots\ldots, M \\ S(\vec{p}_k, C_i) > S(\vec{p}_k, C_j), \forall \vec{p}_k \in C_i, i \neq j, i, j = 1, \ldots., M \end{gathered} \tag{3.16}$$

where $S(\vec{p}_k, C_i)$ represents the similarity between $\vec{p}_k$ and the cluster C_i. The assessment in equation (3.16) says that the vectors in C_i are more similar to each other and less similar to the feature vectors of the other clusters. The exact definition of similarity depends on the chosen proximity measure.

The final output of the clustering process is represented by a $(H \times W)$ matrix O of integer values or labels. All the elements of I have a corresponding element in O since the two matrices have the same dimensions. In O, all the elements belonging to the same cluster are labeled with the same value. Several techniques have been proposed to perform a clustering of a digital image. In this work it is defined to have two categories of methods: the constrained clustering techniques and the unconstrained clustering techniques. The first group of methods performs clustering in the image coordinate domain. This means that all these techniques directly take into account the relative positions of the feature vectors in the matrix I in order to guide the classification procedure. Very well known techniques belonging to this group are region growing [78-89], split & merge [90, 91, 92, 93] and pyramidal approaches [94-97]. Their principles and properties are presented in Chapter 4.

The second group of methods performs clustering directly in the feature space domain. In this case, the input matrix I is converted in a non-ordered set of N-dimensional feature vectors. The strategies of these techniques do not take into account the spatial position of the feature vectors unless this has been encoded in the feature vectors. In this sense, the unconstrained clustering techniques are very general approaches and they are not limited to image clustering application. They have been tested and improved over many years of research in a wide range of different domains. Note that even if the spatial information is not directly taken into account during the clustering procedure, each feature vector remains related to its position in the input image. This is necessary in order to convert the clustering results in the feature space domain to the final matrix O. Well known techniques in this group are the very simple histogram thresholding techniques [98-103] and the very well known partitional clustering procedures such as Isodata [104-110] and Forgy [111] and Fuzzy [112, 113]. A review of the most known techniques belonging to this group is proposed in the following Section.

3.5.4.2 Unconstrained clustering

Previous sections discussed those methods that directly take into account the position information during the classification process. The classification strategies adopted by these methods are performed in the image coordinate system and are guided by the feature values associated with each point in the image.

In this section, techniques that perform a classification of the input data I directly in the feature space domain are discussed. This means that the matrix of feature vectors I is now interpreted as a non-ordered set of feature vectors. Each feature vector defines a point in a multidimensional space. The clustering procedure tries to define a certain number of classes in this feature space and to associate all the points in the input data to one of these classes.

From an historical point of view, clustering in the feature space domain is known in several application contexts: from medicine to economics [75]. It has certainly benefited today from a larger research effort compared to the clustering techniques described in the previous section that are mainly limited to computer vision and pattern recognition.

The implementation performances of the two approaches (constrained and unconstrained) are also different. In general, clustering in the feature space domain is a simpler task compared to clustering in the image coordinate domain. This is due to the fact that there are no constraints due to spatial structures: all the data may be analyzed at the same time and not sequentially, as it is necessary for example in the Region Growing techniques.

Other main differences between the two approaches concern the initialization of the procedure and the properties of the final results. These differences are discussed in the following. In order to initialize a clustering procedure in the feature space domain, it is necessary to provide a certain number of points in the feature space. It is not necessary that these points exist in the input data I. In general the initialization procedure does not require a complex analysis of the input data. Many efficient solutions exist that are based on histogram analysis from which it is possible to extract information such as how many clusters should be defined and what are their best representative. It has already been underlined that the clustering results depend on the choice of the features, the proximity measures and the classification strategy. If the clustering procedure is performed in the image coordinate system, the result will be directly available in these coordinates.

In the following subsections, it will discuss in more details three different approaches to clustering in the feature space domain. These are amplitude thresholding, partitional algorithms and sequential algorithms.

a) Amplitude Thresholding

Amplitude thresholding techniques rely on the hypothesis that the amplitude of their features can distinguish the different elements in the scene. In its simplest form (two-level thresholding of a single feature), it is sufficient to define a threshold and to classify all the points whose feature value is above the threshold in a class, whereas all the points whose feature value is below the threshold are classified in a different class.

This approach may appear particularly simple and thus strongly limited. This is true in most applications where real scenes need to be analyzed. However, there are many applications where such a simple approach represents a very good compromise between quality of the results and complexity of the technique. For example, binary data such as printed documents or medical images such as X-rays often satisfy the hypothesis on which this approach is based. Moreover, techniques exist that extend the simple two-level thresholding on a single feature to more general multilevel thresholding on several features. These techniques achieve a better analysis on more general input data.

One of the most important steps for amplitude thresholding techniques is the definition of a correct threshold. Often this is fixed by analyzing the histogram of the input feature values. Histograms show how many times a certain feature value occurs in the input data. When the histogram is bimodal, e.g. it contains two local maxima, and then it is possible to set the threshold according to the position of the minima between the two maxima. This choice will automatically extract the bimodal statistics of the input data. Alternatively, if there have some hypothesis on the size of the desired class, it is possible to fix the threshold so as to obtain the desired number of pixels above or below the threshold.

In [114], other approaches used to fix an efficient threshold are discussed. However, in most situations histograms do not appear to be bimodal. In this case a recursive approach is adopted. The histogram is first divided in two parts: high values and low values. Then the histogram of the two parts is generated. If a bimodal histogram is obtained, then a threshold is fixed by applying one of the methods discussed above. Otherwise another division in high and low values is performed. The process stops when the obtained histogram is mono-modal or contains less then a minimum number of points.

A natural extension to multi-feature amplitude thresholding, simply consists in dynamically choosing on which feature it is better to perform the thresholding. Ad hoc solutions exist as reported by Pratt in [115]. However, they require a narrow interaction with the user to obtain the desired results. An alternative to these techniques consists in defining a multidimensional histogram by combining more features. This solution requires the definition of complex strategies in order to choose the thresholds according to the detected local maxima and minima.

b) Partitional algorithms

Partitional algorithms provide a single clustering result by sequentially scanning all the pixels of the image. Generally these techniques work as follow. In the first phase (Initialization step), a certain number of clusters are defined. Each cluster needs to be described with a feature vector. Then each pixel of the image is associated to the most similar cluster (Association step). The concept of similarity here depends on the features used to describe each pixel and on the proximity measure adopted. Once all the pixels have been associated to a cluster, the cluster properties are updated according to the pixels they contain (Cluster update step). Again the pixels are sequentially compared with the updated clusters.

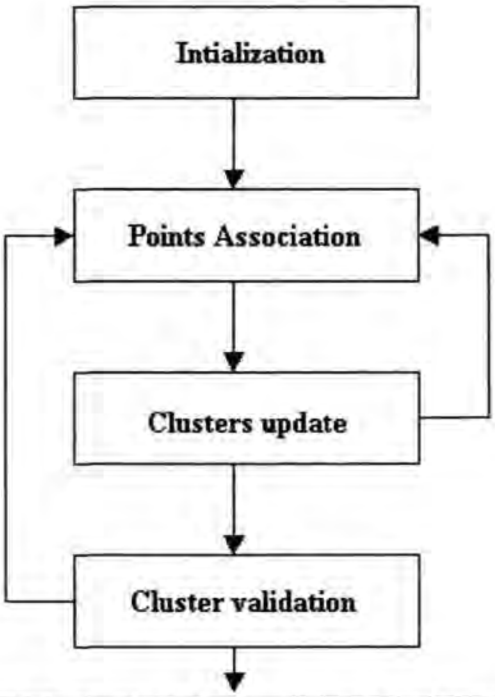


Figure 3.14: Partitional algorithms. This scheme summarizes the steps of a general partitional algorithm.

The process continues until a maximum number of iterations have been performed, or no more changes happen in the clusters. In many algorithms a second phase exists in order to merge similar clusters or to split clusters that are not sufficiently homogeneous (Clusters validation step). Again the process can start from the beginning. A general scheme of this procedure is provided in Fig. 3.14.

Very well known algorithms belonging to this category are the Isodata (also known as k-Means or c-Means algorithm) and the Fuzzy algorithm. Isodata is a particular case of the general scheme described in Fig. 3.14. The proximity measure between a point and a cluster is defined as the Euclidean distance between the point and the representative of the cluster. The representative is defined as the mean vector of all the vectors belonging to the cluster.

c) Hierarchical algorithms

Hierarchical algorithms provide, for a given image, a sequence (hierarchy) of clustering results, with a decreasing number of clusters (divisive hierarchical algorithms) or increasing number of clusters (agglomerative hierarchical algorithms). As for the partitional algorithms several approaches exist that might be classified in this category. In the following, we present the basic ideas of these approaches.

Agglomerative algorithms may be seen as a generalization of Region Merging and Pyramidal strategies. The difference is that the position information of the points in the image is not directly taken into account in the clustering process. Let l be the current level of the hierarchy, when $l=0$ each point in the input data represents a single cluster. In order to define the level $l+1$, all the possible pairs of clusters in the current level are compared by using a proximity function. The two clusters that maximize the proximity are merged in a single cluster. The procedure continues until all vectors lie in a single cluster.

It is soon clear that agglomerative algorithms are computationally very demanding procedures: their complexity is proportional to N^3 [116] and also depends on the complexity of the proximity measure. Moreover, their nesting property excludes the possibility for recovering from a poor clustering occurred in earlier levels.

However, efficient implementation based on matrix theory and graph theory exists that reduce the complexity of these strategies [116]. Divisive algorithms follow a different approach. The input data is considered belonging to a unique cluster at the level $l = 0$. In order to generate the level $l + 1$ all the possible partitions in two parts of each cluster defined at level l are evaluated. Note that if N_C is the cardinality of a cluster C, there will be $2^{N_C - 1} - 1$ partitions of C in two clusters. Again among these partitions a choice is performed according to a specified criterion. The final result will contain N clusters: one for each element of I.

In comparison to agglomerative techniques, divisive techniques are computationally even more demanding. The necessary compromise for practical applications requires to limit the search of all the possible partitions, to those that are more reasonable. However, this compromise requires the introduction of a priori choices. These may be valid for a specific application, but may fail in a more general context.

Both agglomerative and divisive procedures provide a hierarchy of possible clustering results with different number of clusters. It is then necessary to choose what is the level that better satisfies the application requirements. Many solutions are available for this purpose. One of the simplest ways consists in fixing the desired number of clusters: this implicitly defines a level of the hierarchy. More complex techniques take into account the lifetime of single clusters and decide the level according to the presence of stable configurations. It is in general complex to define valid criteria for a broad class of applications.

3.5.5 Clustering algorithm

According to the diagnostic rule, the visual error data should be grouped in areas with uniform visual features. This is the hypothesis for which a multi-feature clustering technique is applied to group in homogeneous classes of the error elements of the input data. Several multi-feature classification techniques are available in the literature. In this context, the partitional clustering algorithms described in the form of Isodata [117] were adopted in this thesis. Flexibility is an important characteristic for our approach since it is necessary to combine both local and global analysis of the input data to emulate the principles of perceptual organization.

3.5.5.1 Initialization procedure

The proposed initialization technique automatically defines the number of initial classes and their properties according to the distribution of the input data. It evaluates at the same time an arbitrary number of input features and can be successfully extended to other partitional algorithms.

- Initialization: Choose arbitrary initial estimates of the cluster representatives $C_j(0)$ with $j = 1,.........., M$.
- Repeat until no change in C_j's occurs between two successive iterations or a maximum number of iterations is to be performed.

3.5.5.2 Association step of the algorithm

According to the proposed clustering scheme, once the initialization step is performed, the association step takes place. The main purpose of this step is to associate all the input data points to the most similar cluster representative. However, the association step is also responsible for verifying if new cluster representatives should be added to the existing set. In order to perform these tasks, the first issue is the definition of a distance measure between a feature point, $\bar{p}$, and a cluster representative C.

In this work it is proposed to adopt proximity measure, the distance defined by the weighted sum of distances:

$$D(\bar{p},C)=\frac{1}{T}\sum_{i=1}^{T}W_i D_i(\bar{p},C) \tag{3.17}$$

Here, $D(\bar{p},C)$ is a normalized linear combination of T distance measures, $D(\bar{p},C)$. The functions D, D_i, W_i are numbers belonging to the cluster C of the vector element $\bar{p}$. Each of these measures describes with real values between zero and one the distance for a specific group of features. The weighting coefficients W_i are responsible for associating an absolute weight for each group of features contributing to the final distance, $D(\bar{p},C)$. The weights values are real values between zero and one. Equation (3.17) is a distance measure generalized for an arbitrary number T of groups of features. In the experiments, the number is limited to four groups. It is described as the low-level chip out, scratch, metallization and bridging. Thus, in our experiments, Equation (3.17) becomes:

$$D(\bar{p},C)=\frac{1}{4}\left(W_C D_C(\bar{p},C)+W_S D_S(\bar{p},C)+W_M D_M(\bar{p},C)+W_B D_B(\bar{p},C)\right) \tag{3.18}$$

where the distances for each group of features are normalized to real values between $[0,.........,1]$ according to the following expression i.e., for example the distance measures for chip out is given as

$$D_C(\bar{p},C)=\frac{|f_C(\bar{p})-f_C(C)|}{500} \tag{3.19}$$

the range of each features are limited to 500 for practical reason (It is assumed that the no of pixels for each defect features in an error image never exceed 500 pixels). Note that each group of feature is associated with a relative weight value: (W_C, W_S, W_M and W_B). These are the only features used in the initialization process. During initialization the weight associated to these features is set to zero. However, this initial configuration changes while the clustering takes place.

The next step is in the prediction of the reliability of each group of features. A dynamic weighting system is proposed to control the feature weights. Its role is to define the set of weight coefficients appearing in equation (3.18) for each cluster C_j: ($W_C(j), W_S(j), W_M(j)$ and $W_B(j)$). The idea behind the choice of these weighting coefficients is simple. If a cluster C_j is described by a uniform color, the variance of the color features computed for all the feature points associated to C_j will be low. In this case it is assumed that C_j is well characterized by that color property for which it is uniform. Thus, the weight associated to the color features should be high. On the contrary, if the cluster has not a uniform color, the variance of the color features will be high. In this case it is assumed that color is not the feature that makes the cluster C_j uniform. By decreasing the value of the associated weighting factor, it can reduce the influence of the color information in the definition of C_j.

3.5.5.3 Update step of the algorithm

The results of the association step is that each feature point $\vec{p}_i$ in I is now associated to a label l_j corresponding to one of the cluster representatives C_j with $j = 1,........,M$. If it is represented in the image coordinate system the label values associated to each feature point, then obtain a $(H \times W)$ matrix O of labels. This representation puts in correspondence each pixel of the input image with the most similar cluster. The main role of the update step is to use the label information in order to update the feature properties of each cluster. This is obtained in computing the mean properties of all the points belonging to the same cluster. The result of this update is that each cluster C_j is associated to a new feature vector. Each element of this feature vector is computed as the mean value of the corresponding element in the feature vectors $\vec{p}_i$ associated to C_j.

The update of the input feature points is certainly computationally expensive. Moreover, new similarity relationships are generated between feature points and existing clusters that need new iterations in order to find stable convergences. For this reasons it is important to update the feature values only when it is known that they are not correctly defined. In the experiments the update is performed first at the beginning of the clustering. A second update is performed after the convergence to a stable classification has been obtained in order to verify the consistence of the updated features.

```
Association Step

For all the input data points $\vec{p}_i \in I$ with $i = 1,.........,P$.
    ->Determine the closest representative, say $C_j$ for $\vec{p}_i$.
    ->Determine the mean distance to all the representatives, $\overline{D}$.
    If the distance between $C_j$ and $\vec{p}_i$ is close to $\overline{D}$.
        Then $\vec{p}_i$ is a new cluster representative.
        M is increased by 1.
        The label $l_i$ of $\vec{p}_i$ is set to M.
    Else set the label $l_i$ of $\vec{p}_i$ to $j$.
End For

Update Step

For all defined $C_j$ with $j = 1,..........,M$.
    ->Update $C_j$ according to the feature properties of the data points labeled with $j$.
    ->Update $W_C(j), W_S(j), W_M(j)$ and $W_B(j)$ according to the feature properties of the
      data points labeled with $j$.
End For
```

Figure 3.15: The clustering algorithm adopted in this work.

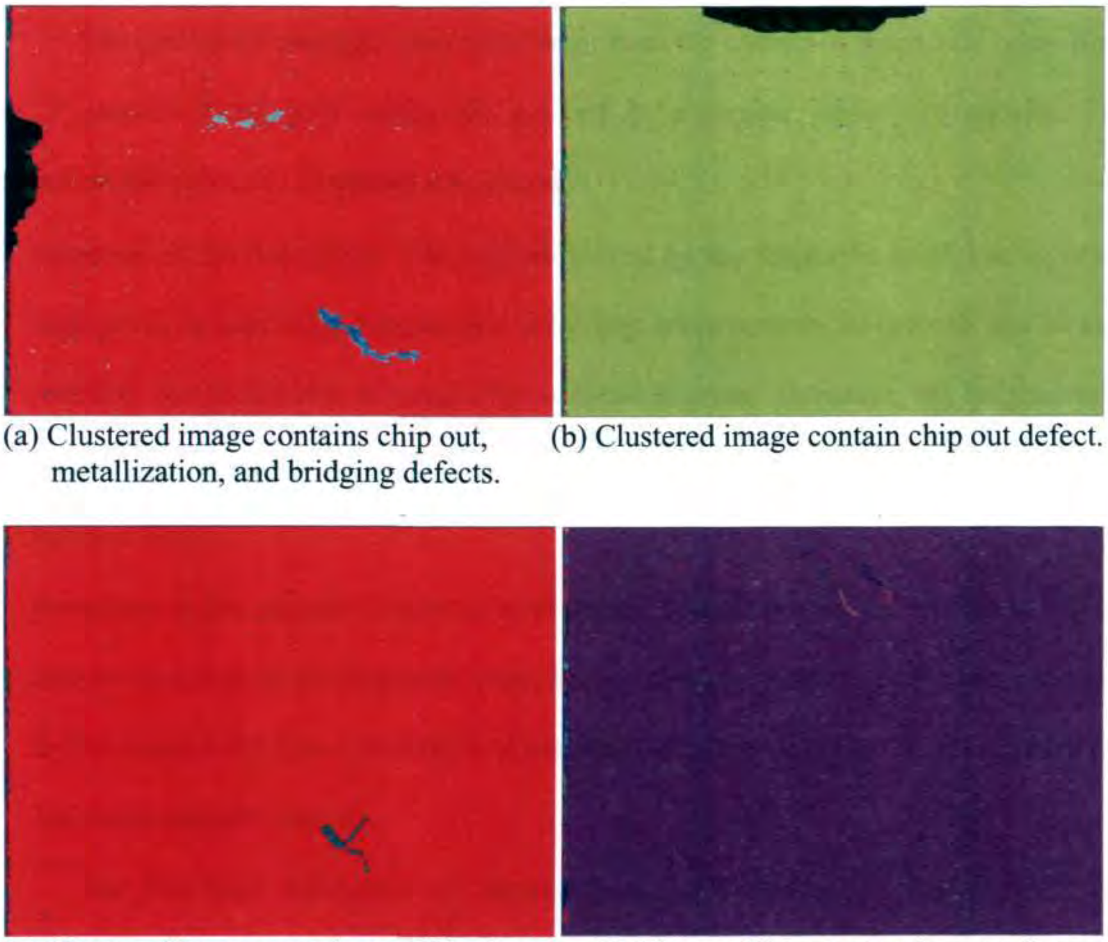
The above-described procedure estimates the reliability of each feature in each cluster. Here the reliability is defined as the statistical uniformity of the different features in each cluster. Since, the main objective of the classification is to define homogeneous clusters this has been considered valid reliability criteria. The update of the weighting coefficients is performed at each iteration of the clustering algorithm. However, as the clustering converge to a stable classification of the input data, the weighting coefficients converge to their final values.

Fig. 3.15 describes the computational simplicity of the algorithm. This is a strong point of the proposed approach compared to the other algorithms available in the literature. Also, as with all the algorithms that use point representatives, the proposed technique is suitable for recovering compact clusters. This improves the homogeneity in the feature domain of the obtained clusters.

Simple morphological image processing tends to join either too many unrelated defects together or not enough together. Expert semiconductor fabrication engineers have demonstrated that they can easily group clusters of defects from a common manufacturing problem source into a single signature. Capturing this thought the process of clustering is ideally suited for fuzzy logic. A system of rules is developed to join disconnected clusters based on the location of defect clusters.

The clusters are evaluated on a pair-wise basis using the rules and are joined or not joined based on a threshold. The system continuously re-evaluates the clusters under consideration as their rules change with each joining action. The techniques used to measure the features, and methods for improving the speed of the system are developed. Features have to be selected so as to encode all the useful information concerning the task of interest.

In the framework of computer vision and pattern recognition, many features are available. As a general comment, the choice and the number of the features to be used in a classification procedure vary according to the application. Fig. 3.16 shows the clustering results of the visual error image generated from Fig. 3.11 using the proposed clustering algorithm.



(a) Clustered image contains chip out, metallization, and bridging defects.

(b) Clustered image contain chip out defect.

(c) Clustered image contain metallization defect.

(d) Clustered image contain scratches.

Figure 3.16: Results of clustering (a) image contains chip out, metallization and bridging defects, (b) image contain chip out defect, (c) image contain metallization defect, and (d) image contain scratches.

3.6 Similarity Rule or Similarity Level

The role of the diagnostic level is to group the pixels of the input error image into M perceptually uniform clusters C_j. It is seen that this is achieved through a diagnostic rule that perform a low-level analysis of the input data. Each cluster C_j can be represented in the image coordinate system as a region R_j. The role of the similarity level is now to combine the regions R_j (corresponding to visual screening) into objects (corresponding to visual test results) according to what it is defined as similarity rules in analogy with the clinical language.

The context of similarity rules is different from the context of diagnostic rules. Now, the P pixels of the error image are grouped in a number $M << P$ of regions. This representation offers two important advantages:

- Accuracy of the description. The regions defined by the diagnostic level, are important supports to be used in the computation of features. More accurate models can now be used reducing the probability of generating artificial features. Moreover, the neighborhood relationship between regions provides a new tool for measuring the border similarity between them.
- Reduction in the amount of data to be evaluated. Similarity rule are applied on regions and not on pixels as the diagnostic rules. Even if a larger number of descriptors are used to characterize the visual properties of regions, the relationship ($M << P$) guarantees a low computational analysis.

The improved description of regions is not sufficient to directly provide their association into objects. This is because there is no unique definition of what an object is. This varies a lot according to the application. It is noticed that it is generally possible to find those similarity rules that correctly merge regions into valid objects in a specific application.

3.6.1 Region Merging Algorithm

In this level, the proposed algorithm is very similar to the merging phase of the Split & Merge technique discussed in Section 4.3.2. However, some advantages have been obtained compared to this classical technique. First, the regions defined by the similarity level, are not artificial blocks but perceptually homogeneous entities. Thus, their spatial descriptions are more reliable than the description of artificial blocks. Moreover, the number of regions defined by the similarity level, reflects the visual content of the image. In the classical Split & Merge algorithm, the image is often decomposed in many small squares in order to describe the visual details. The description of visual content in a significant number of regions improves the efficiency of the analysis and extends the academic interest of the proposed technique to practical applications.

The definition of the region descriptors (color and contrast) and the associated distance measures are the basis for the proposed similarity rules. The definition of more accurate models in the description of spatial properties of each region R_j provides a rich set of features. The models adopted in this research are the CIE model [118] for color and Michelson [119] model for contrast. According to CIE model the perceptual color difference between two regions R_i and R_j is defined as:

$$\Delta C(R_i, R_j) = \left[[(L(R_i) - L(R_j))/1.5]^2 + (A(R_i) - A(R_j))^2 + (B(R_i) - B(R_j))^2 \right]^{1/2} \tag{3.20}$$

where, $L(R_i)$ and $L(R_j)$ are the lightness values of the regions, $A(R_i)$ and $A(R_j)$ are the size of the regions, and $B(R_i)$ and $B(R_j)$ are the background area of all the remaining regions.

In the definition of a valid model for the description of contrast between regions, it is only consider two adjacent regions, R_i and R_j whose lightness values are respectively $L(R_i)$ and $L(R_j)$. According to the Michelson's contrast [119], the contrast between two regions is defined as:

$$C(R_{i,}R_j) = \frac{|L(R_i) - L(R_j)|}{L(R_i) + L(R_j)} \tag{3.21}$$

Similarity rules are composed of three successive steps: searching, evaluation and merging. The complete structure is summarized in Fig. 3.17. In the framework of similarity rules, two searching criteria have been defined as follows:

1. Small area: all regions R_j below a certain size are evaluated. The limit size is arbitrary and is fixed to 50 pixels. Here, small regions are compared with their neighborhood regions and disappear only if they can be associated with a sufficiently similar region.
2. Low contrast: all regions that present a color contrast $\overline{C}$ with a neighborhood region below the contrast sensitivity function are evaluated.

Once the searching criteria are defined, all the regions that satisfy the criteria undergo an evaluation step. During the evaluation step, the selected regions are compared with all their neighborhood regions. An evaluation step is performed for each search criterion.

Several evaluation criteria can be defined on these properties. It is possible to define a specific similarity measure taking into account only color information. The choice of the correct similarity depends on the available descriptors and on the application purposes. In the framework of similarity rules, it is proposed to apply the following measures:

1. The small regions evaluated first are compared using the contrast measure.
2. The low contrasted regions that are evaluated after the small regions are compared with the color distance.

For each searching list the regions are ordered according to their degree of similarity with their neighboring regions. An ordered list (target parade) of the most similar pair of adjacent regions is defined. Starting from the most similar ones, if a pair of regions satisfies the minimum similarity level (fixed by threshold values), a merging step takes place.

The merging step simply links two regions that have been defined sufficiently similar. This step is responsible for updating the relationships of all the neighborhood regions of the merged pair of regions. The new similarities relationships are used to update the ordered list and again the next most similar pair of regions undergoes the merging step.

As it appears from the procedure steps summarized in Fig. 3.17, it is necessary to fix a number of thresholds to stop the merging procedures. The first threshold corresponds to the minimum contrast required to distinguish the difference between two regions. The other threshold θ_C defines the minimum color similarity. These thresholds have been chosen according to the performed experimental results. The definition of similarity rules focused on a specific application and to achieve better and more stable results as it is demonstrated in Chapter 6.

```
1. Search all regions with small area ( ≤ 20 pixels).

        Evaluate contrast with their neighborhood regions
        Create target parade of most similar pair of regions.

        Merge best pair and update target parade.
        Repeat merging till the minimum contrast is obtained.

2. Search all regions with low contrast.

        Evaluate color similarity with their neighborhood regions.
        Create target parade of most similar pair of regions.

        Merge best pair and update target parade.
        Repeat merging till the minimum color similarity is obtained according to the
        threshold $\theta_C$
```

Figure 3.17: Definition of the similarity rules.

3.6.2 Regions refinement using Priority Rules

The defect segmentation performed here assigns a label ("chipping," "peel off," "scratch", "bridging," or "clear die") to each pixel in the error image. The classification system described here uses higher-level, domain-dependent knowledge to refine initial classification results. There are several motivations for such an approach. First, the initial classification depends primarily on information from very small image neighborhoods, and identifies defects on a pixel-by-pixel basis. It therefore ignores information such as defect shape, size, and position within the die. Second, the priority rules in Table 3.2 can employ fundamentally different types of rules for different defects, whereas the previous approaches used are restricted to a single topology for all classes that it can identify. Assigning a label to each pixel in the error image (the labels are "chip out", "peel off, scratch", "bridging", or "clear die") performs defect pattern classification.

The regions obtained with the similarity rules defined above, undergo a further refinement procedure. The region refinement utilized here depends heavily on mathematical morphology; morphological operations are performed to remove spurious regions and refine defect region shapes from the error images.

a) Refinement of Chip out Regions

In this application, it is reasonable to assume that chip out should lie on the outside of a chip. It is therefore possible to state this simple rule: Retain only those regions in the image that lie at the outer boundary of the chip. This is done by morphology dilation using a 3 x 3 structuring element, and points overlapping the chip out regions are removed from the dilated version.

The next step is to eliminate chip out regions that do not touch the boundary. Apply morphological opening to the initial chip out region, using the same structuring element that is used in the extraction of the chip boundary. This removes small "necks" that connect larger regions, smoothes region boundaries, and additionally removes some of the smaller regions from further consideration.

b) Refinement of Bridging Regions

A bridging is a longitudinal and radial connection of the two sensitive regions in the chip. Morphology operations [120] are employed to extract a *skeleton* of each region. A skeleton is a thinned representation of a region that retains information associated with its original size, orientation, and connectivity.

The skeletonization will typically produce a thinned image in which some small, spurious regions remain. Now, eliminate all regions having an area that is smaller than an empirically chosen threshold.

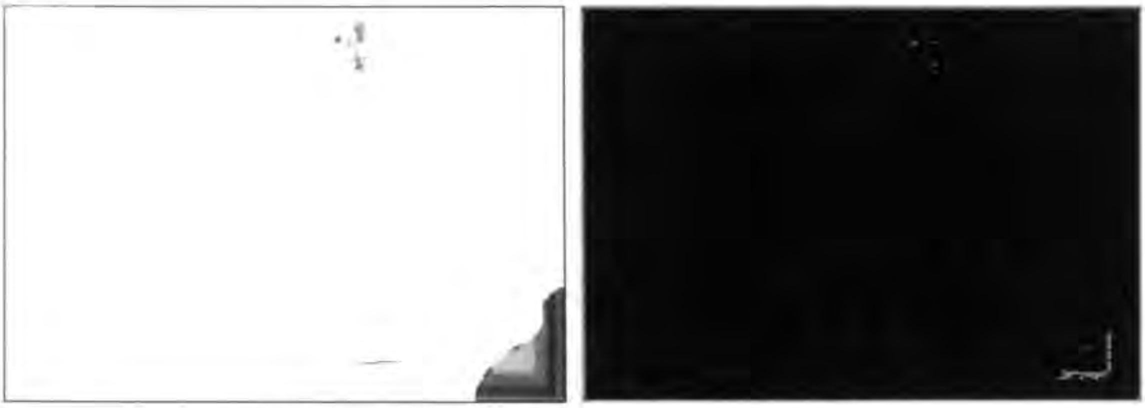
Table 3.2: Priority rules to resolve pairs of overlapping labels. The unshaded labels in the table represent the "winners" for cases that two labels (one shaded label from the left and one shaded label from the bottom) have been assigned to a single pixel.

Diffusion Layer	Diffusion Layer				
Scratch	Scratch	Scratch			
Bridging	Bridging	Bridging	Scratch		
Metal Lifting	Metal Lifting	Diffusion Layer	Metal Lifting	Bridging	
Chip Out	Chip Out	Chip Out	Chip Out	Chip Out	Chip Out
	Clear Die	Diffusion Layer	Scratch	Bridging	Metal Lifting

However, some defect regions still exist after refinement, causing some other defect regions from different layers to overlap. To resolve those overlaps, priority rules (Table 3.2) are developed to resolve conflicts when a pixel is assigned different labels by separate processing steps. For example, the region-refinement algorithm that is applied to clear die (die without any defect) will create overlaps with diffusion layer (a layer by which electrical isolation of one or more active circuits) regions. Because it is important to distinguish diffusion layer fault, it is given those regions preference over clear die regions. Whenever clear die overlaps another defect type, clear die surrenders its pixel label at that specific point. Because bridging and scratch can appear inside the chip out, they are given priority over diffusion layer whenever a conflict occurs.

3.6.3 Construction of Objects

After pixel classification the regions are obtained using similarity rule, then the objects are built by merging neighboring regions of the similar class. The objects belonging to the black/neutral/white classes can be built or not, depending on the needs. A two-step process is applied to build objects from the regions of the error image.



(a) (b)

Figure 3.18: (a) Visual error image, (b) Black/neutral/white pixel classification of error image.

First, the gray level images are decomposed into homogenous light/medium/dark areas by using the double thresholding technique: the values below the lower threshold are said to belong to the black class; the values above the upper threshold are said to belong to the white class. The remaining pixels are said to belong to the neutral class. The thresholds can be adaptive, i.e. they can be defined pixel-wise rather than remain constant over the whole image. In this case, the local threshold values must be provided as user-defined images. One possible way to prepare suitable threshold images is to take a Golden Master (GM) image and add tolerance margins to it. Fig. 3.18 shows the pixel classification of error image using thresholding. Even though the system handles double thresholding, simple thresholding can still be used. In this case, only the upper threshold and the neutral/white classes are considered.

Second, after pixel classification, the objects are built by grouping neighboring pixels of the same class. The objects belonging to the black/neutral/white classes can be built or not, depending on the needs. The objects may have holes. Pixels can touch each other along an edge or by a corner. Pixels touching by an edge are considered neighbors, if one speaks of 4-

connectedness. Pixels touching by a corner are also considered neighbors, if one speaks of 8-connectedness. Fig. 3.19 shows the objects built using similarity rule.

Figure 3.19: Objects built using similarity rule.

For the sake of computational efficiency, the objects are described as lists of runs. A run is a horizontal segment of pixels of the same class. It allows working at the object level but also down to the runs level. This allows faster processing in critical cases to compute the object features.

3.7 Logical Rules or Logical Level

After the thresholding and segmentation parameters are set, processing the error image generates a list of objects. Each object receives a distinct number starting from 0. The list of objects can be traversed sequentially forwards or backwards. An object can also be reached directly by its number. Sequential access is more efficient. When objects have been built, one can compute a series of energy features, chosen among the objects. Then the value of the computed feature(s) of any object can be retrieved at will. These objects are designated with number. Once an error image has been segmented and the corresponding blobs have been built. The texture features of the objects will be computed based on the co-occurrence matrix.

3.7.1 Feature Extraction of Objects

'The problem of classification is basically one of partitioning the feature space into regions, one region for each category.'[121]. So, high discriminating features will lead to high and accurate classification rates. Color values (RGB-channels), as local features, are directly related with the images, so they were introduced to the system without any change. So, two groups of color features were introduced to the system; one-to-one pixel mapping of color feature set in the first, whereas n-to-one in the other, with n (or neighborhood) determined by the rgb-window size.

Structural analysis will yield important information for classification, so co-occurrence matrix [122] is used to extract textural features in the regions. Co-occurrence matrix is a single level dependence matrix that contains the relative frequencies of two coordinate elements separated by a distance d. As you move from one pixel to another on the image, entries of the initial and final pixels become the coordinates of the co-occurrence matrix to be incremented, which in the end will represent structural characteristics of the image. Therefore, moving in different directions and distances on the image will lead to different co-occurrence matrices. In literature, most commonly used pixel separation distance and directions (angles) are 1 pixel and 0, π /4, π /2, and 3 /4 radians, respectively, which are also used in this research.

The total textural energy features $f_t(d)$ derived from the co-occurrence matrices for the error image are:

$$f_t(d) = \sum_{i=0}^{255}\sum_{j=0}^{255} S(i,j,d)^2 \tag{3.22}$$

In the above equation, $S(i,j,d)$ refers to the normalized entry of the co-occurrence matrices found by dividing the initial entries with total number of pixels of the sub-image, where (i, j) are the coordinates of the co-occurrence matrices and d is the pixel separation distance.

Building the co-occurrence matrix S is done by computing for each pair of full grayscale pixel range (usually 0-255), the number of adjacent pixels that hold this pair in the image, for different directions and distances.

For example, for image of size $(M \times N)$ pixels, for a distance $d = 1$, and in the horizontal direction, the S matrix will be defined as follows:

$$S(i, j, d, 0^{\circ}) = \#\{((u, v), (m, n)) \in (M \times N) \times (M \times N) | u - m| = 0, |v - n| = 1, I(u, v) = i, I(m, n) = j\} \tag{3.23}$$

Where $i, j = 0....255$ (number of possible gray-levels), $u, m = 1...M$ (width of the image), and $v, n = 1...N$ (height of the image). When S is calculated for different distances and directions, it requires expensive computational resources, but it contains a huge amount of texture information.

In order to locate the spectral differences within and between images, many of the spectral analysis methods like Fourier; wavelet or cosine transforms could be used. The advantage of localization in time and frequency made wavelets preferable. Within the orthogonal and compactly supported wavelets (daubechies, symlets, and coiflets), coiflets have more number of vanishing moments at the same order, so have more information on the details. Therefore, 2^{nd} order coiflets wavelet decomposition is applied on each sub-image retrieving 1 approximate and 2×3 detailed (horizontal, vertical and diagonal for each order) coefficients.

Calculating the texture and wavelet features of the whole image will yield important global results maybe, but obviously will not provide us enough information about both the size and type of the defects that are crucial in classification or discrimination between chip out, bridging, and metallization areas. Because of that, these features were calculated on windowed sub-sections of each image.

Two different window approaches were used to get the sub-images. In discrete window approach, images were divided into 64 16×16 non-overlapping sub-images. On each sub-image, both textural and wavelet features were calculated and they were related to each pixel within that sub-image. However in sliding approach, features were calculated a pixel at a time on the 16×16 neighborhood by zero-padding the areas outside the image. That's why sliding window method required 256 times more computation than discrete window for 128×128 image size and 16×16 window size.

The resulting four texture features of a pixel were from the average of co-occurrence matrices in all directions. Wavelet features were found by taking the average and standard deviations of the coefficients of each decomposition class. At the end of feature extraction, there were 8 textural, 28 wavelet, and 3 color features (27 for 3×3, or 75 for 5×5 rgb-windows) making a total of 39 (63, or 111) features. To show how domain-dependent rules can be developed and applied. It is necessary to state the simple rules used in the detection process.

Let us define the total defect components (only consider four defect types) in the error image as

Defect Components = Chip off + Scratch + Metallization peel off + Bridging

Dimensions of the images were differing within the data set. In order to decrease computation time while doing mathematical operations, images had to be square. In the experiments only square images are considered. Fig. 3.20 (a) shows image under inspection and Fig. 3.20 (b) shows the error image recovered and the objects built using similarity rules. The textural energy features derived from the co-occurrence matrix for each defect components are explained as follows.

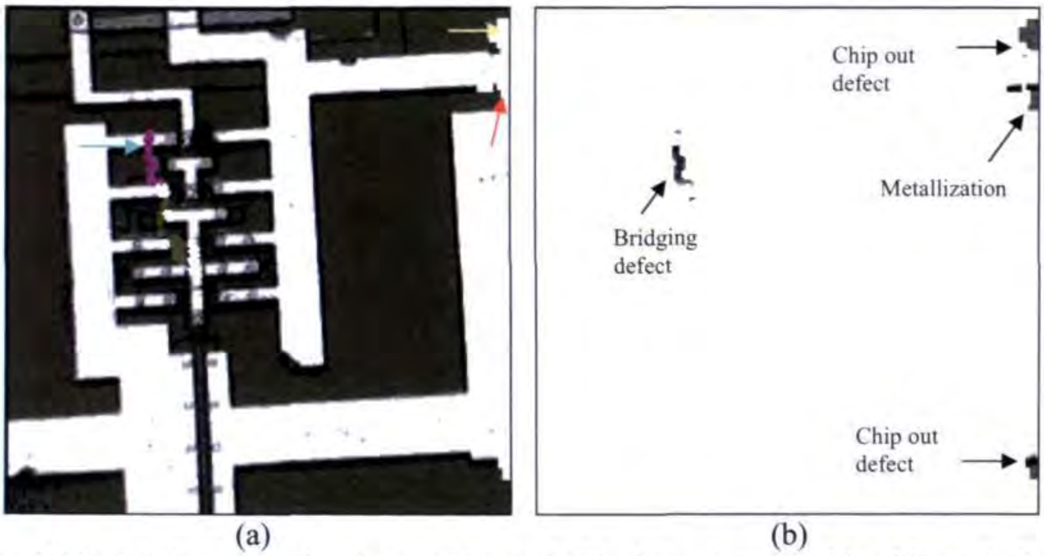


Figure 3.20: (a) Image under Inspection, and (b) Error image with objects built using similarity rule.

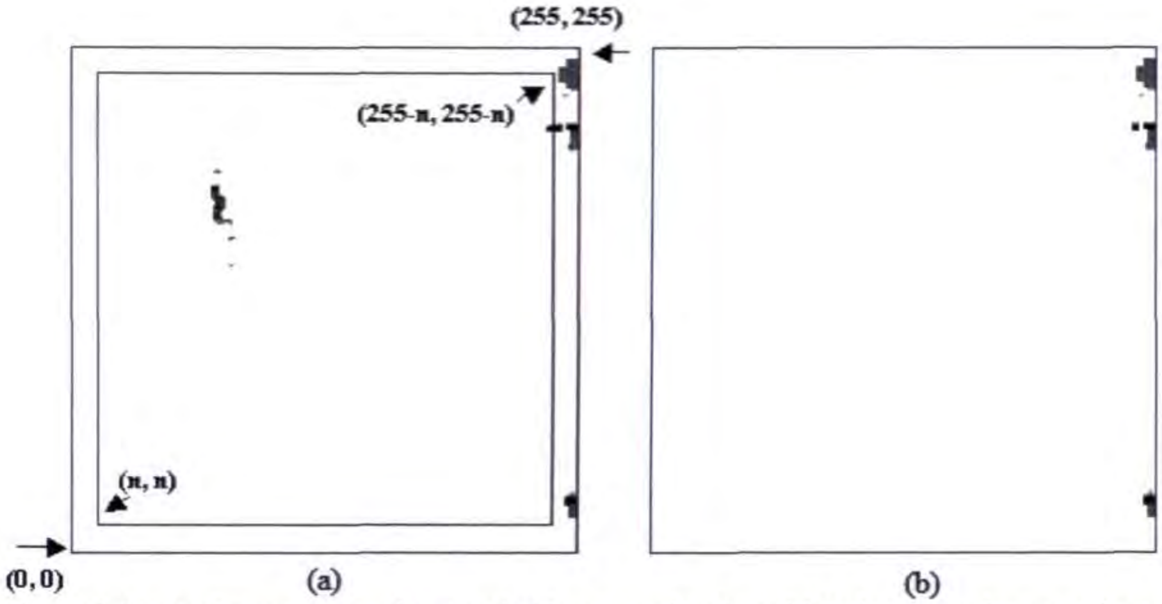
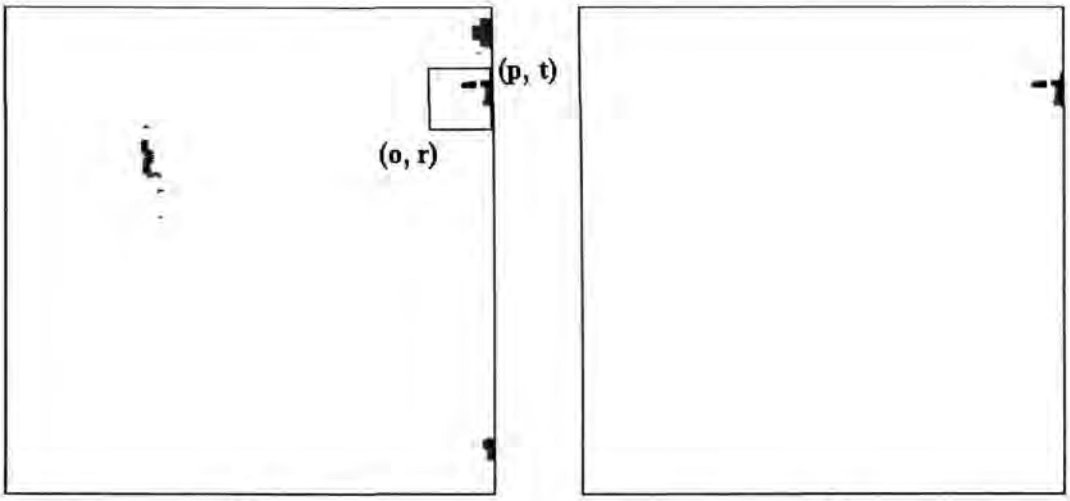


Figure 3.21: (a) Image shows the actual chip out region for which the textural energy features to be calculated, and (b) texture feature of the chip out region.

Fig. 3.21 (a) shows the actual chip out region for which the textural energy feature to be calculated and Fig. 3.21 (b) shows texture feature of the chip out region for the given test image. Textural energy feature derived from the co-occurrence matrix for the chip out region is given as

$$f_c(d) = \left(f_t(d) - \sum_{i=n}^{255-n} \sum_{j=n}^{255-n} S(i,j,d)^2 \right) \tag{3.24}$$

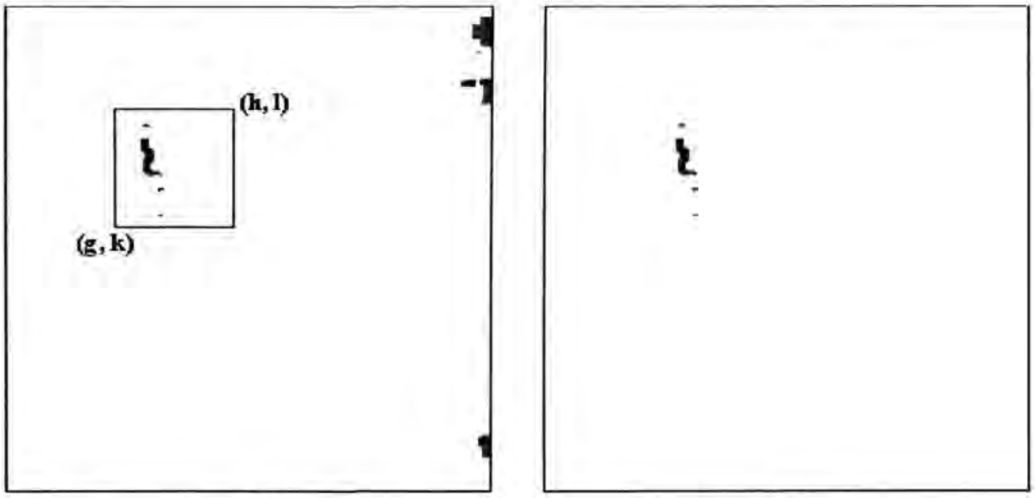
where n is the non-chip out region.



(a) (b)

Figure 3.22: (a) Image shows the metallization region for which the textural energy features to be calculated, and (b) texture feature of the metallization region.

Fig. 3.22 (a) shows the metallization region for which the textural energy feature to be calculated and Fig. 3.22 (b) shows the texture feature of the metallization region.



(a) (b)

Figure 3.23: (a) Image shows the bridging region for which the textural energy features to be calculated, and (b) texture feature of the bridging region.

Hence, for the given error image the metallization peel off textural energy feature is given as

$$f_m(d) = \sum_{i=o}^{p} \sum_{j=r}^{t} S(i,j,d)^2 \tag{3.25}$$

where (o,r) and (p,t) are the metallization peel off region and the values used in the experiment for (o,r) and (p,t) are (221,202) and (255,236) respectively. Fig. 3.23 (a) shows the bridging region for which the textural energy feature to be calculated, and Fig. 3.24 (b) shows the texture feature of the bridging region.

The bridging defect energy term is given as

$$f_b(d) = \sum_{i=g}^{h} \sum_{j=k}^{l} S(i,j,d)^2 \tag{3.26}$$

where, (g,k) and (h,l) are the bridging region and the values used in the experiment for (g,k) and (h,l) are (58,132) and (118,192) respectively.

For the scratch region equation (3.24) becomes

$$f_s(d) = \left(f_t(d) - (f_m(d) + f_b(d))\right) \tag{3.27}$$

3.7.2 Rule-based object finding

The similarity level describes the content of a single frame in terms of objects of interest. The next step is to predict what kind of defect pattern exists after eliminating unwanted objects. This is achieved using rule-based object finding. It starts with a rule-base, which contains all of the appropriate knowledge encoded into If-Then rules, and a working memory, which may or may not initially contain any data, assertions or initially known information. The system examines all the rule conditions (IF) and determines a subset, the conflict set, of the rules whose conditions are satisfied based on the working memory. Of this conflict set, one of those rules is triggered (fired). Which rule is chosen is based on a conflict resolution strategy. When the rule is fired, any actions specified in its THEN clause are carried out. This loop of firing rules and performing actions continues until one of two conditions is met: there are no more rules whose conditions are satisfied or a rule is fired whose action specifies the program should terminate. Which strategy is chosen can be determined by the problem or it may be a matter of preference. In any case, it is vital as it controls which of the applicable rules are fired and thus how the entire system behaves.

3.7.3 Rules for Semiconductor Chip Defects

a) Rule 1

The Golden Master (GM) die image in Fig. 3.8 (a) is matched (section 3.2.3) to the inspected die image in Fig. 3.8 (b), and the resultant image is the error image as shown in Fig. 3.8 (c). Figure 3.8 (c) contains defects such as chip-out, bridging and scratch. Rule 1 state that if the defect size in the error image exceeds 1 μm, then there is a substantial defect found in the particular inspected die image.

b) Rule 2

The texture energy features of the chip out regions are extracted based on the co-occurrence matrix as expressed in Equation 3.24. The total area of the objects in the chip out region are expressed as $N_{c1} + N_{c2} + ... + N_{cn}$, then according to rule 2 if the area of objects in the chip out region i.e., $\sum_{s=1}^{n} N_{cs}$ is greater than the area of objects in scratch, bridging, and metallization regions i.e., $N_c > N_s > N_b > N_m$ where N_c, N_s, N_b, and N_m is the area of objects in chip out, scratch, bridging, and metallization regions, then the defect is caused by chip out. The area of object is calculated from the feature extracted from each defect regions using co-occurrence matrix as in section 3.7.1 based on the following geometry.

Area is defined as the number of pixel count (P_i):

$$\text{Area } N = \sum P_i \tag{3.28}$$

Hence, it is clear that any texture energy feature originated from the chip out region of the error image results in a defect of type chip out.

c) Rule 3

The texture energy features of the bridge occurring regions in the error image are extracted based on the co-occurrence matrix as expressed in Equation 3.26. The total area of the objects in the bridging regions are expressed as $N_{b1} + N_{b2} + ... + N_{bn}$, then according to rule 3 if the area of objects in all bridging regions i.e., $\sum_{s=1}^{n} N_{bs}$ is greater than the area of objects in scratch, and metallization regions i.e., $N_b > N_s > N_m$ then the defect is caused by bridging. Hence, it is clear that any texture energy feature originated from the bridge occurring regions of the error image results in a defect of type bridging.

d) Rule 4

The texture energy features of the possible metal lifting regions are extracted based on the co-occurrence matrix as expressed in Equation 3.25. The total area of objects in the metallization regions are expressed as $N_{m1} + N_{m2} + ... + N_{mn}$, then according to rule 4, if the area of objects in all the metallization regions i.e., $\sum_{s=1}^{n} N_{ms}$ is greater than the area of objects in bridging, and scratch regions i.e., $N_m > N_b > N_s$ then the defect is caused by metallization of the inspected die. Hence, it is clear that any texture energy feature originated from the possible metallization regions of the error image results in a defect of type metallization.

e) Rule 5

The texture energy features of the scratch regions are extracted based on the co-occurrence matrix as expressed in Equation 3.27 i.e., $f_s(d) = \left(f_t(d) - (f_m(d) + f_b(d))\right)$. According to rule 5, if the area of objects in the scratch region N_s is greater than the area of objects in bridging, and metallization regions i.e., $N_s > N_h > N_m$ then the defect is caused by scratch. Rules 3, 4 and 5 is valid only if $N_c = 0$.

Table 3.3: Summarizes the results of object finding procedure.

Coordinate or Position	Rules	Results
Chip Out	Rule 2	Chip Out Fault
	Rule 4	Mettallization or Glassivation Fault
Bridging	Rule 3	Bridging Fault
	Rule 5	Scratch or Glassivation Fault
	Rule 4	Metallization or Glassivation Fault
Miscellaneous	Rule 5	Scratch or Glassivation Fault

The results of objects finding procedure are summarized in Table 3.3. Hence using a rule-based system, the structural form is consistent, the variables, comparators, rules etc. are all given names, which are meaningful to both the user and the developer.

3.8 Summary

In this Chapter a new rule-based defect image segmentation and object finding scheme were presented. Many techniques exist in literature that solves defect image segmentation for a specific application. However, these techniques cannot be straight forwardly extended to different applications, because they are based on particular hypothesis that are not valid in general.

The technique presented in this Chapter approaches differently the defect image segmentation problem. The major concern is to define a general framework able to automatically analyze the visual information in terms of objects of interest. For this, a clinical language like structure has been defined that is composed of three successive levels: the diagnostic, the similarity and the logical level.

The diagnostic level defines general rules that simplify the visual data from a set of P pixels to a much smaller set of M regions. These rules are based on the properties of the human visual system, and takes into account for the major mechanisms of human perception. This choice has two results. First, the extracted regions correspond to the perceptually uniform areas in the image. Second, the same analysis can be applied to a broad class of natural images independently from the application at hand. Note that the diagnostic level finds analogy with the clinical language where symptoms are combined to form proper screening on the error image according to specific diagnostic rules.

The similarity level defines rule that takes advantage of the simplification performed by the previous analysis by introducing application specific priority rules. Note that the similarity level finds analogy with the clinical language where the screenings are combined in test results according to specific similarity rules.

The logic level finds analogy with the clinical language where test results keep a logic relationship in a defect object. The proposed approach compared to existing solutions is original under several aspects. First of all its modularity offers a high degree of adaptivity with a minimum user interaction. The segmentation problem is decomposed into a fully automatic simplification and a successive specialized interpretation.

In summary, the diagnostic rule generates screening of disease from symptoms, similarity rule generate test report from screening and logical rules generate the final decision of the disease from the test report. Thus, a large collection of complex defect patterns can be described by a small number of elementary sub-patterns and clinical rules. They are: -

- **Diagnostic rule**

It controls the association of pixels in perceptually uniform regions.

- **Similarity rule**

The role of the similarity rule is to combine the regions into objects.

- **Logical rules**, which is further divided in to five sub- rules. They are: -

Rule 1: if the defect size in the error image is greater than 1 μm, then search defect pattern.

Rule 2: if the areas of objects in chip out region $\sum_{s=1}^{n} N_{cs} > N_s > N_b > N_m$ then it is chip out defect.

Rule 3: if the area of objects in bridging region $\sum_{s=1}^{n} N_{bs} > N_s > N_m$, then it is bridging defect.

Rule 4: if the area of objects in metallization region $\sum_{s=1}^{n} N_{ms} > N_b > N_s$, then it is metallization defect.

Rule 5: if (no other rule fires), then it is scratch defect.

Note that in addition to the access to a more accurate analysis of the visual information, the data simplification process makes the interpretation task computationally efficient. The possibility for integrating application dependent rules directly on the simplified data is another important element of originality of the technique. Examples on few applications are proposed in Chapter 4, 5, and 6. The validity of the proposed defect image segmentation method requires a validation procedure. This is discussed in the next chapter in the context of semiconductor, printing and biological applications.

Table 3.4: Summary of the logical rules to detect semiconductor defect patterns.

If Conditions	Rule 1	Rule 2	Rule 3	Rule 4	Rule 5
If the defect size in the error image greater than 1 μm	Then search for defect				
If the area of objects $\sum_{s=1}^{n} N_{cs} > N_s > N_b > N_m$		Then Chip out			
If the area of objects $\sum_{s=1}^{n} N_{bs} > N_s > N_m$			Then Bridging		
If the area of feature $\sum_{s=1}^{n} N_{ms} > N_b > N_s$				Then Metallization	
If No rule fires					Then Scratch

Besides the originality on the global structure, new procedures are proposed, to enhance the performances of each level of the technique. In particular the clustering procedure used in the diagnostic level has been enhanced by a new initialization procedure that automatically selects the number of cluster representatives. In the same level, the definition of a dynamically weighted multi-feature distance measure has improved the classification performances. The rules to find the defect objects are summarized in Table 3.4. These results need to be validated quantitatively however, in the subsequent Chapters.

Chapter 4

Defect Pattern Detection in Semiconductor Wafer Images

4.1 Introduction

In the context of this thesis it is interesting in exploiting color and texture perception mechanisms in order to describe the defect image in terms of perceptually uniform regions. Chapter 2 reviews the various defect pattern detection systems in the semiconductor industry and Chapter 3 explains the theoretical background of a new rule-based system. This Chapter focuses on the various constrained clustering segmentation techniques together with the application to segment semiconductor defects. This Chapter also focuses on the texture values of the defect image. Luminance carries important information about the visual image. In particular, regular variations of luminance in a certain area of the visual field can be described in terms of texture features. Regular texture feature are perceived as homogeneous properties of the scene by the human visual system. This Chapter will also define features from textures models. Texture features will be used to classify into perceptually homogeneous classes the pixels of the image.

This Chapter is composed of two main sections. Section 4.3 discussed two different models of textures: stochastic and structured textures. Different methods have been evaluated that provide a description of textures with different degree of accuracy. In particular it is found that stochastic textures can be efficiently described with simple statistical approaches, while structured textures are better characterized by frequency-based approaches. Section 4.5 will discuss the application of a new rule-based approach in semiconductor images.

4.2 Need for Rule-Based Systems and Algorithms

Humans are certainly the best pattern recognizers. Although it is not fully understood how humans recognize patterns, it is now possible to explain and describe several mechanisms of human vision. In particular, psychovisual studies have shown that the process of seeing may be comparable to a system where the input information is successively treated and interpreted [123]. Typical example for such an approach is that how an operator make decision to segregate defect pattern's such as, foreign material, saw into metal, crack, diffusion fault, missing probe mark, scratches, chip die, and glassivation void, etc., in a semiconductor die surface by manual means. The identification and segmentation of these defects by automated means is tried out for the first time in the semiconductor industry using rule-based approach. The first levels of this system are responsible for detecting the visual signal and perform a first data reduction, where the redundant information is simplified, while the information of interest is highlighted. Following this simplification process, perceptual higher-level rules are defined that interprets the visual content. Some rules are hardwired in the human physiology system; others are applied according to the human experience or subjectivity reasons.

The increased number of defect types drives up the cost of fault reduction. Since yield engineers have observed that different types of defect inspection equipment capture some defect types more effectively than they do others, companies are forced to buy many types of defect inspection tools in order to identify most potential sources of electrical faults. As process complexity increases, defect detection may pose a daunting economic challenge to all but the largest semiconductor manufacturers and up to date no body have found a final solution to in-line inspection after wafer dicing. In general, it is difficult to detect defects because of the inherent variability of wafers in the assembly line. Although much of the work reported above has yielded results that are good in a quantitative sense, evaluation suggests that the defect analysis results could be improved in a qualitative sense by refining the resulting shapes and extent of detected defect regions in the images. According to the rule-based approach it decomposes the general defect detection problem in several levels. The lowest levels have a high degree of generality. Their main role is to simplify the input data and to extract the perceptually meaningful information. The proposed process is sufficiently general to be applied on a wide class of applications and input data. It is based on the principle of statistical pattern recognition [6] and applies a clustering process on perceptual significant features. The highest levels combine the information generated by the lowest levels according to the specific criteria of the application at hand.

The focus of this research has been on developing an automated method for detecting defects on the wafer die surface using rule-based approach. Also, this research focuses to edit the original wafermap and thereby generates final (before die-attach) wafermap of each and every wafer under inspection. Fig. 4.1 shows a scenario for collecting and analyzing defect data in a production environment. As wafers exit a fabrication process (e.g., fabrication process A) wafermap data was generated by an in-line defect detection workstation generally incorporating a microscopy or light scattering system. Once the instrument has scanned a wafer, its electronic wafermap is moved to a yield management system for prediction or to alarm operators when process is going out of specification or when maintenance or calibration must be scheduled.

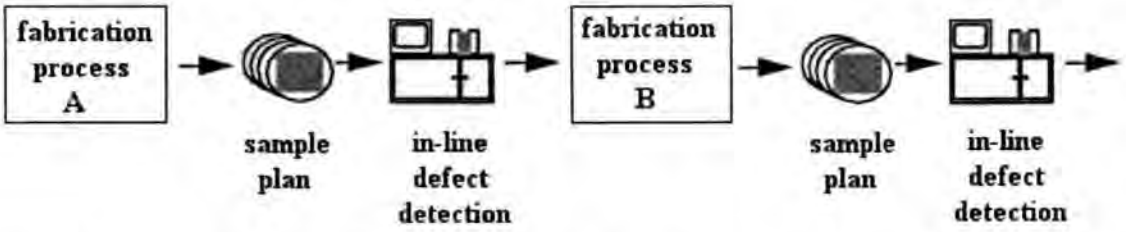


Figure 4.1: Schematic representation of a scenario for collecting and analyzing defect data in a production environment.

The information in the wafermap consists of detected defect coordinates as well as process information such as step, layer, and product. An operator viewing a plot of the coordinate points during analysis typically observes the existence of patterns in the wafermap data manually. Automation tools for wafermap analysis currently use simple nearest-neighbor clustering [124] techniques to group data primarily into "random" or "grouped" events but specific classifications are not obtained.

For example, small spurious defect regions may have only a small effect on pixel-wise statistical classification accuracy, but are objectionable to the human observer. In many cases, such regions can be identified and eliminated easily.

4.3 Texture models

4.3.1 What is visual texture?

Visual texture is a perceptual property of a region in an image. Texture is not the property of a single pixel, but of a set of adjacent pixels. Texture is independent of the absolute color and brightness but it depends on their relative variations of color and brightness in the region. Fig. 4.2 display some examples of textures extracted from the album of photographs published by Brodatz [125]. Even though texture is an intuitive concept, its formal definition has proven to be elusive [126, 127]. Still, many authors have tried to provide a qualitative definition of texture. According to Hawkins in [128], the notion of texture depends on three ingredients:

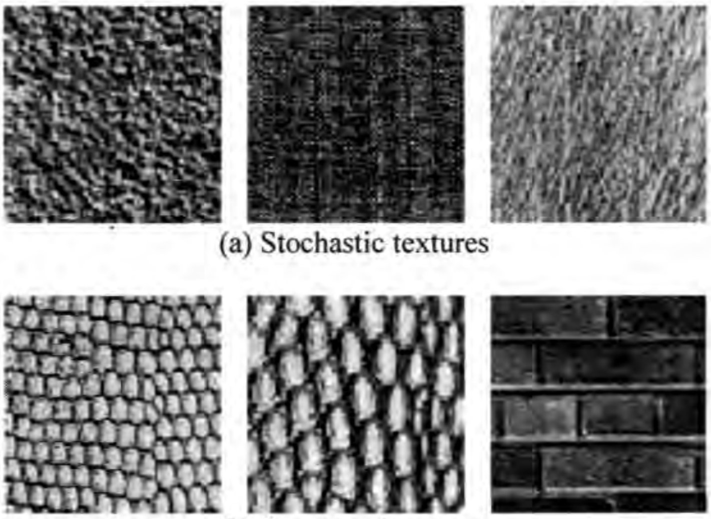
- some local order is repeated over a region, which is large in comparison to the order's size,
- the order consists in the non-random arrangements of elementary parts,
- the parts are roughly uniform entities having approximately the same dimensions every-where within the textured region.

The difficulty in the definition of texture depends on the fact that it is related to the perceptual mechanisms of human vision [129]. Textures do not have uniform intensity, but are none-the-less perceived as homogeneous regions by a human observer. However, different interpretations at different distances and at different degrees of visual attention are possible for the same texture stimulus [130, 131].

Despite this lack of a universally agreed definition, all researchers agree on two points. Firstly, there is significant variation in intensity levels between nearby pixels; that is, at the limit of resolution, there is non-homogeneity. Secondly, texture is a homogeneous property at some spatial scale larger than the resolution of the image. These properties are used as guidelines in the effort to describe texture information effectively.

Measuring visual texture has many important applications. At a large scale, image texture has been used in the analysis of satellite images. At a much smaller scale, the texture of cell nuclei is among the best indicators of precancerous cells in medical screening programs.

Recently, techniques to automatically generate texture have important applications in the movie industry, in video coding and in virtual reality.



(a) Stochastic textures

(b) Structured textures

Figure 4.2: Examples of textures from the Brodatz collection. (a) represents structured textures, while (b) represents stochastic textures [125].

In the context of this thesis, measuring texture is important to describe the visual content of an image. In particular it is interesting in extracting those image features that are able to describe an image area so as to assess if its texture is perceived similar or different from adjacent areas by a human observer.

In literature, many texture descriptors have been proposed to measure the similarity or dissimilarity between image areas or to automatically classify an input image into existing texture classes. In [132, 133], several descriptors are proposed, compared and validated on large image databases. Although some techniques exhibit better performances than others for specific image samples, the extensive comparisons have not been able yet to select a unique valid solution or a clear winner approach.

In this thesis it has investigated two approaches that have revealed good performances in describing texture properties [134]: a statistical approach and a frequency-based approach.

4.3.2 Statistical approaches

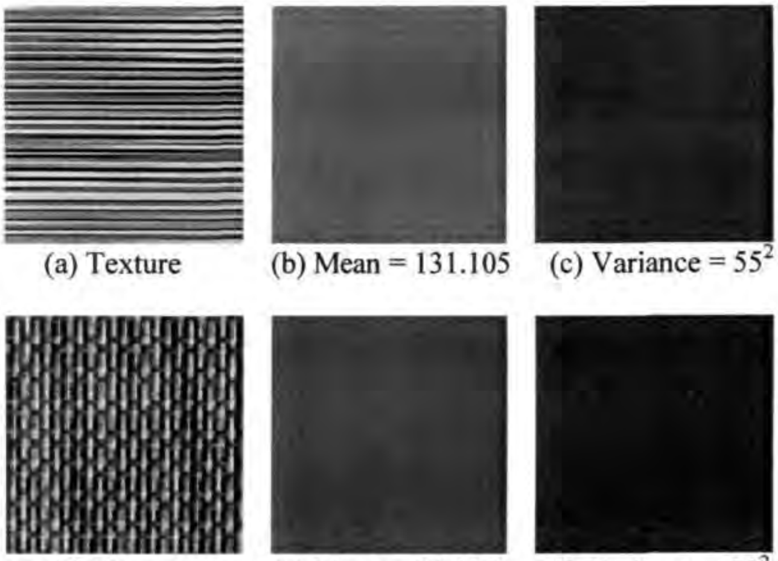
Statistical approaches are inspired by the fact that many natural textures may be well approximated by discrete stochastic fields. These are arrays of numbers that are randomly distributed in amplitude and that are ruled by some joint probability density. Several authors attempted to determine what are the parameters of the stochastic texture fields that most influence their perception. Pratt et al. [115] found that texture field pairs differing in their first-order and second-order distributions can be discriminated. While differences in third order and higher-order distributions texture fields cannot be perceived if the first and second-order are identical. These results suggested the use of two simple moment-based descriptors: the mean μ and the variance σ^2. They are defined as:

$$\mu = E\{x_0\} \tag{4.1}$$

$$\sigma^2 = E\{(x_0 - \mu)^2\} \tag{4.2}$$

In Fig. 4.3 it represents two texture samples extracted by the Brodatz collection together with the results of their statistical analysis. In this example, the mean and the variance are certainly sufficient descriptors to discriminate the two textures.

Statistical approaches to texture description work well for those natural textures that can be approximated by discrete stochastic fields. They are not always significant when the textures are composed by structured elements as in Fig. 4.2(b). In this case, co-occurrence methods (Chapter 3) such as those suggested by Haralick [135] should be preferred. These solutions, however, are responsible for generating a high number of features. Each feature is significant for highlighting a certain structure property. The increase in the number of features make these methods suitable for texture classification [136], but it reduces their validity in the framework of unsupervised clustering due to the curse of dimensionality problem [6, 90]. A feature selection procedure should be applied to select only the significant texture features to avoid masking the important descriptors.


(a) Texture (b) Mean = 131.105 (c) Variance = 55^2

(d) Texture (e) Mean = 94.4151 (f) Variance = 50^2

Figure 4.3: Examples of statistical descriptors of texture. Two textures (a) and (d) have been described with their local mean μ and variance σ^2. These are represented in (b) and (c) for texture (a) and (e) and (f) for texture (d) [125].

If a more precise description of the texture properties is available a priori (e.g. looking for a texture with a specific orientation or frequency component), an alternative technique to co-occurrence method is represented by the frequency-based approaches to texture analysis. These are described in the following subsections.

4.3.3 Frequency-based approaches

Frequency-based approaches to texture description may be summarized by the scheme of Fig. 4.4. In this scheme, two main stages are reported: a filtering stage and a feature selection stage. First the input image undergoes a filtering approach that transforms its content from the image plane to the frequency domain. Filtering is generally applied on the brightness or the lightness component of the image, where the texture information is encoded. Several filtering techniques are applied in the literature, they are the Gabor filter [137-140], the Wavelet transform [141-144], the Discrete Cosine transform [145] and the Quadrature Mirror filter [146].

The performances in the context of texture analysis of some of these filtering approaches have been recently compared in [133, 147]. It appears that none of the above filtering methods provide an optimum analysis for all type of textures.

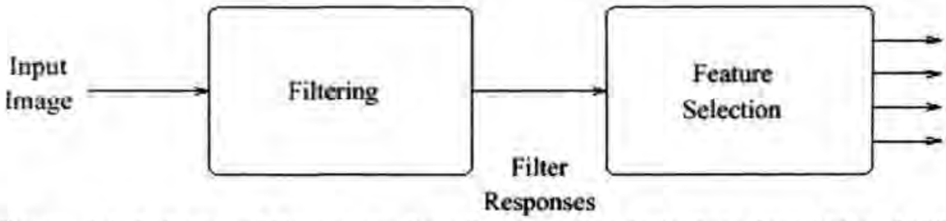


Figure 4.4: Scheme of Frequency-based approaches to texture description [137].

However, among the several filtering strategies, Wavelet transform and Discrete Cosine transform filtering is particularly appealing in the context of this thesis. In addition, 2-D wavelet transform [148, 149] offers the ability of robust feature extraction in images as a popular tool in image processing. Concerning the classical segmentation problem, that is dividing an image into homogeneous regions, the discovery of a generally effective scheme remains a challenge. Many interesting techniques have been suggested so far including spatial frequency techniques [150] and relevant ones like texture clustering in the wavelet domain. Most of these methodologies use very simple features like the energy of the wavelet channels [150] or the variance of the wavelet coefficients [151].

The advantage of localization in time and frequency made frequency based approaches preferable in my research work. Also, it provides the maximum joint resolution in space and spatial/frequency. This is highly significant in the process of texture extraction in which conflicting objectives of accuracy in texture representation and texture spatial localization are important. Secondly, it doesn't suffer of the curse of dimensionality problem because not all the features are significant in describing a specific texture.

4.3.3.1 Feature Extraction in the K-Level Wavelet Domain

The approach stems from this line of research related to the wavelet domain judicious processing. However, there is need for much more sophisticated wavelet feature extraction methods if one wants to solve the segmentation problem in its defect recognition, taking into account the high accuracy required. Following this reasoning it is proposed to incorporate in the research efforts multidimensional wavelet features, unlike the previously presented scalar feature extraction methodologies in the wavelet domain [150, 151]. These multidimensional features, coming from the application of the K-Level 2-D DWT, namely, efficient multidimensional feature selection in the wavelet domain based classification.

The problem of defect discrimination, aiming at segmenting the defective areas in images, is considered in the wavelet domain, since it has been demonstrated that discrete wavelet transform (DWT) can in general lead to better image modeling, as for instance to better encoding (wavelet image compression [152, 153] is one of the best compression methodologies) and to better texture modeling [152]. Also, in this way, it can better exploit the known local information extraction properties of wavelet signal decomposition as well as the known features of wavelet de-noising procedures [154]. The popular 2-D discrete wavelet transform scheme is used to obtain the wavelet analysis of the original image data containing defects. It is expected that the images considered in the wavelet domain should be smooth but due to the known time frequency localization properties of the wavelet transform, the defective areas whose statistics vary from the ones of the image background should more or less clearly emerge from the background. It is experimented with the standard 2-D Wavelet transform using nearly all the well-known wavelet bases like Haar, Daubechies, Coiflet, Symmlet etc. as well as with Meyer's and Kolaczyk's 2-D Wavelet transforms [149]. However, Daubechies and Haar wavelets have exhibited similar and the most accurate results.

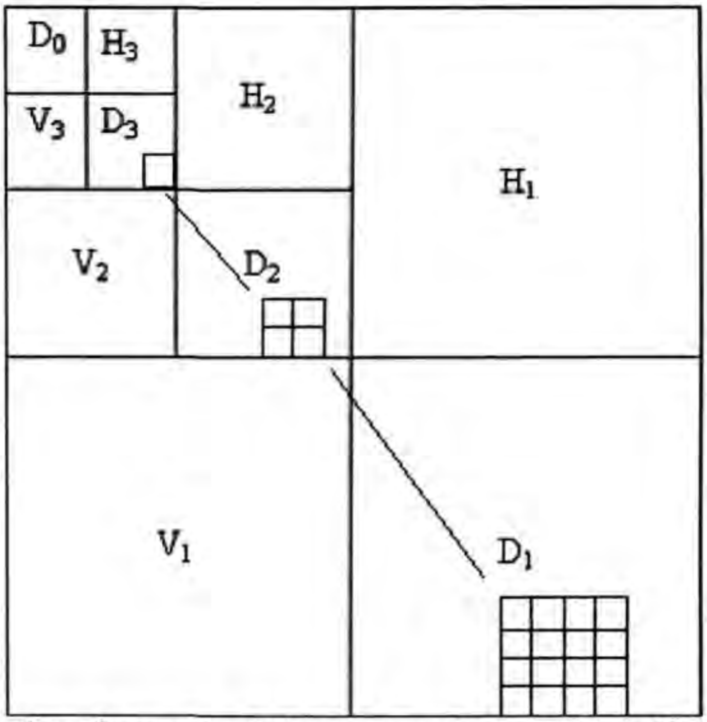


Figure 4.5: Illustration of a sample of the corresponding wavelet coefficients sub-windows taking place in the formation of K-Level Wavelet domain. Three such windows are shown one for each channel.

The proposed methodology involving multidimensional wavelet features obtained from the K-Level 2-D DWT, with application to defect segmentation, can be outlined as follows:

The $N \times N$ image is raster scanned by $W \times W$ sliding windows. Each such window is transformed into the wavelet domain using the K-Level 2-D DWT. As a result, the wavelet coefficients organized in $3 \times K + 1$ channels are obtained (see Fig. 4.5).

Starting from the channel D$_0$ (the upper left window in Fig. 4.5), the multidimensional features are formed from the wavelet coefficients, having as components $3 \times K + 1$ windows (each one associated with one channel) of $2^{(K-MAX\_SIEVE\_SIZE)} \times 2^{(K-MAX\_SIEVE\_SIZE)}$. These points comprise a sub-window of wavelet coefficients belonging in the corresponding channel, and the position of this sub-window, as defined by its upper left point, is exactly the point in the window under consideration associated with the D$_0$ channel point comprising the sieve size say, $r = 3$. For instance, concerning the three-level DWT of Fig. 4.5, each channel comprised of their respective sieve structure size, which are windows of wavelet coefficients. Each such window includes $2^{(3-MAX\_SIEVE\_SIZE)} \times 2^{(3-MAX\_SIEVE\_SIZE)}$ of wavelet coefficients. For D$_0$, H$_3$, V$_3$, and D3 the max sieve size $r = 3$ and, thus, 1 DWT coefficient is considered. For H$_2$, V$_2$, and D2 the maximum sieve size $r = 2$ and, thus, 2 * 2 DWT coefficients are considered. Finally, for H$_1$, V$_1$, and D1 the maximum sieve size $r = 1$ and, thus, 4 * 4 DWT coefficients are considered.

The practical aspects of the above proposed feature extraction approach, are presented:

Step 1:

It is experimented with 256×256 images and it is found that W = 64 is a good size for a sliding window raster scanning them and capable of locating defective areas.

Step 2:

A two-level 2-D DWT wavelet decomposition of these sliding windows associated images has been performed for each such window, resulting in seven main wavelet channels.

Step 3:

The above leads to having 4 × 1 + 3 × 4 = 16 wavelet coefficients as components that span the 2-Level wavelet domain.

Thus, using the above in detail outlined feature extraction procedure, it has obtained 16 feature input vectors efficiently describing spatial distribution in the wavelet domain of each 64 × 64 sliding window raster scanning the images.

4.3.3.2 Application into Semiconductor Images

The efficiency of this approach in recognizing defects in automated inspection images, based on utilizing wavelet domain information, is illustrated by applying it to the semiconductor images shown in Figures 4.6 (a) and 4.7(a), which contain various types of defective areas. Two other rival feature extraction methodologies are applied to these images too. The former of them uses all the 64 × 64 wavelet coefficients obtained by the 2-D DWT transformation of each 64 × 64 sliding window without any further processing, while the latter uses the 64 × 64 image intensities corresponding to the same sliding window. Therefore, the first feature extraction procedure used in this experimental study is the suggested novel one outlined in Section 4.3.3.1, which involves 16 components features.

The two images shown in Figures 4.6(a) and 4.7(a) are have (430 × 304) and (432 × 432) dimensions and their associated 2-Level 2-D DWT are shown in Figures 4.6(b) and 4.7(b) respectively. The channels shown in these figures have been obtained through applying the 2-D DWT with Haar wavelet bases to the original images. Obviously, the defective areas are preserved and enhanced in the corresponding wavelet domains and this explains the selection of the 2-D DWT as the baseline for the herein presented feature extraction methodology. There exist 25312 sliding windows of 64 × 64 size for each original image. The extraction procedures used in this study are applied to every such sliding window, yielding the corresponding features.

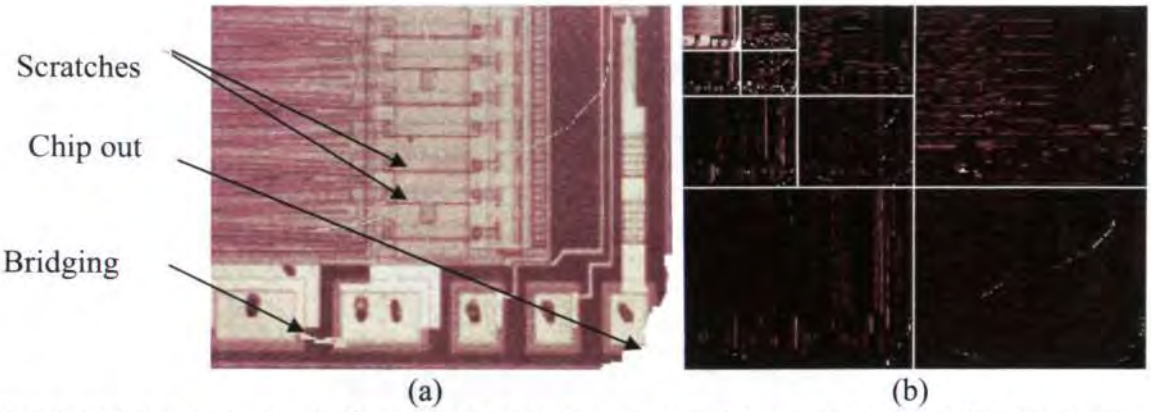


Figure 4.6: (a) First original semiconductor image containing a defect and (b) 2-Level 2- D Wavelet transformation of this image.

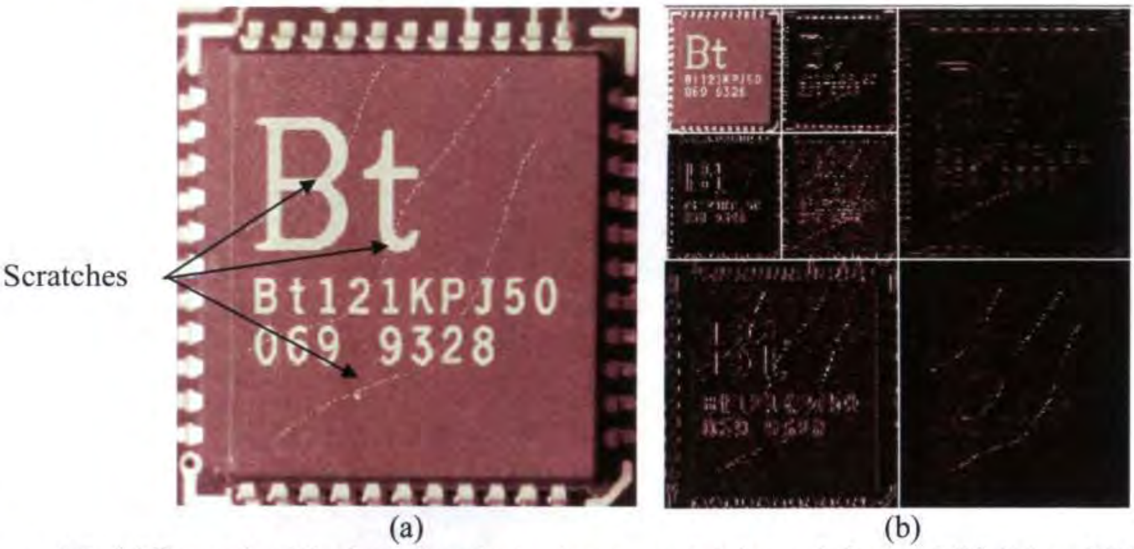


Figure 4.7: (a) Second original semiconductor image containing a defect and (b) 2-Level 2-D Wavelet transformation of this image.

The results obtained for the semiconductor image in Fig. 4.6 (a) are shown in Figures 4.8, 4.9, and 4.10 similarly the result obtained for the semiconductor image in Fig.4.7 (a) is shown in Figure 4.11 using the 2-Level 2-D Wavelet methodology and clearly are very favorably compared, in terms of defect classification performance.

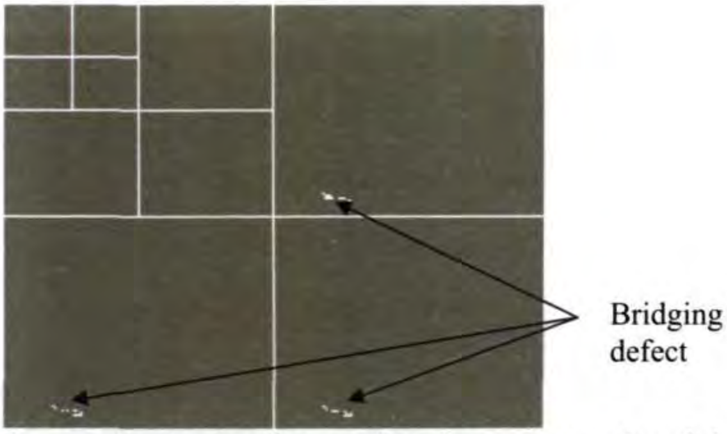


Figure 4.8: Defect detection results for the first semiconductor image. Level 1 shows the actual bridging defect found in the original image obtained using the 2-Level 2-D Wavelet feature extraction method.

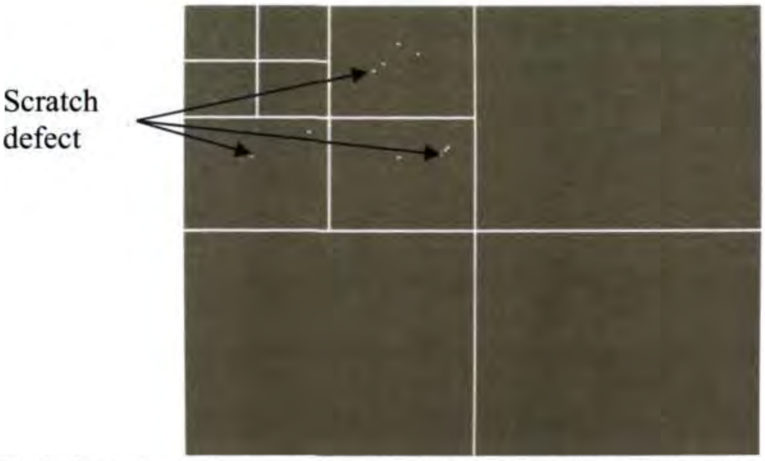


Figure 4.9: Defect detection results for the first semiconductor image. Level 2 shows the actual scratch defect found in the original image obtained using the 2-Level 2-D Wavelet feature extraction method.

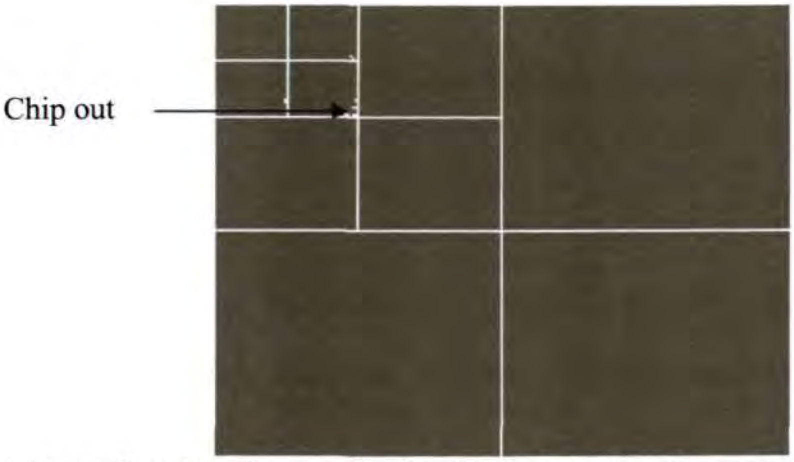


Figure 4.10: Defect detection results for the first semiconductor image. Level 3 shows the chip out defect found in the original image obtained using the 2-Level 2-D Wavelet feature extraction method.

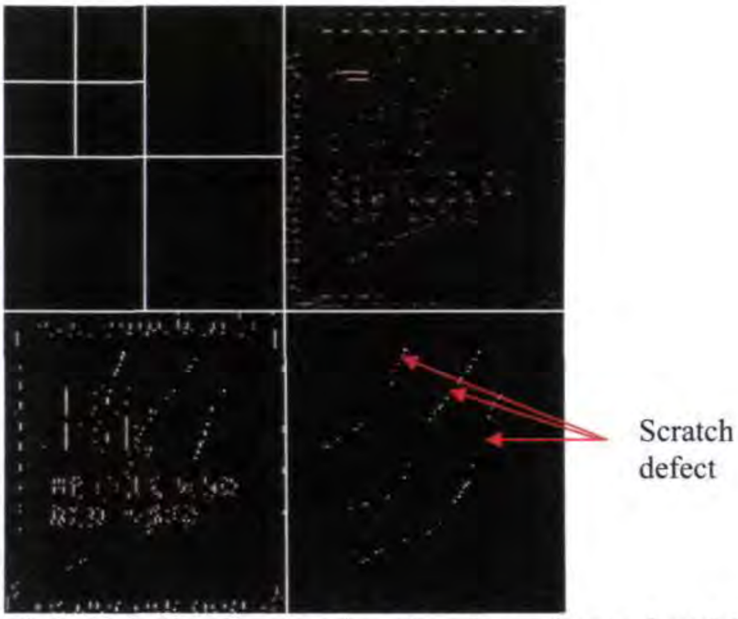


Figure 4.11: Defect detection results for the second semiconductor image. Level 1 shows the actual defect found in the original image obtained using the feature extraction method.

4.4 Constrained clustering

A pixel, in itself, has little visual meaning. Several isolated pixels may be described by a rich set of features, but may not provide us with meaningful information. Instead, if the pixels are adjacent to each other, their combined properties reveal the structure of the pattern.

In this subsection, four of the most known techniques will be discussed, namely Region Growing, Split & Merge, Pyramidal and Morphological techniques. For each method the basic principle will be explained and an example will be discussed. The advantages and drawbacks of these techniques will be also evaluated in terms of complexity, flexibility and accuracy of the results.

It may be important to notice that most techniques that are presented in this section are seldom considered as clustering techniques. This is due to the fact that their strategy relies on the relative positions of the input data. This element may be introduced in a computer vision and pattern recognition framework, but it is not present in other applications such as medicine or mathematics, where it is meaningless to present the data in a specific order.

However, even if algorithms such as Region Growing or Split & Merge are mostly known as techniques for image segmentation, they share the same principles of more general clustering techniques.

4.4.1 Region growing

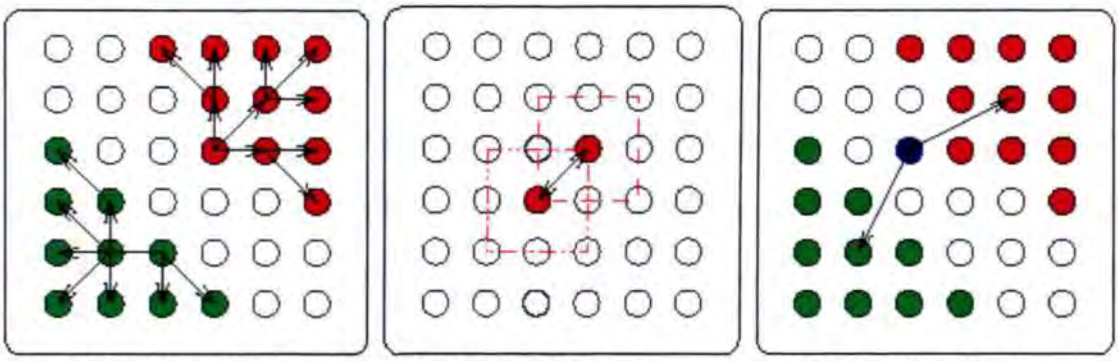
Region growing is one of the simplest approaches to image segmentation. The strategy of this technique finds similarities with an expanding ink stain: given a starting point within the image, neighboring pixels with similar properties are grouped together to form a region. In terms of successive steps, the procedure can be summarized as follow [88].

Initialization: This step requires the selection of a certain number of points in the image. The selected points are associated to different labels and represent the origins from which the growing process starts. They are the seeds of the regions that will constitute the final segmentation. For this reason, the initial points need to be chosen wisely in order to obtain a satisfactory segmentation results.

Search: For each starting point defined by the initialization step, a search procedure is extended to all the neighborhood image positions that have not been associated to any initial label. The purpose of this step is to find for each starting point, the most similar neighborhood pixel according to defined similarity criteria.

Mark: The Mark step decides if the most similar neighborhood found by the search step, is sufficiently similar to the properties of the starting point. In this case the neighborhood pixel is marked with the same label as the corresponding starting point. All marked points are now image positions from which successive Search and Mark steps take place. The iterations of these steps make each starting point grows up to its final shape.

Several other region-growing techniques are reported in literature. They differ in the definition of pixel similarities used in the Search step and in the growing strategies they adopt. As an example, the single-linkage region growing considers the image as a graph where each pixel corresponds to a node. The technique simply consists in linking those neighboring pixels whose properties are sufficiently similar. The result is a set of connected pixels that represent the regions in the image. This simple approach is strongly influenced by the presence of noise in the image features: this is why a pre-filtering process is usually necessary. To further reduce effect of the noise, a hybrid-linkage region growing can be applied. In this case the similarity between two adjacent pixels is computed by taking into account their neighborhood pixel's properties. The use of local neighborhood properties reduces the effect of noise when compared with the simpler single-linkage region growing approach. It is also possible to compare each pixel to the mean properties of existing neighborhood segments.



(a) Single-linkage (b) Hybrid-linkage (c) Centroid-linkage

Figure 4.12: Three different approaches in region growing. (a) The Single-linkage approach, (b) The Hybrid-linkage approach, and (c) The Centroid-linking approach [88].

The segment properties are updated each time a new pixel is linked to it. In Fig. 4.12, the three different approaches are illustrated with schematic examples. However they generally represent a too simple approach to provide good segmentation results on a wide range of images. As a general comment, the main limitation of these techniques is their lack of a global analysis of the input data. In Region Growing, each pixel is compared with its neighborhood: local similarities or dissimilarities due to noise or other effects may be responsible for wrong associations that compromise the quality of the final results.

a) **The Single-linkage approach:** Starting from a given position in the image, the pixel properties are compared with those of its neighborhood. A link is established when the similarity is sufficiently high.

b) **The Hybrid-linkage approach:** The strategy is similar to the Single-linkage procedure, but each pixel is characterized by computing the mean of a small neighborhood. In this example the red square around the pixel.

c) **The Centroid-linking approach:** Here the blue pixel is compared with the global properties of two already existing segments.

As for most segmentation approaches, their performances strongly depend on the choice of proximity measures. Moreover, Region Growing techniques require a threshold in order to state when a pixel is sufficiently similar to one of its neighborhood or to the properties of an adjacent set of pixels. This threshold limits the use of Region Growing techniques in fully automatic segmentation procedures. Another major limit of Region Growing techniques is represented by their sequential nature. The image is scanned in a sequential way, generally from left to right and top to bottom. The scanning strategy may ease the merging of certain pixels simply because they have been encountered first. This dependence may influence the quality of the final result.

Finally the most delicate step of the process is represented by the choice of the initial growing points. These represent the seeding points from which the final segments will grow: a wrong choice is likely to provide a wrong result. To avoid a random definition of the seeding points, it is necessary to analyze the image data. This analysis may already result in a rough segmentation of the image that will be refined through the region growing process.

All the limitations discussed above suggest that Region Growing techniques are probably not the best approach to automatically obtain a segmentation of an image. However, it is important to point out that their strategy is more and more appealing with the improved computing power as an efficient segmentation refinement procedure.

4.4.2 Split & Merge

Split & Merge algorithms require the definition of an initial segmentation of the image. Two opposite actions are used in this technique: a split action and a merge action [90]. The split action is performed on the segments that are not sufficiently uniform. The merge action is applied when two adjacent segments are sufficiently similar. In general the algorithms successively apply the split action till sufficiently uniform features characterize all the segments. Then adjacent segments are recursively merged together in order to obtain larger uniform areas.

If the initial segmentation is not available, an artificial partition is used. The most commonly used is the so-called quad-tree partition. Here the image is decomposed in square blocks of fixed size. Each block contains a number of points that are described by the chosen features. It is possible to measure the attributes of the block by integrating the properties of its points. When a block is not characterized by uniform attributes, it is decomposed in smaller blocks (generally 4 squared sub regions). Once the splitting phase is completed, the merging phase between adjacent blocks takes place. When two adjacent blocks present sufficiently similar properties, they are merged together. Several uniformity and similarity criteria are proposed in literature as described in [155].

Split & Merge algorithms based on quad-tree partitions produce rather blocky segments. On one hand this representation is effective for object-based video coding purposes, as the contour-coding step is not needed anymore and the image is organized in spatially homogeneous regions. On the other hand the visual impact might degrade some object functionalities that represent today one of the key-features of object-based video coding schemes.

Compared with Region Growing techniques, Split & Merge provide a global analysis of the data content. Region Growing techniques use point-to-point or point-to-set proximity measures. Instead, Split & Merge techniques apply set-to-set proximity measures between adjacent entities such as blocks or more in general free-form regions. This strategy reduces the effect of noise if compared with Region.

Region growing techniques on the other hand does not allow the technique to perform a refinement of the regions at a pixel level. As for Region Growing the choice of the proximity measures is crucial. Moreover, Split & Merge requires the definition of two thresholds, one that assesses when two sets are sufficiently similar so as to be merged together and one that says when a set is not sufficiently homogeneous and requires a splitting procedure. The efficiency of Split & Merge techniques may be improved if the initial segmentation is not a fixed quad-tree structure, but a more realistic representation of the visual content. In this case, Split & Merge may represent an efficient solution to improve the segmentation results.

4.4.3 Pyramidal Segmentation

An attempt to combine local and global analysis is represented by pyramidal segmentation methods. They consist of four main steps [97]: blurring, linking, root labeling and downward projection. The blurring step is used to create copies of the input image at different larger scales. Often, this process simply consists of applying a Gaussian kernel on the input image and then recursively on the obtained filtered results. This step builds a pyramid whose height might change according to the size of the input image and the size of the used filtering kernel.

The linking step is responsible for defining, starting from the base of the pyramid, a relationship between the elements of a layer and the elements of the layer above. The links are defined according to the similarity between the elements. As for the previous methods, similarities are defined according to the features and the proximity measures. The definition of links between a layer and the layer above may be used to update the properties of the elements. After each update, a new linking step is applied. The procedure ends when no more changes occur in the pyramid or after a maximum number of iterations.

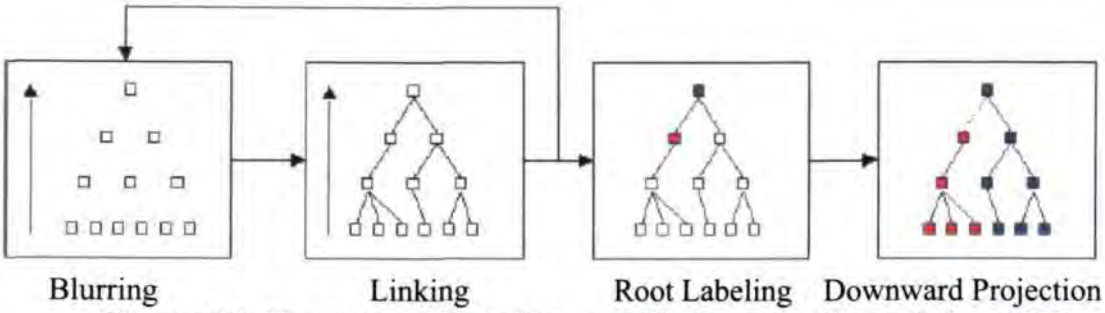


Figure 4.13: The steps performed by pyramidal segmentation technique [97].

The blurring and the linking steps generate a tree-like linkage structure. This is used to define a segmentation of the image that is obtained by applying two steps: root labeling and downward projection. Root labeling defines which element in the pyramid represent a region in the ground level. Downward projection follows all the links from this element down to the ground level and implicitly defines the segmentation. A general scheme of pyramidal segmentation is represented in Fig. 4.13.

When compared with the Region Growing algorithms and the Split & Merge algorithms, the pyramidal approach shows a better use of local and global analysis since it combines the two in the same strategy. On one hand this method starts by grouping single pixels in a local area of the image. This procedure is based on a point to set proximity analysis: thus it is a local analysis. Then, while climbing the pyramid, single pixels are linked into sets. Adjacent sets are then compared in order to assess their proximity. At this level the comparison takes into account the global properties of the sets. Thanks to the linking procedure, a complete set of segmentation results is available in memory. Each level of the pyramid represents a segmentation result. A user could easily select the best according to his judgment. Moreover, it could be possible to define criteria of segmentation quality that would suggest, by comparing the different results, the one that shows the best quality. This approach is particularly interesting when a method to evaluate the segmentation results exists.

However, pyramidal algorithms are usually constrained at the base to a fixed quad-tree structure generated by fixed size windows. This is responsible for the creation of artificial features. An artificial feature is the result of computing the mean of the features belonging to non-homogeneous entities. They might degrade the coherence of the final results.

Another drawback is represented by the complexity of most pyramidal algorithms. These often require more memory resources than previously discussed methods. Also, according to the linking strategy, the final segmentation results may provide disconnected regions, that would require a successive connect component analysis.

Finally, pyramidal algorithms are not sufficiently flexible compared to other approaches. Their bottom-up strategy does not consider the possibility for descending the pyramid and correcting a level according to the results of upper levels. This rigidity makes the procedure difficult to adapt to a general input data.

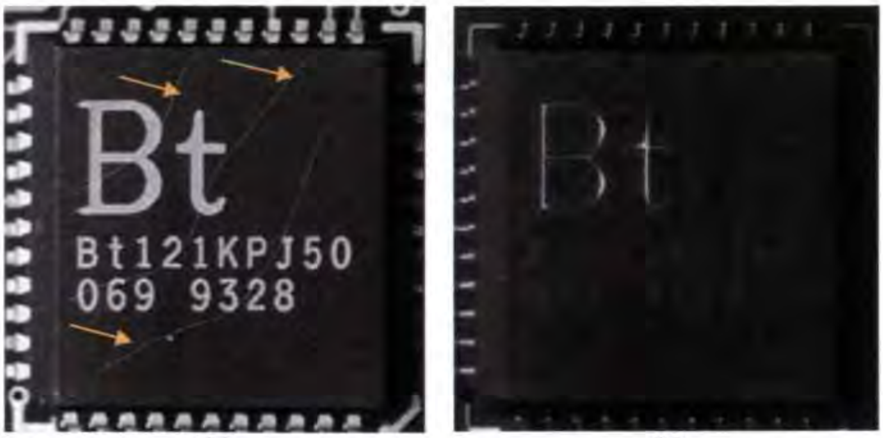
4.4.4 Morphological Segmentation

Another approach to image segmentation is represented by the one based on Mathematical Morphology [156]. This mathematical theory offers a set of tools that can be used for different image analysis tasks from filtering and image simplification to image segmentation. It is very attractive because it represents a geometrical approach to signal processing: it easily deals with criteria such as shape, size, contrast, connectivity, etc. Those are important in the process of interpretation of the visual content.

Mathematical morphology operations can be used to modify image shapes, reduce noise and detect features of interest. Here we consider only binary images and binary morphology. In this case only two different pixel values are possible, often called "foreground" and "background" levels. It is possible to represent a binary image as a set of (row, column) coordinate locations for all of the foreground points. Most morphology operations involve a *structuring element,* which is another set of (row, column) pairs. The structuring element is typically quite small, and its shape has a direct impact on the results. Two fundamental operations of mathematical morphology are *dilation* and *erosion.* Intuitively, these operations tend to enlarge and reduce the sizes of foreground regions in images.

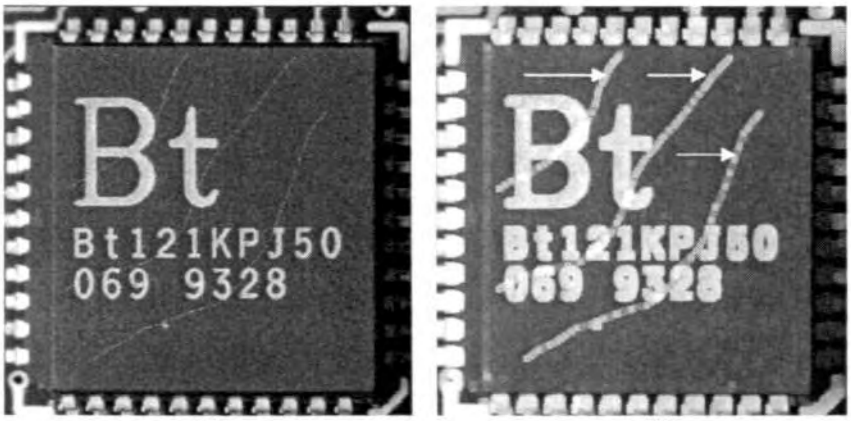
(a) Erosion and dilation

Erosion thins the white objects by removing one layer of pixels along the object edges. When the kernel size gets large, the tiny white objects completely disappear and the black ones get fatter. Dilation is the dual of erosion. Dilation thickens the white objects by adding one layer of pixels along the object edges. When the kernel size gets large, the white objects get fatter and the tiny black ones disappear. Figures 4.14 and 4.15 show examples how both erosion and dilation work on an image.



(a) (b)

Figure 4.14: (a) Image before erosion; (b) Image after erosion.

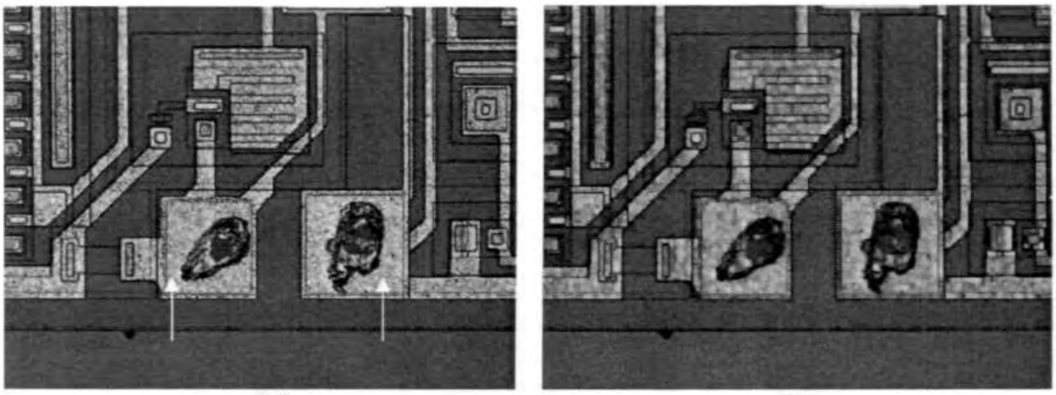


(a) (b)

Figure 4.15: (a) Image before dilation; (b) Image after dilation.

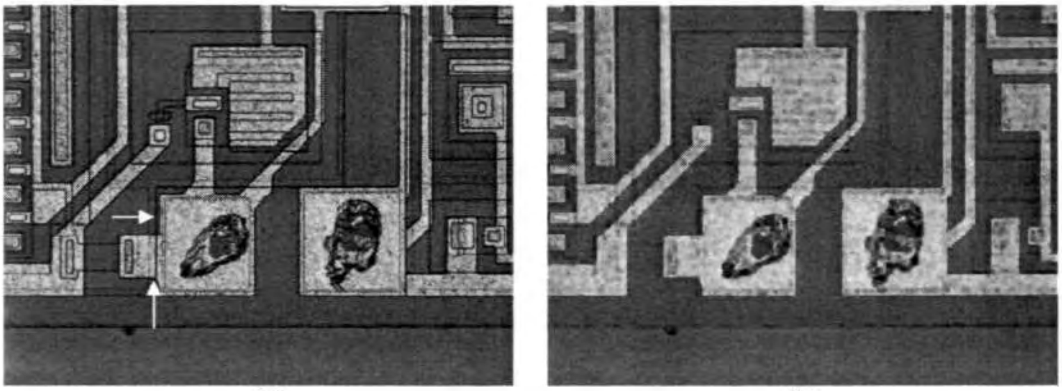
(b) Opening and closing

Opening is erosion followed by dilation. The global effect is to preserve the overall shape of objects, while dropping the white details smaller than the kernel size. This provides an efficient way to remove white dust or other tiny objects.



(a) (b)

Figure 4.16: (a) Test image, and (b) Test image after removal of white dust objects.



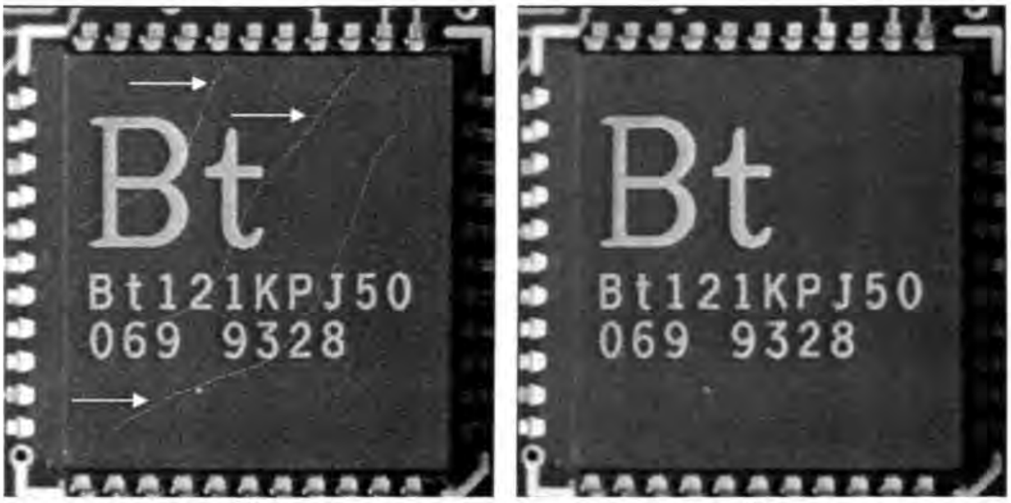
(a) (b)

Figure 4.17: (a) Test image, and (b) Test image after removal of black dust objects.

Closing is the dual of opening. It is a dilation followed by erosion. The global effect is to preserve the overall shape of objects, while dropping the black details smaller than the kernel size. This provides an efficient way to remove black dust or other tiny holes. Figures 4.16 and 4.17 show example of how both close and open work on an image.

(c) Median filter

The median filter replaces every pixel by the median (central value) of its neighbors in a 3 x 3 square kernel. This way, outlier pixels are discarded. The net effect is a very effective removal of impulse noise, while edges and image sharpness are well preserved. Fig. 4.18 shows an example how the median filter works on an image.



(a) (b)

Figure 4.18: (a) Test image, and (b) Test image after removal of scratches and sharp edges.

The morphological operators combine the pixel values in a neighborhood of given shape (square, rectangular or circular) and replace the central pixel of the neighborhood by the resultant shape. The combining function is non-linear, and in most cases it is a rank filter. Consider the N values in the given neighborhood, sort them increasingly and select the K_{th} largest. Three special cases are most often used: K can be 1 (minimum of the set), N

(maximum) or $\frac{N}{2}$ (median). They correspond to the so-called erosion, dilation and median filter operations.

Erosion, dilation, opening, closing and gradient operations performed in this work all use rectangular kernels as shown in Fig. 4.19. It is important to understand how the kernel size is specified and the kernels are kernel half height and kernel half width. The actual kernel width and height are (2 × Kernel Half Width + 1) and (2 × Kernel Half Height + 1). Most of the morphological operations (except median filters) can be applied destructively, i.e. the destination image being the same as the source, or not. The destructive operations are faster.

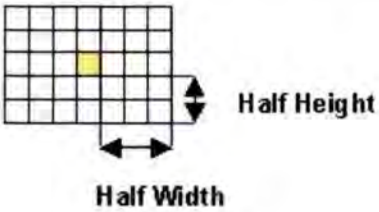


Figure 4.19: Rectangular kernel.

4.4.4.1 Granulometry for Defect's Segmentation

There are many more applications that morphology can be applied to. Morphology has been widely researched for use in image and video processing, but with no application to defect separation. One of the properties not utilized for defect separation so far is that of 'Granules'. A granule is an element that falls through a 'Sieve' in the same way that small stones fall through a sieve [122, 157]. If a set of structuring elements is used with increasing size, $\{B(r): r > 0\}$ with the property that $B(t)$ is $B(r)$ opened for $t \geq r$. This just means that each structuring element is bigger than the previous. Using this, an opening can be defined to use these structuring elements:

$$\alpha_r(E) = X \circ B(r) \tag{4.3}$$

where, α_r is the area sieve with scale factor r, E is the error image, B is the structuring element, $\circ$ is the open operator and t is the granule target size. If equation (4.3) satisfies the property that $\alpha_r\alpha_t = \alpha_t\alpha_r = \alpha_t$ where $t \geq r$, then the openings are called a '**Granulometry**'. The morphology techniques have already proven that it is good for filtering. However, it is sometimes found that the images show a more noticeable degradation than others. This is due to the fact that the filters operate on the entire image having no knowledge of the image statistics. To overcome this, a filter that will act on the entire image, but that only changes local minimum and maximum point is needed. The method used to describe such a filter is known as 'Area Morphology'.

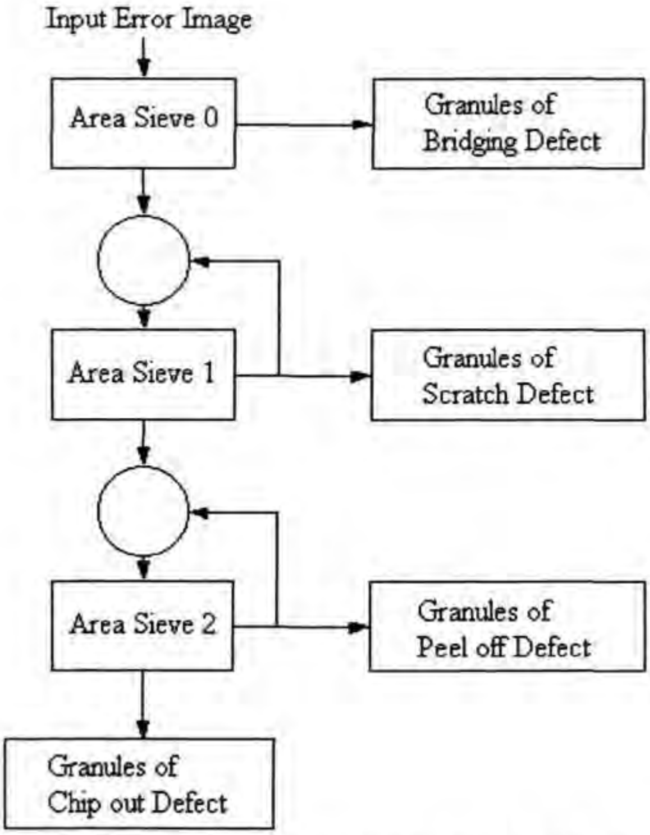

Input Error Image
Area Sieve 0
Granules of Bridging Defect
Area Sieve 1
Granules of Scratch Defect
Area Sieve 2
Granules of Peel off Defect
Granules of Chip out Defect

Figure 4.20: Example of a 2D sieve structure using area morphology.

It is difficult to see from the above definition, exactly what Granulometry does. The opening used above is good, but a more useful method would be to use Open-Close, which will remove both dark and light spots. Thus, an Open-Close can be defined for a sieve as:

$$S_r(E,B) = (E \circ B_r) \bullet B_r \tag{4.4}$$

where, r is the size of sieve and $\bullet$ is the close operator. From equation (4.4) it is clear that the size will determine the defect data to be removed. For example, if $r = 3$, the corresponding structuring element will remove runs of length 3. However, for a true sieve, this must be done in order. For example, for a target size of 4, the input would first be processed to a size of 1. The result would then be processed to 2 and so on until the target is reached. This is defined as:

$$R_t(E) = S_t(S_{t-1}(...(S_1(E,B_1)...), B_{t-1}), B_t) \tag{4.5}$$

where, t is the target size of sieve. This can be seen more clearly in Fig. 4.20. Hence, granules of the classification system are simply the result subtracted from the input error image and it is expressed as:

$$G_t(E) = S_t(E) - S_{t-1}(E) \tag{4.6}$$

Area morphology can be used to create 2D sieves. The basic sieve structure is used. Figure 4.20 shows how a sieve works and how the defect granules fall through the sieve. Assigning a label to each pixel in the error image the labels are “chip out”, “peel off, scratch”, “bridging”, or “clear die” performs defect pattern classification.

4.4.4.2 Refinement of Chip Out Region Using Rules

The defect classification is performed using rules such that each defect assigns a label ("chipping," "peel off," "scratch", "bridging," or "clear die") to each pixel in the error image. The classification system described here uses higher-level, domain-dependent knowledge to refine initial classification results. There are several motivations for such an approach. First, the initial classification depends primarily on information from very small image neighborhoods, and identifies defects on a pixel-by-pixel basis. It therefore ignores information such as defect shape, size, and position within the die. Second, the rules that we describe here can employ fundamentally different types of rules for different defects, whereas the previous approaches used are restricted to a single topology for all classes that it can identify.

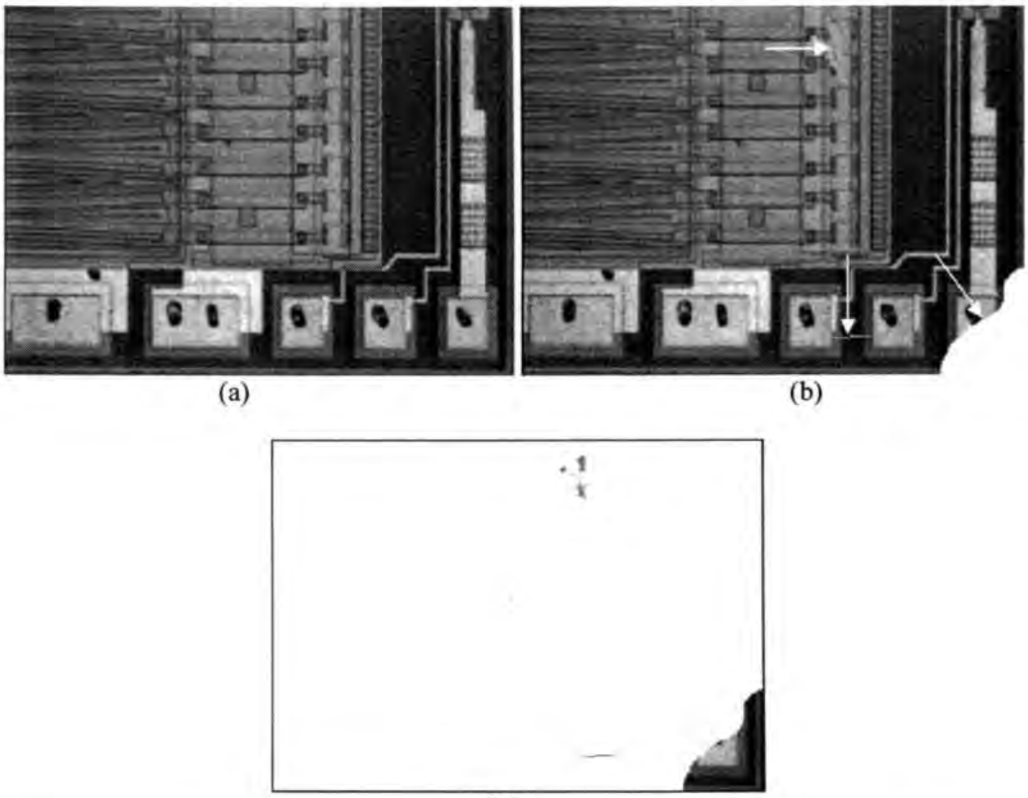

(a)
(b)
(c)

Figure 4.21: (a) reference circuit image, (b) image under test (with defects), and (c) error image.

In order to show how domain-dependent rules can be developed and applied, the experiment begins with an example concerning chip out region of the die under inspection. In this application, it is reasonable to assume that chip out should lie on the outside boundary of a die. It is therefore possible to state this simple rule: Retain only those chip out regions in the image that lie at the outer boundary of the die. The high-level processing strategy is first to determine which point lie on the outside edge of the die, and then to retain only those chip out regions that overlap one or more boundary points.

Reference circuit image in Fig. 4.21 (a) is matched to the image under test in Fig. 4.21 (b) and the resultant image is the error image as shown in Fig. 4.21 (c). Fig. 4.21 (c) contains defects such as chip out, bridging and scratch. Now it is therefore required to separate these defects by applying domain-dependant rules.

Let us first consider morphological operations that can be used to obtain the boundary. A binary die error image in which dark foreground points represent the die boundary is extracted. This binary error image is filtered (median) to remove the two bridging defect components in the error image using a 3 x 3 structuring element and the resultant image is shown in Fig. 4.22. Further the binary error image is dilated using a 3 x 3 structuring element, and points associated with scratches in the image are removed and the resultant error image contains only chip out region is shown in Fig. 4.23 from the dilated version.

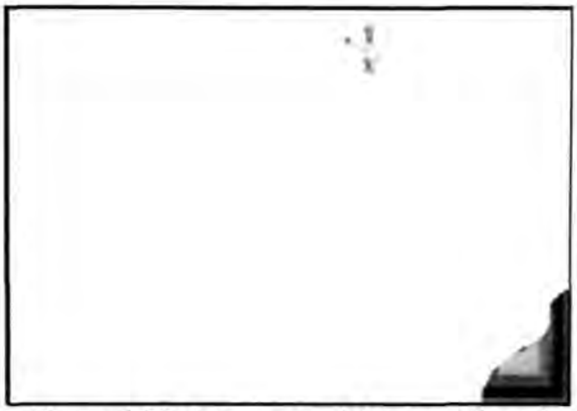


Figure 4.22: Image containing scratch and chip out.

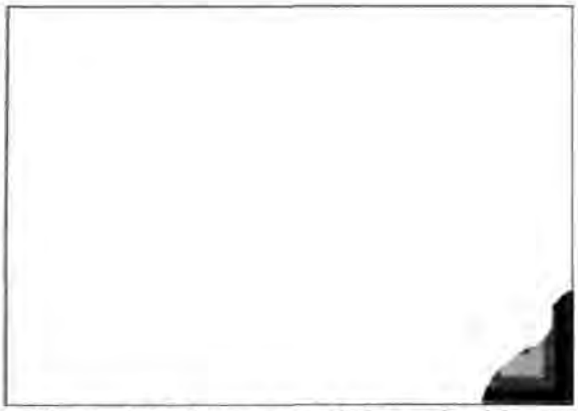


Figure 4.23: Image containing only chip out.

Algebraically, the procedure can be written as

$$C = (E \oplus B) \setminus E \qquad (4.7)$$

where $\oplus$ is the dilation operator, \ is the intersection operator, E is the binary error image, B is the structuring element, and C is the result. The structuring element can be expressed formally as $B = \{(-1, -1), (-1, 0), (-1, 1), (0, -1), (0, 0), (0, 1), (1, -1), (1, 0), (1, 1)\}$. Dilation by this structuring element causes A to be enlarged by one pixel in every direction. The dilations and intersections continue until no change occurs.

4.5 Detection of Semiconductor Defects Using Rule-Based Approach

A template-based vision system for the 100% inspection of wafer die surfaces has been developed. Design goals included a requirement for the detection of flaws as small as two thousandths of an inch on parts up to 8-inch wafer size. Each die is treated as one part of the whole wafer. One of the good dies is trained and kept as template die for the whole wafer. The die physical location data are generated, beforehand, from the wafer map supplied by the wafer manufacturers. A separate software package called the wafer map editor (WME) was developed to generate setup data needed during the runtime of the inspection process. The WME generates an updated wafermap with in-house defects after inspection of the wafer surface. Each unique die pattern in the wafer is defined as an object and these objects are grouped into user-defined categories such as good die pads and defect die pads. The defect wafer map is used during the runtime inspection of die attach to avoid picking of bad dies.

4.5.1 Experimental Setup

The system consists of four hardware modules, namely, the variable Zoom-Optics with CCD camera and adjustable focus, the vertical circular LED light illumination for optimum contrast of all types of dice, high precision linear motor driven X-Y table for ultra-fast and vibration free die presentation, and the real-time image acquisition and analysis platform, Fig. 4.24. In addition, image storage unit is utilized to record the captured images. These images were used to develop the image analysis using Rule-based algorithms. The CCD camera is a progressive scan device with 768 horizontal and 576 vertical pixels. The

CCD camera consists of variable Zoom-Optics and adjustable focus with a magnification of 38 x Zoom-Optics (minimum 5 x) to acquire different die sizes.

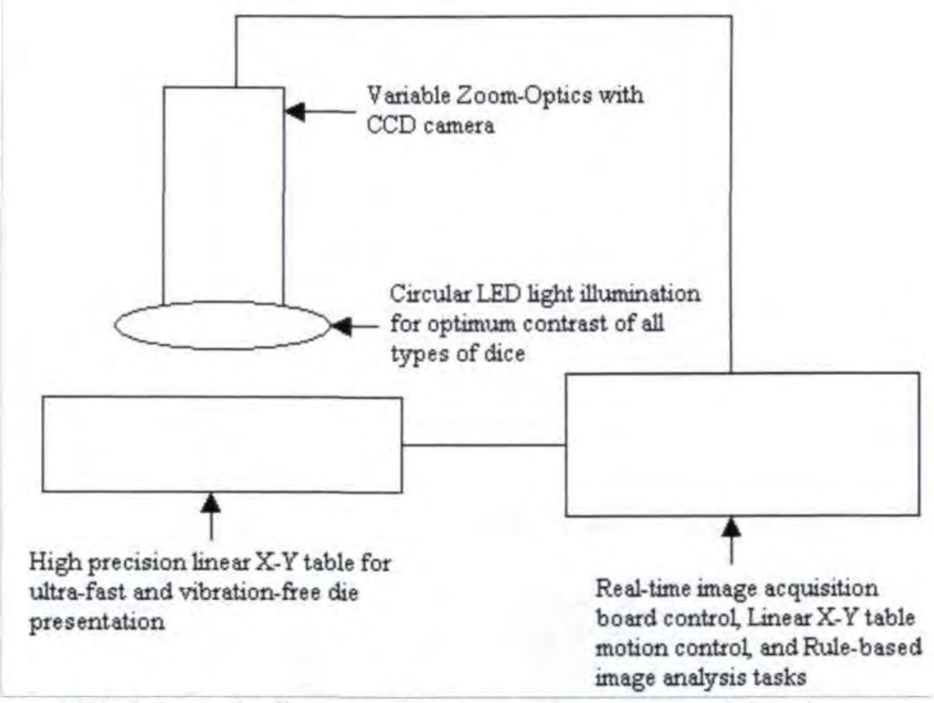

Variable Zoom-Optics with CCD camera
Circular LED light illumination for optimum contrast of all types of dice
High precision linear X-Y table for ultra-fast and vibration-free die presentation
Real-time image acquisition board control, Linear X-Y table motion control, and Rule-based image analysis tasks

Figure 4.24: Schematic diagram of the inspection system and its primary components.

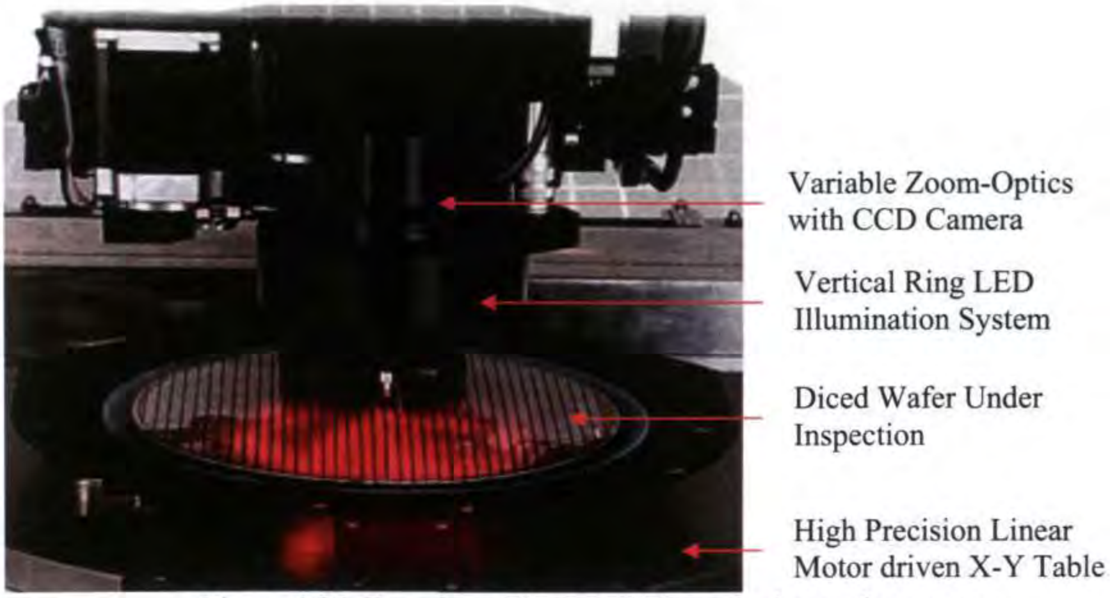

Variable Zoom-Optics with CCD Camera
Vertical Ring LED Illumination System
Diced Wafer Under Inspection
High Precision Linear Motor driven X-Y Table

Figure 4.25: Experimental setup of the wafer inspection system.

The circular LED light is a rugged unit designed to withstand high temperatures. The LED light provides illumination for optimum contrast of all types of dice. The computer system, controlling the above modules, is an industrial PC platform that houses PCI-based image capture and analysis board, linear X-Y table motion control board with pipeline architecture. The image capture board accomplishes all of the data acquisition tasks, in addition to some of the image analysis tasks. Those analysis tasks that cannot be mapped to the particular architecture of this image capture board such as, rule-based image analysis algorithms, and wafer map editor (WME) are assigned to the host processor of the computer. The experimental setup of the wafer inspection system is as shown in Fig. 4.25. The software interface to a new rule-based system is as shown in Fig. 4.26.

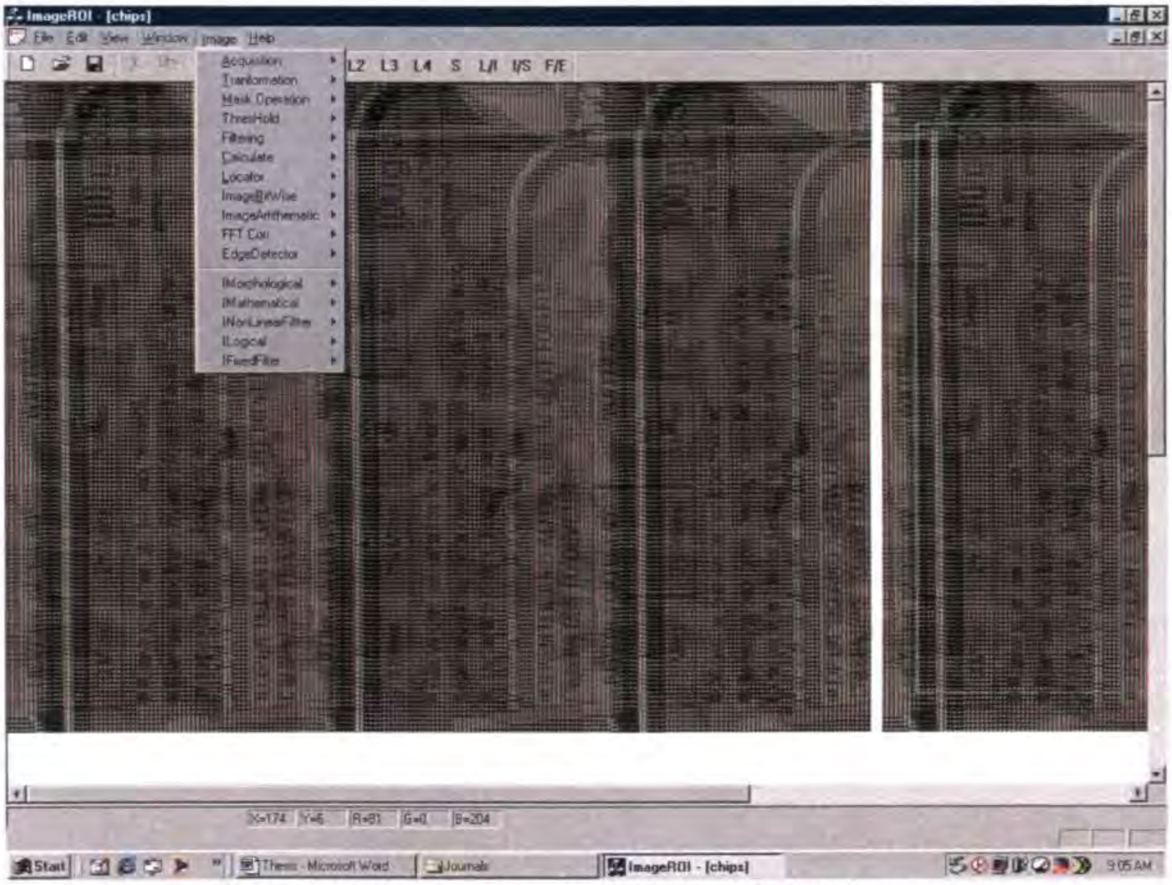
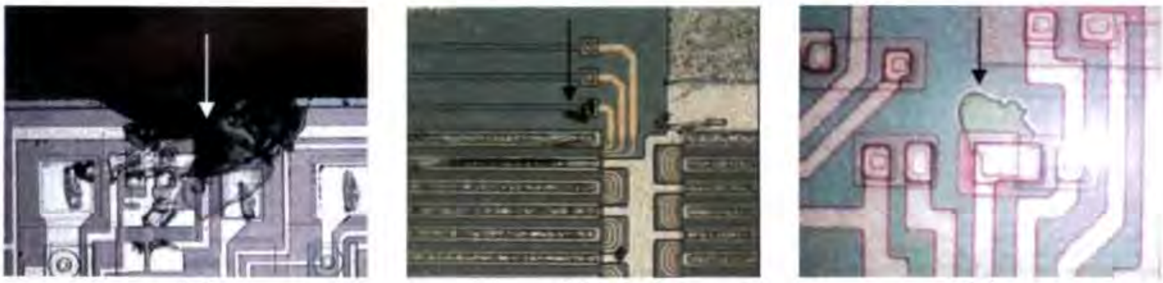


Figure 4.26: Software interface of a new rule-based system.

A separate software package called the wafer map editor (WME) is developed to generate setup data needed during the runtime inspection process. The wafer map editor includes the location of new defects found during inspection, to the original wafermap supplied by the wafer manufacturer. The collection of images shown in Fig. 4.27 provides examples of the acquired defect images.



(a) (b) (c)

Figure 4.27: Images acquired using the above experimental setup (a) chip die, (b) scratches, and (c) diffusion fault.

4.5.1.1 Camera Resolution

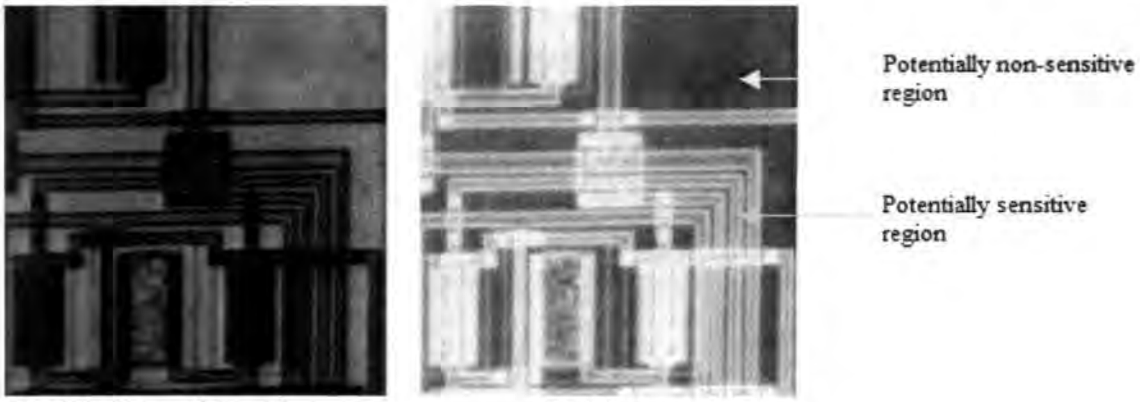
The task is to determine the minimal camera resolution that would be required. Given the size of the die pattern and the relatively tiny size of the potential defects, cost would rise prohibitively if unnecessary resolution was employed. Through a series of experiments and demonstrations it was determined that a 2-mil per pixel resolution could do the required inspection task. Below the Nyquist sampling limit, use of this camera resolution leads to the detection of 2-mil high-contrast defects most of the time [158]. The defect signal rapidly approaches the noise level as the size drops below 2 mils. The possibility of using a line scan camera together with a precision scanning stage is investigated but it is learned that vibration and scanning stage durability became over riding issues. It is decided to use a 768×576 resolution camera mounted on a linear X and Y axis stage to acquire images of wafer under inspection.

Illumination was provided by integrated vertical circular LED ring illumination located together at the front end of the camera. LED illumination system consists of a collection of individual LEDs, providing much greater shape flexibility than other illumination systems. The system sensitivity must be changed for each part type to provide maximum contrast. Camera registration was greatly improved if the image array is adjusted to be as nearly parallel to the plane of the part as possible. A white sheet with an 11 by 15 array of filled circles at 1-inch center-to-center spacing is placed on the inspection area.

4.5.1.2 The Mask Image

The mask image in Fig. 4.28 is derived by edge detection of the reference die (template) beforehand. The absolute difference image is obtained by subtracting the image of the test die from the template and the difference image is multiplied by the mask image to reduce unavoidable pixel differences that arise from variety of sources. When the mask image is black, the non-zero pixel differences are reduced to zero.

The edges of the die are extracted using the canny operator [159]. Edge differences arise from slight misregistrations that exist between the test image and the template. The mask is also used to eliminate acceptable defects in the template since a perfect template in reality does not exist. If the defects are in the potentially sensitive regions they are called as serious defects if the defects are in the potentially non-sensitive regions they are called as acceptable defects. Finally, masking is added to eliminate differences due to other marks that are present as part of the manufacturing process.


Potentially non-sensitive region
Potentially sensitive region

"Golden" template reference

Mask image

Figure 4.28: Shows the extraction of mask image from the reference die pattern.

4.5.1.3 Referential Inspection

The use of the "golden" template reference-based inspection was an obvious choice for this inspection. Images of die parts are precisely aligned with a flawless template and are compared by pixel subtraction. Two images that are identical will yield an image containing all zero pixels. However, any differences between the template and the test image will appear as non-zero pixels. The entire process is summarized in Fig. 4.29.

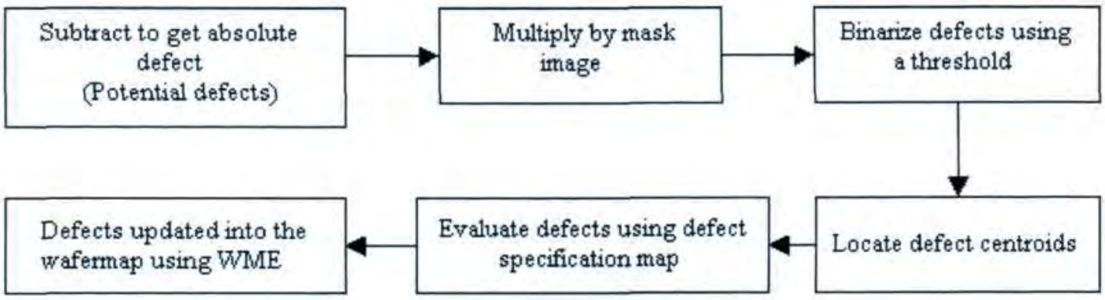

Subtract to get absolute defect (Potential defects)
Multiply by mask image
Binarize defects using a threshold
Locate defect centroids
Evaluate defects using defect specification map
Defects updated into the wafermap using WME

Figure 4.29: Summary of the entire inspection process.

4.5.2 Defects in the semiconductor assembly and the rejection criteria

There are many reasons for poor quality of die surface, which arises due to defects on the die surface. The defects and the rejection criteria in IC assembly after wafer dicing are categorized as follows.

a) Scribing and chipping rejection criteria:

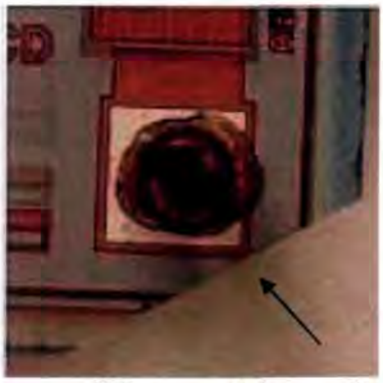


Figure 4.30: Chip out in corner of die extends into active bond pad metallization.

A chip-out or crack (crack > 5.0 μm in length) in the active circuit area is shown in Fig. 4.30 can cause peripheral metallization (one or more layers of microcircuit metal conduction paths) that is at the same potential as the die. Four-sided peripheral metal, that is at the same potential as the die, is rejectable when it exhibits a crack or chip that reduces the metal to less than 50% of its original width and a second defect that totally isolates a contact from the remainder of the peripheral metal.

b) Metallization scratch rejection criteria: A scratch is any tearing defect, including probe marks, in the surface of the metallization. Scratch in the metallization, that exposes underlying passivation (insulating layer directly over a circuit or circuit element to protect the surface from contaminants, moisture, or particles) is shown in Fig. 4.31.

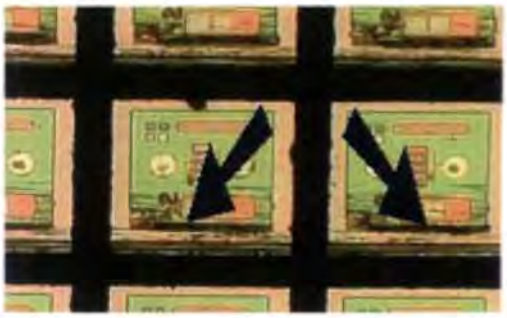


Figure 4.31: The metal scratch in this photograph extends across several adjacent metal runs and has pushed metal such that it bridges two of them.

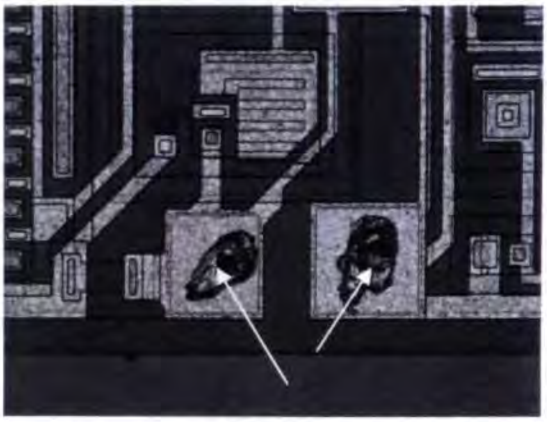


Figure 4.32: The probe marks in this photograph are deep enough to expose underlying passivation and disturb greater than 25% of the unglassivated bond pad.

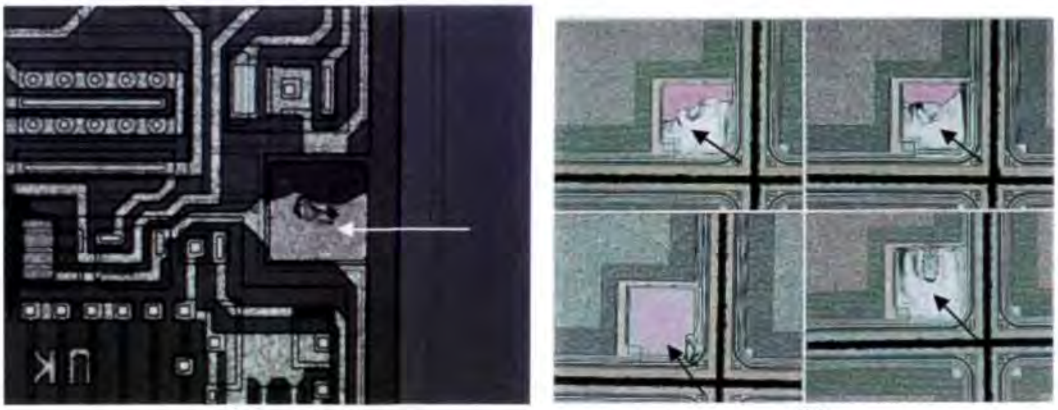
Scratch(es), probe marks in Fig. 4.32, etc. in the bonding pad area that exposes underlying passivation or substrate and leaves less than 75% of the unglassivated metallization area undisturbed.

c) Metallization void rejection criteria: A void is any defect in the metallization where underlying metal or passivation is visible and is not caused by a scratch. The following are some of the metallization void rejection criteria:

i) Void(s) in the metallization that leaves less than 50% of the original metal width undisturbed as shown in Fig. 4.33 (a).

ii) Void(s) in the metallization over a passivation step that leaves less than 75% of the original metal width at the step undisturbed as shown in Fig. 4.33 (b).

vi) Void(s) in the bonding pad or fillet area that reduces the metallization path width connecting the bond to the interconnecting metallization to less than 50% of the narrowest entering interconnect metallization stripe width. If two or more stripes enter a bonding pad, each shall be considered separately.



(a) (b)

Figure 4.33: (a) Void(s) in the metallization that leaves less than 50% of the original metal width undisturbed, and (b) Void(s) in the bonding pad area that leaves less than 75% of its original unglassivated metallization area undisturbed.

d) General metallization rejection criteria:

Any metallization corrosion shown in Fig. 4.34 is rejectable. Metallization corrosion can have various colorations associated with it but is characterized by the "pitting" of the affected metal.

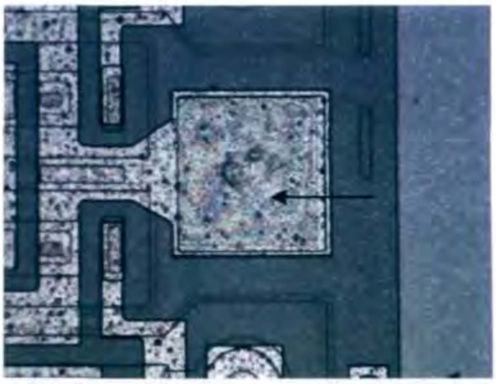


Figure 4.34: Metallization corrosion in the bond pad.

e) Foreign material rejection criteria: Foreign material is defined as any material that does not originate from the microcircuit or any semiconductor material that is displaced from its original or intended position within the microcircuit. The following are some of the material rejection criteria:

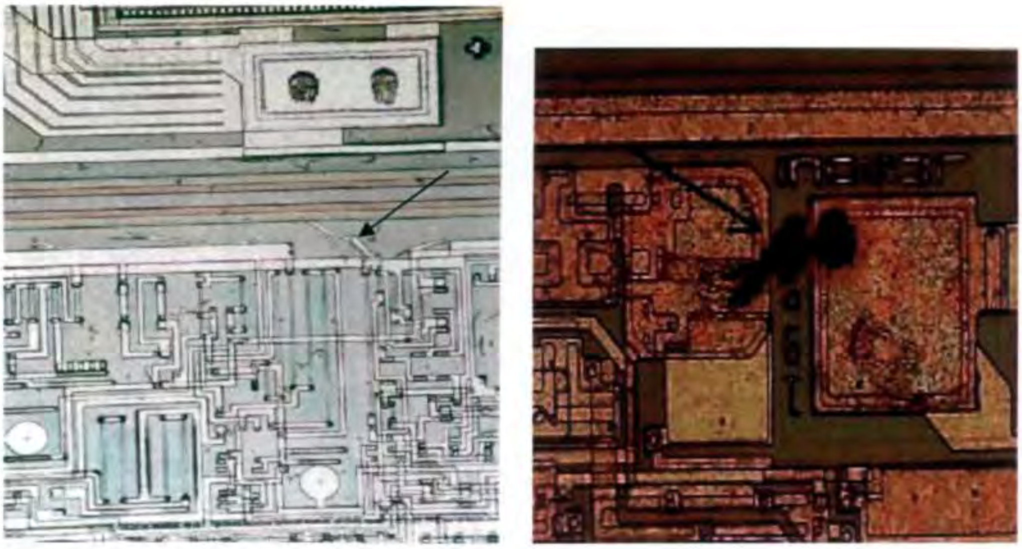


Figure 4.35: Foreign material attached to or embedded in the die surface.

i) Foreign particle(s) on the surface of the die as shown in Fig. 4.35 that is (are) large enough to bridge the narrowest space between unglassivated operating material (metallization, bare silicon, etc.) The narrowest space will be either (a) between bond pads (window edge to window edge) or (b) bond pad window edge to bare silicon (except for metal that is at the same potential as the substrate), whichever is smallest.

ii) Liquid droplets, ink (Fig. 4.36), chemical stains (Fig. 4.37), or photo resist on the die surface that appears to bridge any combination of unglassivated (missing top layer of transparent insulating material that covers active area except for bond pads).

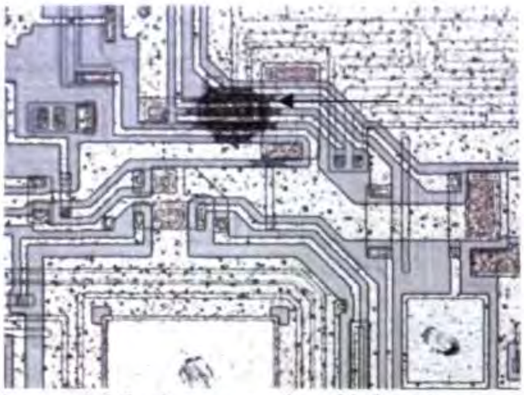


Figure 4.36: An example of ink splatter.

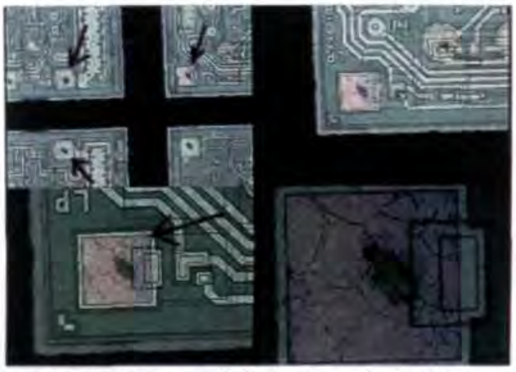


Figure 4.37: Image of defective dark die pads.

f) Passivation fault rejection criteria

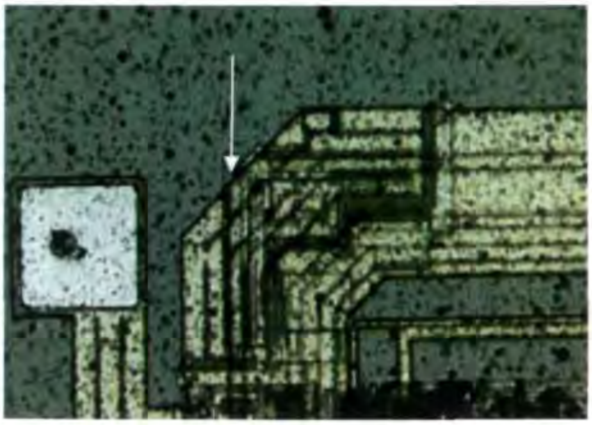


Figure 4.38: Image shows multiple passivation faults existing in both the open oxidation field and under the metallization.

Passivation is an insulating layer that covers a circuit or circuit element to protect the surface from contaminants, moisture, or particles. This fault occurs as either multiple lines or a complete absence of passivation visible (as shown in Fig. 4.38) at the edge and continuing under the metallization. Double or triple lines indicate that the moisture or particles can have sufficient depth to penetrate down to bare silicon.

g) Diffusion fault rejection criteria: Diffusion is a process by which electrical isolation of one or more active circuits is achieved. Diffusion fault (as shown in Fig. 4.39) allows bridging between diffused areas. Any isolation diffusion that is discontinuous (except isolation walls around unused areas or unused bonding pads) or any other diffused area with less than 25% (50% for resistors) of the original diffusion width remaining.

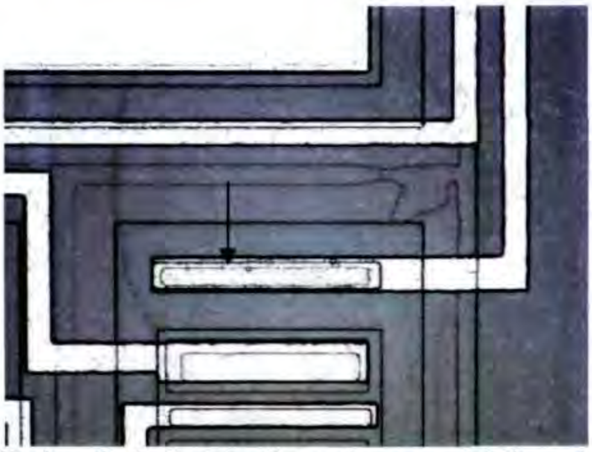


Figure 4.39: Shows diffusion fault that is discontinuous with less than 25% of the original diffusion width remaining.

h) Glassivation rejection criteria: It is a glass crazing that prohibits the detection of visual criteria contained herein. Any lifting or peeling of the glassivation (top layer of transparent insulating material that covers active area except for bond pads) as shown in Fig. 4.40 will lead to damage in active circuit.

In the case of peripheral ground, exposure of metal at the die edge is allowable due to normal etching and alignment, and shall be considered "by design". The maximum width allowed to expose is 50% of the metal path width at its narrowest point, or 5.0 μm, whichever is less.

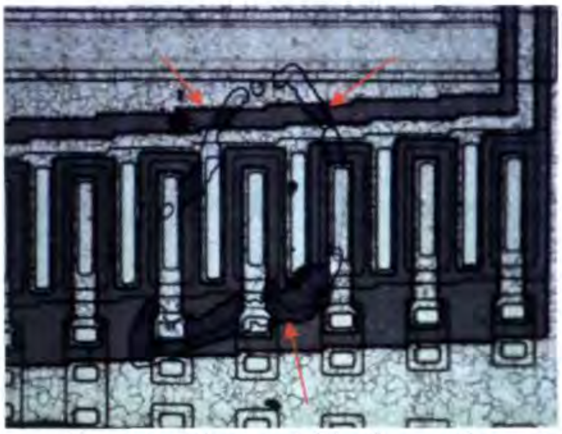


Figure 4.40: Two or more adjacent active metallization paths not covered by glassivation, excluding bonding pad.

i) Bridging rejection criteria:

Material attached to or embedded in the die surface that appears to bridge (Fig. 4.41) the active circuit elements including metallization unless verified as only attached but not embedded by high power dark field illumination.

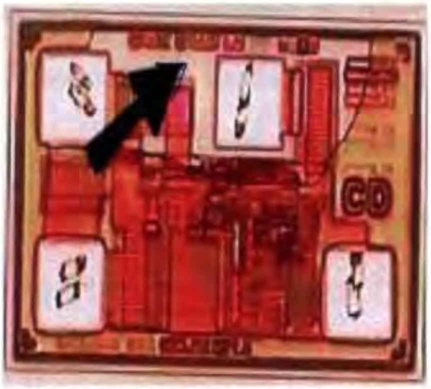


Figure 4.41: Shows an evidence of bridging two distinct aluminum areas.

Hence, it is clear that the defects that arise after wafer die sawing is discarded during die bonding. The goal of this research is to provide an in-line system that can perform 100% inspection of up to 8-inch wafer. The system is required to catch flaws that are unacceptable in the semiconductor industry, and yet allow non-critical variations to pass.

4.5.3 Rule-based Defect Specification

In the previous chapter, a new rule-based has proposed to defect image segmentation, which is based on three successive levels: the diagnostic level, the similarity level and the logical level. Each level is composed of rules that guide the analysis of the defects in the semiconductor chip.

In order to maximize the generality of the method, the rules discussed in Chapter 3 are taking into account the major mechanisms of human vision. The proposed rules have been selected so as to allow the system to simplify the visual data for a wide class of semiconductor images. In addition to these basic rules, new ones can be added to the system in order to adapt it to the needs of specific applications. In this chapter, the proposed method is evaluated on semiconductor defect images.

In this work, it is proposed to select the right set of rules according to the application at hand. The system is thus modularly adaptive to a large set of different applications. Moreover, it is possible to imagine the definition of an automatic application detection that could be used to dynamically adapt the same system to different scenarios.

In order to implement each element of the proposed rules it has taken advantage of well established techniques existing in literature in the domain of sensing modalities and classifiers. These techniques have been evaluated, adapted and modified to better fit in the proposed strategy.

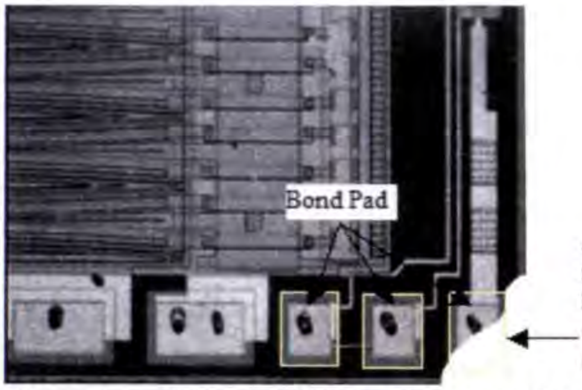
In addition to masking, the system tolerates non-critical flaws by using rule-based defect specification. Differences between the test image and the unmasked template are evaluated with respect to position by using defect wafer map. In the setup phase, the WME is used to assign a number to each object or die in the wafer. The defect map overlays the template so that every pixel of the object or die in the defect map is assigned the same object identifier. When a significant difference "blob" is detected, its centroid is located in the defect wafer map so that its object identifier will be extracted. The centroid of the defect object is located by using the following formula, where P_i is the pixel value

$$\hat{X} = \frac{\sum P_i X_i}{\sum P_i} \qquad (4.8)$$

The identifier then serves as a key to lookup the defect specification rule for the wafer die image. For example, if the defect were located in a bond pad of the die surface, the rule would probably indicate that this defect is significant and should be called to the operator's attention. The rule would simply state that one or more defects of any size found in a bond pad of this type should be considered to be significant. On the other-hand, a small void located in a die bond pad region would be registered for that particular die bond pad but would otherwise be ignored. For the sake of this example, the rule would state that up to three voids are allowed as long as the accumulated area does not exceed 10% of the die area (error image area).

The system then calculates the position and orientation information from an image of the dot pattern acquired by the camera. In particular, the distance of the camera from the sheet and the vertical off-axis angle are used to ensure that the camera is aligned within tolerance for the particular image under consideration [160].

The software interface tool in Fig.4.38 allows the operator to extract the various part structural object features from the error image after similarity level. Our intent was to make this process operator friendly. The interface tool provides the user with a combination of automatic feature extraction tools. The step-by-step application of these tools is recorded into scripts which can be replayed during later editing sessions. The interactive editing operations need only be applied to the component of each multi-component objects. Prompt messages can be used to instruct the user on the next operation to be executed in the overall defect wafer map generation sequence. In general each object positions (e.g. chip out, bond pads, scratches, bridging, metallization, etc.) is extracted from the error image, which is saved by the tool for later processing. Each object image is called a "layer" and will ultimately be assigned its own defect specification rule. Figure 4.42 illustrates use of the rectangular region-of-interest (ROI) extraction tool.



A rectangular region-of-interest (yellow line) is placed around the object that is to be extracted

Figure 4.42: The region-of-interest (ROI) object extraction tool to extract bond pad positions.

An example of the sequence of steps that would be used to extract the object of a defect pattern is shown below in Figure 4.43. Once the interface tool has extracted the object and the operator has entered a defect specification rule for each object, the interface tool reassembles them into the defect wafer map and the object to rule lookup table. Rules are assigned using the interface tool menu driven dialog box. In the thesis already it is mentioned that as per the requirement the proposed rules have been selected so as to allow the system to simplify the visual data for a wide class of semiconductor images. In addition to the basic rules, new ones can be added to the system in order to adapt it to the needs of specific applications. For example if the defect is in the bond pad as shown in Figure 4.42. The textural energy feature for the bond pad region are calculated as,

$$f_{BP}(d) = \sum_{i}^{m} \sum_{j}^{n} s(i,j,d)^2 \tag{4.9}$$

For the sake of this example the rule will indicate that the combined bond pad error texture area should be less than 10% of the total texture area of the error image.

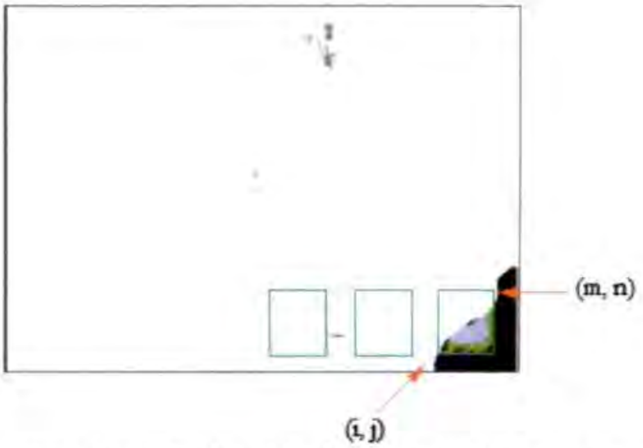


Figure 4.43: Feature extraction from the bond bad region of the error image after similarity level.

4.6 The Wafer Map Editor

The WME also provides the runtime editing of the original wafermap with most of the required preprocessing data. This data is derived from the wafermap supplied along with the wafers. The information generated by the WME is summarized in Fig. 4.44. Processing of the wafermap to a binary map is performed using visual basic software and from the binary map it is known that the good die pattern is marked as binary one (1) and the bad die pattern is marked as binary zero (0). Once, a die pattern is considered as defect using mean square error then the particular die is marked as binary zero using wafer map editor dialog screen.

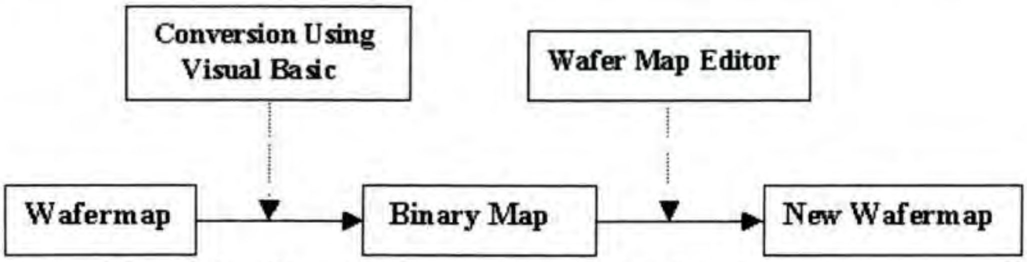


Figure 4.44: The Wafer Map Editor provides the runtime inspection to generate new wafermap with part specific data.

4.6.1 Sum of Squares Due to Error

The measurement method uses a 2-D discrete cosine transform (DCT) mean square analysis. The reference circuit image (template) and the test image are transformed in 8-by-8 blocks. Let $Y = dct2(X)$ returns the discrete cosine transform of X, where $dct2$ is the 2D version of discrete cosine transform. The matrix Y is the same size as X and contains the discrete cosine transform coefficients. $Y = dct2(X,[M,N])$ or $Y = dct2(X,M,N)$ quantizes matrix X with zeros to size M-by-N before transforming. If M or N is smaller than the corresponding dimension of X, $dct2$ truncates X.

This statistic measures the total deviation of the test image values from the reference image response values. It is also called the summed square of residuals and is usually labeled as SSE and it is expressed according to [161] as

$$SSE = \sum_{i=1}^{n} w_i (y_i - \bar{y}_i)^2 \tag{4.10}$$

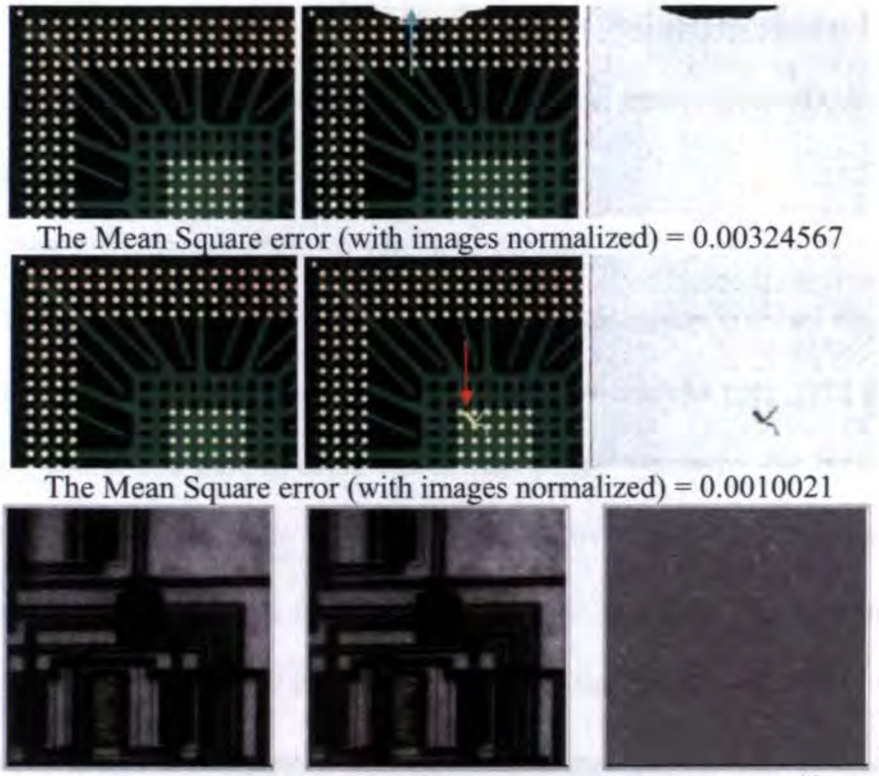
where n is the number of data points included in the image. The error for the i_{th} data point is defined as the difference between the test image response value y_i and the reference image response value $\bar{y}_i$. Then the Mean Square Error is given as

$$MSE = \frac{SSE}{p} \tag{4.11}$$

where p is the number of independent pieces of information involving the n data points that are required to calculate the sum of squares. Usually p is the product of the size of reference image.

4.6.2 Experiment

Reference circuit image is matched to the test image based on the mean square error analysis within the specified range (industry standard 1/10^6). The visual image slides that illustrate the different error images extracted using mean square analysis are shown in Fig. 4.45. The error image i.e., the difference between the reference circuit image and the test image is also displayed.



The Mean Square error (with images normalized) = 0.00324567

The Mean Square error (with images normalized) = 0.0010021

The Mean Square Error (with images normalized) = 1x 10^{-16}

Reference circuit image Image under test Error image

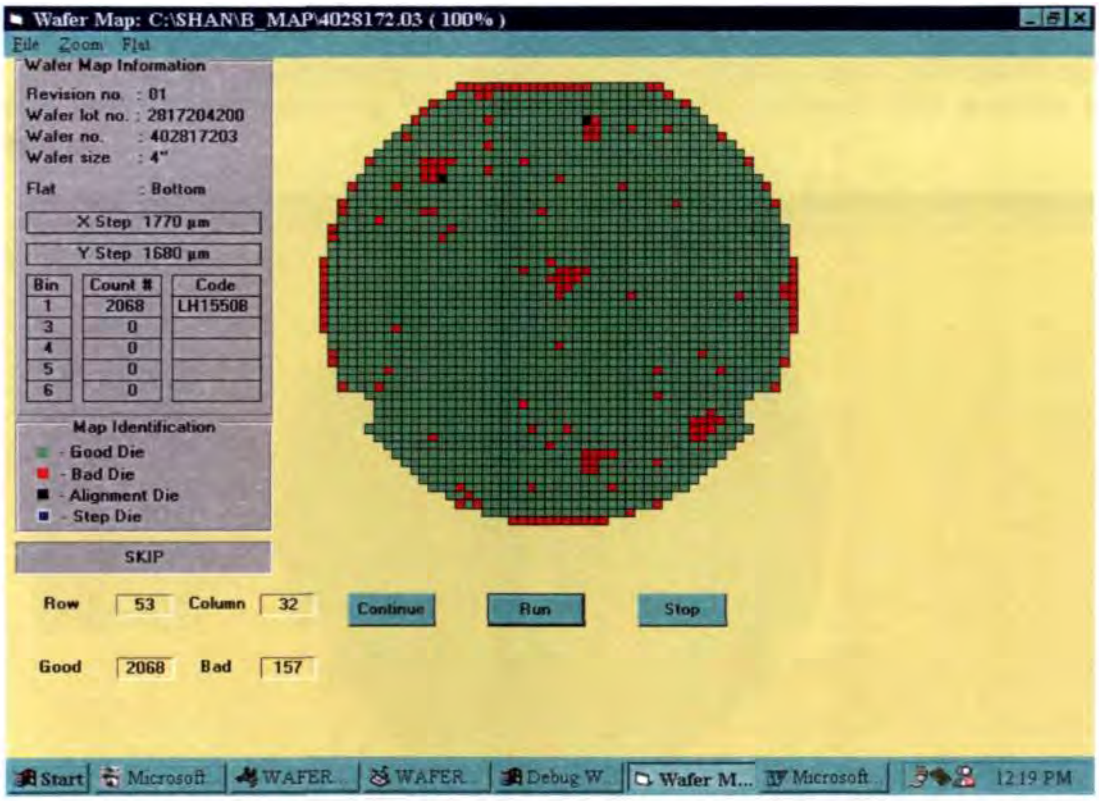
Figure 4.45: The visual image slides that illustrate the different error images extracted using mean square error principle.

Table 4.1: Testing parameters and results of application of the in-line inspection to a variety of data sets.

Wafer size	100 mm	125 mm	150 mm	200 mm
Die size (Square)	0.2 mm - 18 mm	0.2 mm – 18 mm	0.2 mm - 22 mm	0.2 mm - 25 mm
Street Width (1mil = 25.4 micron)	2 mils	2 mils	3 mils	3 mils
Wafer thickness	7 mils - 20 mils	7 mils – 20 mils	7 mils - 20 mils	7 mils - 23 mils
Inspection time per wafer	10 – 17 minutes	10 - 17 minutes	12 – 18 minutes	12 - 20 minutes
Die size tested (square)	1.2mm - 12 mm	1.2 mm – 12 mm	1.6 mm – 12 mm	2 mm – 20 mm
Targeted defects	Foreign particles, saw into metal, crack, scratches, chip die, glassivation void, diffusion fault, bridging, metallization peel-off, and discolored bond pad.			

The algorithm runs until the correction between iterations is below a limit value or the mean squared error does not change sufficiently. A maximum number of iterations terminate the algorithm if convergence is not reached. The result is valid if the mean square error is less than the projected error value (10^{-6}). Otherwise the result is invalid and the particular test die is considered a defect die whose position is updated into the wafermap with the help of wafer map editor. Table 4.1 summarizes the testing parameters and results obtained from in-line inspection to a variety of data sets including the die size, inspection time, targeted defect types, and wafer thickness.

The binary wafermap image in Fig. 4.46 contains 2068 good die before inspection (good die is in green color, defective die is in red color, and alignment die in black color) and the corresponding final binary wafermap image in Fig. 4.49 contains only 1978 good die after rule-based inspection and editing using wafer map editor. Similarly, the binary wafermap images in Fig. 4.47 and Fig.4.48 contains 2052 and 1997 good dies respectively before inspection and the corresponding final binary wafermap images in Fig. 4.50 and Fig. 4.51 contains only 2012 and 1925 good dies respectively after rule-based inspection.



Wafer Map Information

Revision no. : 01
Wafer lot no. : 2817204200
Wafer no. : 402817203
Wafer size : 4"
Flat : Bottom

X Step 1770 µm

Y Step 1680 µm

Bin	Count #	Code
1	2068	LH15508
3	0	
4	0	
5	0	
6	0	

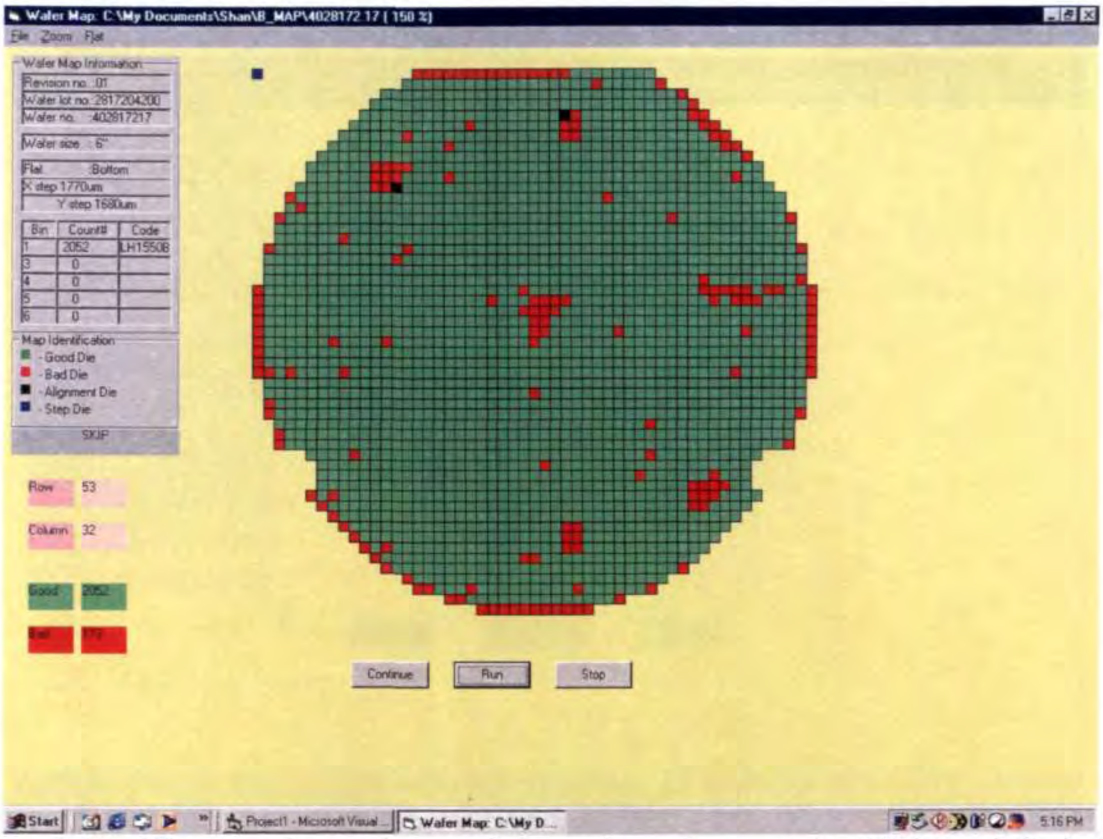
Map Identification

🟩 - Good Die
🟥 - Bad Die
■ - Alignment Die
■ - Step Die

Row 53 Column 32

Good 2068 Bad 157

Figure 4.46: Binary wafermap image of the wafer surface contains 2068 good die before inspection.


Wafer Map: C:\My Documents\Shan\B_MAP\4028172.17 (150 %)
File Zoom Flat
Continue
Run
Stop
Start
Project1 - Microsoft Visual ...
Wafer Map: C:\My D...
5:16 PM

Wafer Map Information

Revision no. :01

Wafer lot no.:2817204200

Wafer no. :402817217

Wafer size : 6"

Flat :Bottom

X step 1770um

Y step 1680um

Bin	Count#	Code
1	2052	LH1550B
3	0	
4	0	
5	0	
6	0	

Map Identification

🟩 - Good Die

🟥 - Bad Die

■ - Alignment Die

■ - Step Die

SKIP

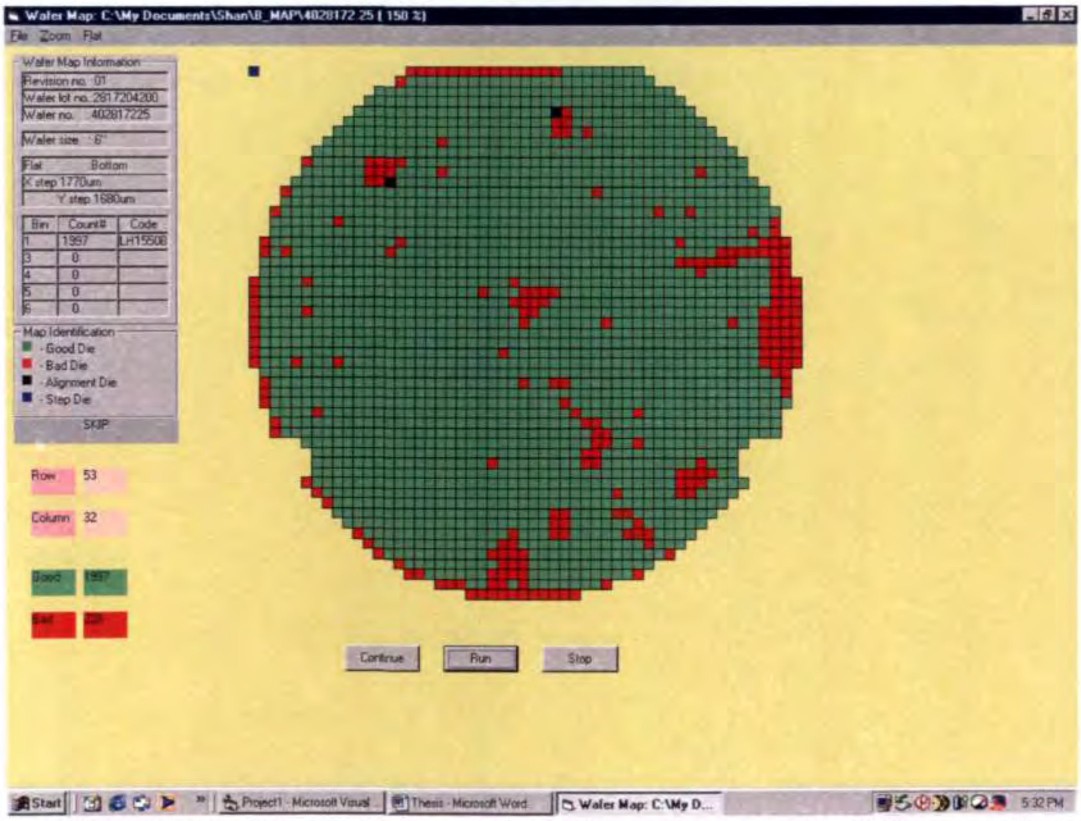
Row 53

Column 32

Good 2052

Bad [illegible]

Figure 4.47: Binary wafermap image of the wafer surface contains 2052 good die before inspection.


Wafer Map: C:\My Documents\Shan\B_MAP\4028172.25 (150 %)
File Zoom Flat
Continue
Run
Stop
Start
Project1 - Microsoft Visual ...
Thesis - Microsoft Word
Wafer Map: C:\My D...
5:32 PM

Wafer Map Information

Revision no. :01

Wafer lot no.:2817204200

Wafer no. :402817225

Wafer size : 6"

Flat :Bottom

X step 1770um

Y step 1680um

Bin	Count#	Code
1	1997	LH1550B
3	0	
4	0	
5	0	
6	0	

Map Identification

🟩 - Good Die

🟥 - Bad Die

■ - Alignment Die

■ - Step Die

SKIP

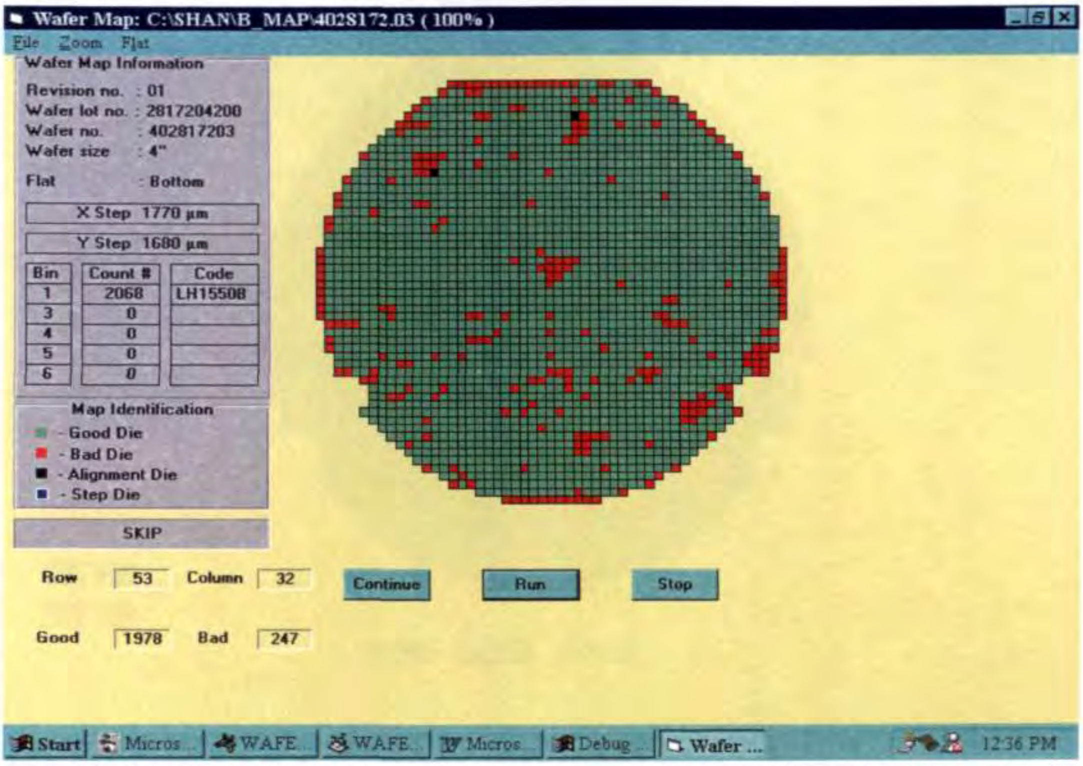
Row 53

Column 32

Good 1997

Bad [illegible]

Figure 4.48: Binary wafermap image of the wafer surface contains 1997 good die before inspection.



Wafer Map: C:\SHAN\B_MAP\4028172.03 (100%)

Wafer Map Information

Revision no. : 01
Wafer lot no. : 2817204200
Wafer no. : 402817203
Wafer size : 4"

Flat : Bottom

X Step 1770 µm

Y Step 1680 µm

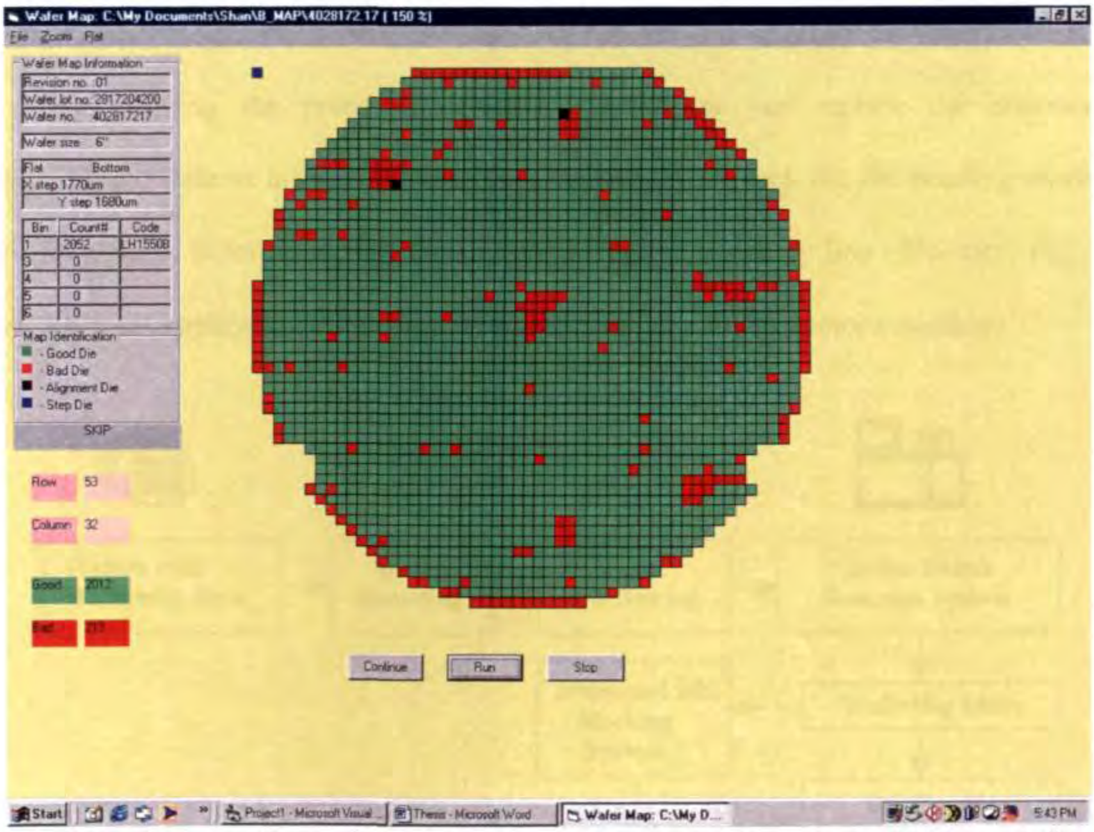
Bin	Count #	Code
1	2068	LH15508
3	0	
4	0	
5	0	
6	0	

Map Identification
🟩 - Good Die
🟥 - Bad Die
■ - Alignment Die
■ - Step Die

Row 53 Column 32

Good 1978 Bad 247

Figure 4.49: Binary wafermap image of the wafer surface contains only 1978 good die after rule-based inspection and editing.



Wafer Map: C:\My Documents\Shan\B_MAP\4028172.17 (150 %)

Wafer Map Information

Revision no. :01
Wafer lot no. 2817204200
Wafer no. 402817217

Wafer size 6"

Flat Bottom
X step 1770um
Y step 1680um

Bin	Count#	Code
1	2052	LH15508
3	0	
4	0	
5	0	
6	0	

Map Identification
🟩 - Good Die
🟥 - Bad Die
■ - Alignment Die
■ - Step Die

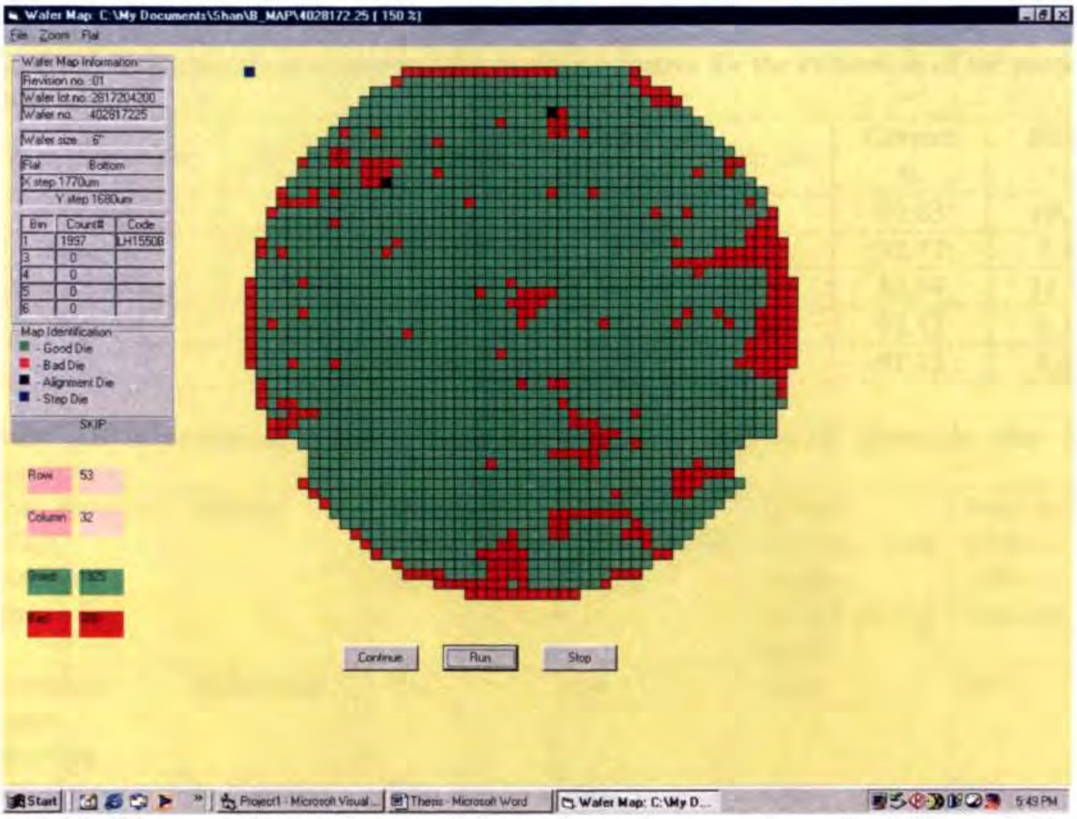
Row 53

Column 32

Good 2012

Bad 217

Figure 4.50: Binary wafermap image of the wafer surface contains only 2012 good die after rule-based inspection and editing.



Wafer Map: C:\My Documents\Shan\B_MAP\4028172.25 (150 %)

Wafer Map Information

Revision no. :01

Wafer lot no :2817204200

Wafer no. 402817225

Wafer size : 6"

Flat Bottom

X step 1770um

Y step 1680um

Bin	Count#	Code
1	1997	LH1550B
3	0	
4	0	
5	0	
6	0	

Map Identification

🟩 - Good Die

🟥 - Bad Die

■ - Alignment Die

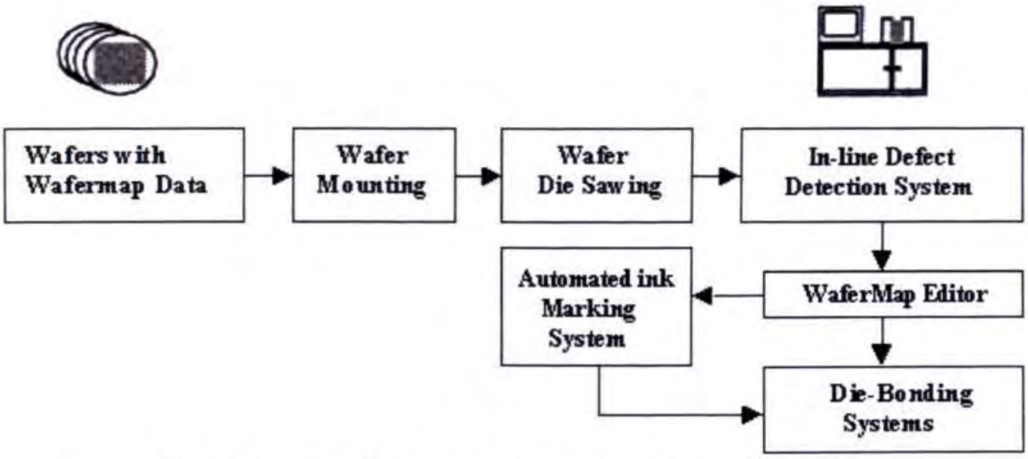
■ - Step Die

Row 53

Column 32

Figure 4.51: Binary wafermap image of the wafer surface contains only 1925 good die after rule-based inspection and editing.

By utilizing the proposed detection system one can replace the unnecessary verification for defects in die-bond machines (at present for each die the bonding machines check for defects before bonding) thereby improving the assembly line efficiency. Fig. 4.52 shows a typical application of the proposed system in the semiconductor assembly.



Wafers with Wafermap Data
Wafer Mounting
Wafer Die Sawing
In-line Defect Detection System
Automated ink Marking System
WaferMap Editor
Die-Bonding Systems

Figure 4.52: Typical application of the proposed defect detection system.

Table 4.2: The classification rate and the confusion matrix for the evaluation of the proposed method.

True\Estimation	Bridging	Scratch	Metal Lifting	Chip out	Correct %	Error %
Bridging	242	12	5	11	89.63	10.37
Scratch	5	308	4	15	92.77	7.23
Metal Lifting	2	14	242	15	88.64	11.36
Chip out	8	7	11	370	93.43	6.57
Average rates					91.12	8.88

Table 4.3: Comparison table showing advantages of rule-based approach over other approaches

Semiconductor Defect Detection Approaches	Method	Detection before wafer dicing	Detection after wafer dicing	Collect defects from single manufacturing source	Provide final solution before bonding
Automatic Defect Detection	Referential	Yes	No	Yes	No
Spatial Signature Analysis	Referential	Yes	No	Yes	No
Dark-field Imaging	Referential	Yes	No	Yes	No
Digital Holography	Referential	Yes	No	Yes (Only etched high aspect ratio defects)	No
Rule-based	Referential	Yes	Yes	Multiple source	Yes, using wafer map editor
NRDD	Non-referential	Yes	No	Yes	No

A collection of defect images obtained from the same process layer of a product was used for evaluating the proposed system. The set consisted of 270 bridging defect class, 332 scratch class, 273 peel off class, and 396 chip die class images. The classification rates were evaluated and verified using manual inspection. The reason why manual inspection is used for verification is that up to date no body has found a final solution to in-line inspection after wafer dicing. The results are shown in Table 4.2. By employing proper logical rules, it is found that the non-diagonal elements in the confusion matrix could be reduced drastically. The obtained classification rate is considered to be comparable to those of human experts. Table 4.3 shows the advantage of rule-based approach over other approaches.

4.7 Summary

The definition of features keeping into account for the perceptual mechanisms of vision increases the amount of data to be analyzed in several Computer Vision and Pattern Recognition applications. A data reduction and simplification is important to make their analysis a practical issue.

This Chapter discussed more specifically the use of clustering techniques in Computer Vision and Pattern Recognition problems. In this context, it is found useful to classify the existing solutions into two categories: the spatially constrained clustering techniques and the spatially unconstrained clustering techniques.

The analysis of few representatives of each category has suggested that partitional clustering offers an extremely efficient and flexible structure. However, its conventional structure can take advantage of specific solutions found in other approaches such as Region Growing, Split & Merge and Morphological segmentation for a post-clustering refinement. This Chapter has evaluated analytically and empirically the segmentation method proposed in Chapter 3.

The analytical evaluation of the proposed simplification process has underlined its generality, modularity, and robustness. These properties have been proved by applying the method with the same configuration on several test images in semiconductor application. The empirical evaluation of the obtained results, based on an independently defined quality measure, has demonstrated the goodness of the obtained results. The relative complexity in terms of computational load is balanced by the important property of being fully automatic.

The visual simplification obtained, is used as the general starting point for tackling more specific application needs. The rules are simple to be defined and they are efficient in providing the final results. The figures obtained demonstrate the efficiency of the approach. Also, by utilizing the proposed detection system together with the wafer map editor one can replace the unnecessary verification of defects in die-bond machines thereby improving the assembly line efficiency.

Chapter 5

On-Line Defect Detection in Web Offset Printing Using Rule-Based Approach

In modern printing machine systems like web offsets, gravure etc, quality and reliability are essential requisites, being the fact that printing houses do not want to waste precious print materials like paper. Determining the web offset parameters at an early stage can potentially allow the monitoring of the production event. The feedback to printing systems about the flaws provides valuable guidance to future considerations in the printing process. Unfortunately, not much research has been conducted on the on-line defect detection of web offset printing using computer vision techniques.

5.1 Need for defect detection in web offset printing

Purll [162] first defined web as being any material produced in the form of strips. Textile, paper, glass, wood, metal, food, and industrial parts on a conveyor belt can all be cited under this heading. Computer tomography images were employed to detect internal defects in hardwood logs [163]. Multi-thresholding, morphological processing, and focus of attention mechanisms were used for the segmentation and recognition of the defects [164]. Texture analysis was also a popular way of looking and segmenting defects on wood board surfaces [165]. Graf et al. [166] used clustering and filtering software to classify sequences of defects on high quality specialty paper. Objects were classified according to their size, shape, contrast, and location. Fourier of transforms and frequency analysis were applied to paper products for the construction of floc contour maps and the retrieval of paper grammage [167]. The algorithm developed here addresses the problem of defect detection and measurement at a location of the web offset rarely studied before (so for by manual means). The computer vision technique, applied for the first time to this particular problem, achieved very satisfying results and yielded a high rate of detection as well as good accuracy of measurements. There are many reasons for poor quality, which arises due to defects while printing. These defects are categorized as follows.

5.2 Printing Visual Inspection Criteria

There are many reasons for poor quality, which arises due to defects while printing. Real time defect detection [69] helps to identify these types of faults at an early stage. This helps the operator to take protective actions, improving print quality and reducing wastage. These defects are categorized as follows.

Ductor streaks: This defect prevails predominantly in rotogravures (web offset printing machines), where, due to particles between the blades which apply print paint to imprint cylinders, and prevents the paint from getting transferred properly. As a result of the same one can see a streaking fault in the direction of paper travel. Also, it is quite possible to have sharp contrast streaks in color jobs of less than 2 colors i.e., the contrast reduces in multicolor prints and that makes the defect detection process very difficult.

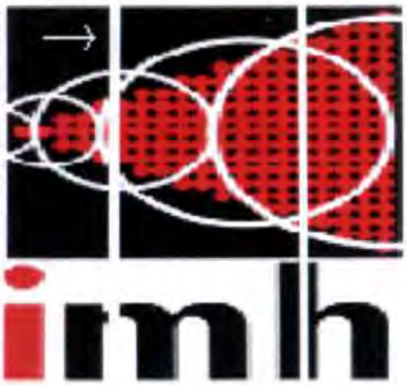


Figure 5.1: Printed image showing ductor streaking faults.

Color splashes: Defect due to splashing of print ink on paper or print media.

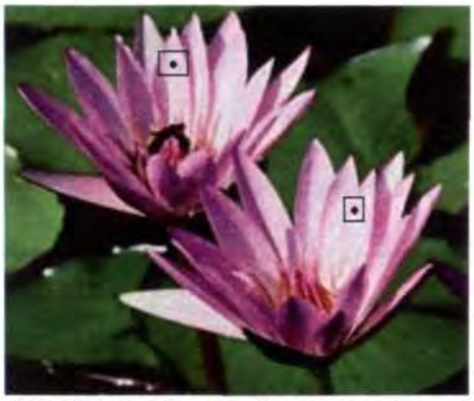


Figure 5.2: Printed image showing color splashes.

Hazing: Defects due to merging of print colors.

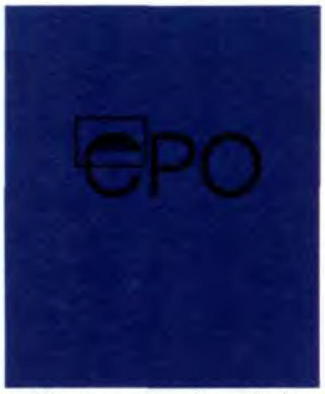


Figure 5.3: Printed image showing defects due to hazing.

Structure defects: Defects in print structure due to ink smudges etc.

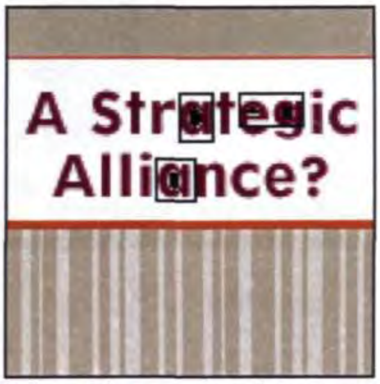


Figure 5.4: Printed image showing structure defects.

5.3 Real-Time Print-Defect Detection System

A real-time print-defect detection system (RTPDS) works using high-resolution progressive scan CCD cameras to capture print images. The images are digitized using a digital frame grabber for analysis in PC. This Chapter explains in detail the principles behind the analysis and identification of faults in print images. The Chapter does not go in detail about the principles of illumination, CCD cameras and frame grabbers etc.

RTPDS requires proper synchronization to the printing process, which is very critical for the system to do on-line detection. Normally the RTPDS employs one CCD camera or multiple cameras based on the required configuration. Most of the times the cameras are fitted on linear traverse to facilitate the movement of cameras in the direction perpendicular to the printing. This allows the cameras to acquire the pictures in x and y directions of the paper.

The input image in Fig. 5.5 and its corresponding edge detected image using x and y direction filters (as explained in section 3.2.1.1) are shown in Figures 5.6 and 5.7 respectively.

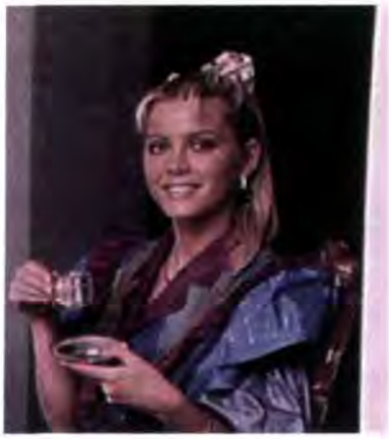


Figure 5.5: Input image.

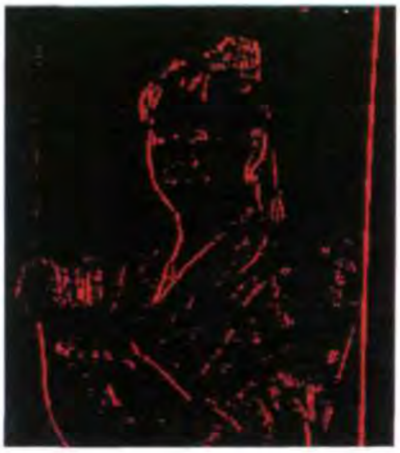


Figure 5.6: Edge detected image using the x direction filter.

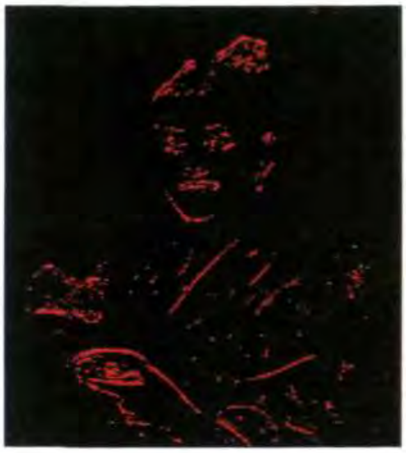


Figure 5.7: Edge detected image using the y direction filter.

A typical print-defect detection system is shown in Fig. 5.8. The system consists of the following components.

CU – Color Units, where the ink gets transferred to print media. The number of color units starts from 1 and goes up to 12 in case of rotogravure.

Unwinding station – The print media (paper, foil, plastic film) get unwound from here before transferred through color station for print.

Rewinding Station – This is the place where the print media get wound once again after printing for further processing.

Illumination Hood – This component controls the illumination required for CCD cameras to acquire pictures.

Control Station – Controls the CCD cameras, illumination, the defect detection process, etc.

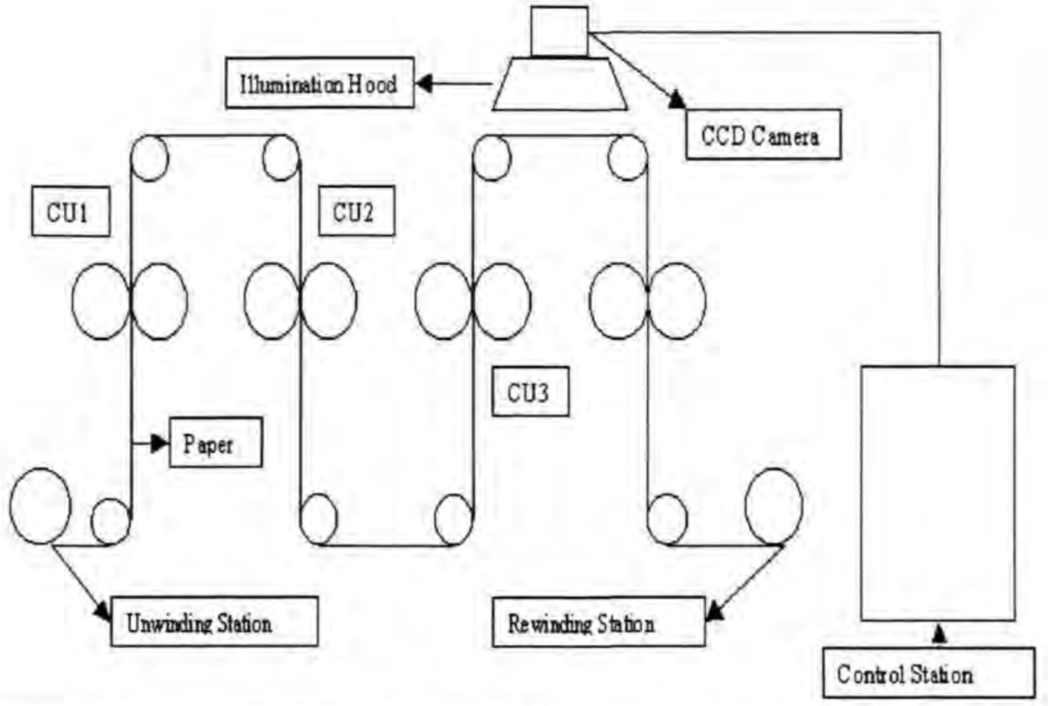


Figure 5.8: Print-defect detection system.

To acquire the pictures in any directions, the camera works in conjunction with a pulse generator mounted on the printing cylinder, which consists of a traverse for the camera movement and a delayed triggering mechanism. The pulse generator generates continuous pulses during printing. For every print job there is a print mark printed to mark the complete revolution of the print cylinder one time. In most of the cases the RTPDS first uses a calibration mode where the number of pulses between two print marks is found. The operator feeds the exact width of the print job into the system. The RTPDS internally remaps the width values as number of steps the cameras to acquire one complete print. Once all the setup procedure has been done, the RTPDS starts the print-defect detection process.

5.4 Defect Detection Process

Real-time print-defect detection system (RTPDS) works on the principle of comparing two images during the identification of faults. One image is called the Golden Master (GM) or Reference Image (RI), while the other image is the actual print image. To have both images compared, they must be in the same position during comparison. The GM is stored in the reference image area of the RTPDS in the initial stages of printing, and an image for comparison is acquired on a regular basis for defect checking. During acquiring of images, there is always shifting in the images. This can be due to many reasons like accuracy in synchronizing camera systems to trigger pulses, sudden jerks, vibration causing the camera to shake etc. This is assumed to happen and RTPDS should take into consideration of all the changes.

The basic defect detection can be separated into the following major steps.

- Golden Master (GM) Image Selection
- Matching of the GM and Print Image
- Comparison of GM to subsequent print images for defect's identification

GM Image selection and matching of the GM and print image are explained in detail in Chapter 3 (section 3.1). The next step is to identify the defects using Rule-based approach. The diagnostic rules for pixel classification and similarity rules for regions merging of the input image are well explained in Chapter 3. In this thesis the only rule that differs from the other applications is the logical rules. The basic defects during printing are classified as streaks, color splashes, structural, hazing as explained in the introduction section. The faults are identified using the following logical rules, and the rules are explained in the subsequent sections.

5.5 Logical Rules

Comparison of GM to subsequent print images for defect's identification

Based on correlation, the GM and the actual live images are compared and the rules are:

Rule 1: If the pixel value at a given position in the actual live image I_{live} is not equal to I_{avg} and if the value is above or below the values of the maximum or minimum deviation of a pixel value namely I_{min} or I_{max}, then the corresponding position is mapped to another test image as 255.

Rule 2: If the pixel value is well within the range of I_{min} or I_{max}, then the corresponding pixel position is marked 0. The above rules 1 and 2 are applied for all pixels in the input error image, resulting in a binary image containing fault region as 255 and non-fault region as 0.

Rule 3: The above rules are followed by a connectivity analysis where the actual image is analyzed and pixels that are connected together are grouped into regions. The outcome of this rule is a list of data containing grouped pixels as regions.

Rule 4: According to rule 4 the algorithm finds the enclosing area for each of the grouped pixels. If the area covered by each grouped pixels is more than 2×2 pixel wide then these regions are marked structural or color splashes. Small regions are marked for evaluation. These positions are ignored as sporadic faults, if the area does not grow in the subsequent defect detection process.

5.5.1 Color Splashes and Structural Faults

Figures 5.9, and 5.10 show the identification of a structural defect from a defect print image using the above rules.

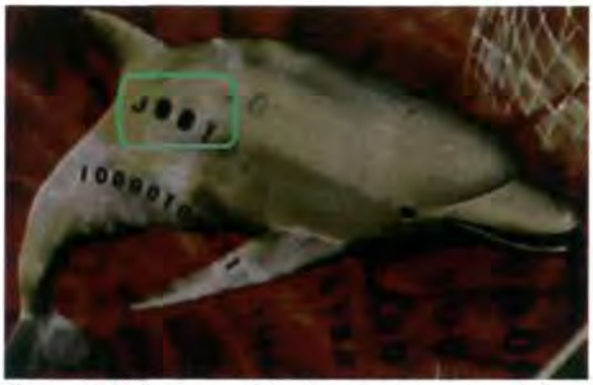


Figure 5.9: Image with detected structural fault.

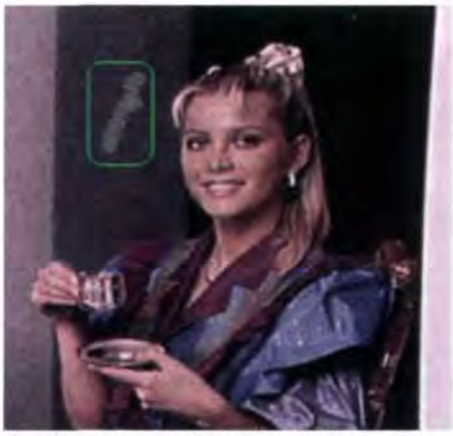


Figure 5.10: Image with detected structural fault.

5.5.2 Ductor Streaks

The identification of the streak faults is the same as the structure faults, except that the fault structures are analyzed, to check whether they represent any lines in the printing directions. If they match the criteria of lines in the printing direction, then the faults are classified as streak faults. Figures 5.11, and 5.12 show the identification of a streak fault using the rules explained in section 5.5.

Identification of contrast errors is much tedious as rigorous calculation is required. For example a light yellow streak on white background is much difficult to detect due to a very small change in the intensity level of the pixels. Additional analysis has to be done for the change in pixel intensity. Otherwise a special filter has to be fitted on the CCD camera to bring in more contrast.

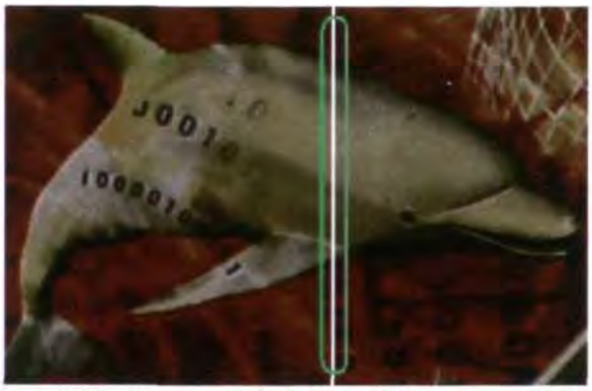


Figure 5.11: Image with detected ductor streak fault.

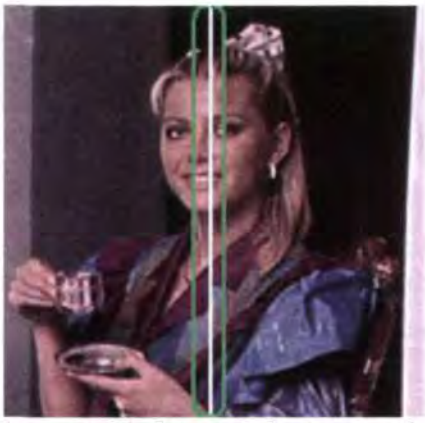


Figure 5.12: Image with detected ductor streak fault.

5.5.3 Hazing

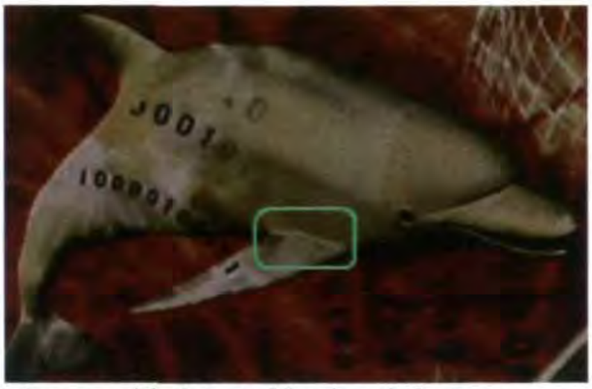


Figure 5.13: Image with detected hazing defect over a period of time.

For the identification of a hazing defect the pixel intensity difference in the actual live image to the GM image is accumulated to a series of defect detection phases. If the difference is not constant and if it shows signs of gradual increase or decrease in the pixel intensity value then those pixels are regrouped and marked as hazing defects. Figures 5.13, and 5.14 show the identification of a hazing defect over a period of time.

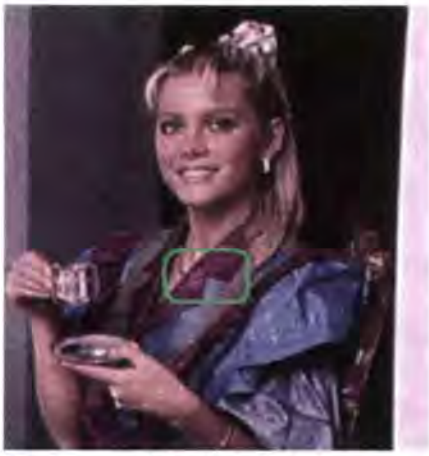


Figure 5.14: Image with detected hazing defect over a period of time.

Table 5.1. The classification rate and the confusion matrix for the RTPDS evaluated with three-layer neural network. The numbers in bold typeface are for the cases when RTPDS was used.

True \ Estimation	Streaks	Hazing	Structural Fault	Color Splashes	Correct %	Error %
Streaks	79 **92**	10 **2**	4 **0**	0 **0**	84.0 **97.8**	16.0 **2.2**
Hazing	8 **1**	85 **92**	2 **2**	0 **0**	89.4 **96.8**	10.6 **3.2**
Structural Fault	0 **0**	2 **1**	58 **69**	13 **3**	79.5 **94.5**	20.5 **5.5**
Color Splashes	0 **0**	1 **1**	4 **5**	81 **80**	94.2 **93**	5.8 **7**
Average rates					86.8 **95.5**	13.2 **4.5**

A collection of defect print images obtained from the same process layer of a product was used for evaluating the proposed system. The set consisted of 94 streaks defect class, 95 hazing defect class, 73 structural defect class, and 86 color splashes defect class images. The classification rates of the proposed RTPDS system was evaluated using three-layer neural network the network has L inputs, N hidden units and O output units and trained to the target error of $E_0 = 0.01$. The results are shown in Table 5.1. By employing RTPDS system to detect print defects, it is found that the non-diagonal elements (errors) in the confusion matrix could be reduced drastically, even when a large portion of printed defects is judged as rejected. The obtained classification rate is considered to be comparable to those of human experts.

5.6 Summary

In this Chapter, a vision system in terms of its hardware modules, as well as the image processing algorithms that it utilizes to scan the color images and locate areas of defect from the printing web using rule-based approach is well explained. The algorithm for this defect detection process is developed in VC++ using multithreading. The reason for multithreading is for the fact that defect detection has to be fast and reliable. The operator has to be informed about the defect at the earliest. The defect detection is implemented to use one of the color planes or all. Most of the time the green channel is used, since this contains most of the structural information. In addition to being tested in a laboratory environment, a prototype of this system was constructed and deployed to a web printing system, where its performance was evaluated under realistic conditions. The system was installed on a Flexo gravure-print-press machine for testing, and it was found that the vision system was able to successfully monitor and detect non-uniformities.

Chapter 6

Segmentation of Biological Images Using Rule-Based Approach

6.1 Introduction

The increase in biological knowledge (especially with respect to the structures of cells, genes and proteins) over the past 20 years, combined with advances in computer technology, has led to the creation of a number of image processing tools to process biological images. These include images that focus on a particular type of information (e.g., cell structures, diseased cells) for all organisms, as well as those that combine various types of information for a single organism. There has therefore been interest recently to extract and segment cell structures from a given biological image in an automated fashion. However, much important information in biological literature is in the form of figures, and little published work has been done on automating the extraction of information from them. It is reported here initial work aimed at accomplishing this task for such figures, microscope images.

Image segmentation finds widespread use in the analysis of biological images, image registration, target location and detection in images and so on. Segmentation also appears to be a key issue in modern biological image analysis enabling numerous clinical applications. With the introduction of digital imaging devices in medicine, computerized tissue recognition and classification have become important in clinical diagnosis systems. One such application is where one may be interested in locating and segmenting anatomical structures in a biological image. Another very important and related application of image segmentation is automated biological image understanding. For example, it plays a vital role in evaluating the shape and size of microorganisms in a microscopic image. Thus, image segmentation plays a vital role in biological image understanding and analysis.

The starting point for this work is previous development of numerical features that can describe complex subcellular patterns in such images and neural network classifiers that are capable of recognizing all major subcellular structures in at least one cell type [168-171]. For this purpose, Subcellular Location Features (SLF) that utilizes geometric moments, texture measures, and morphological image processing were developed and reported [170, 171]. Such classifiers can serve not only as resources for practicing biologists but also as input for systems that can generate new hypotheses using data mining methods. Detection of bilateral symmetry in complex biological images [172] and rule-base segmentation of semiconductor defects have been reported [20, 21].

Image segmentation may be defined as the partitioning of the spatial domain on which the image is defined into regions of interest (ROI). The segmented region may be a complete object or a part of it. All segmentation algorithms make systematic use of image features to segment out the regions of interest. Region growing approach is based on the concept of thresholding and extracting the regions over which a homogeneity criterion is satisfied. However, with straightforward region growing, spectrally similar but spatially disjoint regions are never associated together, thus complicating their identification. Also, it is often not clear at what point the region growing process should be terminated, resulting in under and over-segmentation.

Boundary detection approach is based on the detection of discontinuities in an image commonly known as contour or edge detection. It exploits spatial information through examining local edges found throughout the image. However, conventional techniques for edge detection, e.g., Sobel, Prewitt and Laplacian operators, generally yield a lot of false edges (whose removal requires further processing) and are computationally expensive.

Another promising approach is contour-based image segmentation, which is based on deformable models that has emerged as a vigorously researched model based approach for computer-aided image analysis. One important class of deformable models is the active contours or snakes proposed by Kass et al [173]. The active contour model approach is a supervised technique in the sense that a priori knowledge about the location, size and shape of the ROIs are necessary and the contours require to be initialized accordingly. Image segmentation using deformable models has been used successfully in medical image analysis [174]. Among all the different deformable models proposed, the active contour models have attracted the most attention to date especially in case of biological images. However, the performances of the active contour models highly depend on how close the initial contours are to the actual boundaries of the ROIs. Unfortunately, a priori information about the ROIs is generally not available in biological images. This makes it difficult to employ active contour models in biological image segmentation. Hence, an unsupervised segmentation model is desired so as to guarantee quality segmentation even in the absence of any a priori information about the ROIs. This section presents the application of rule-based approach to segment the defect patterns that appear on the biological images.

6.2 Defect Pattern Segmentation Process

Psycho visual studies have shown that the process of seeing may be comparable to a system where the input information is successively treated and interpreted. Typical example for such an approach is that how a specialist makes decision to segregate defect patterns. The first process of this system is responsible for detecting the visual signal and performing a first data reduction, where the redundant information is simplified, while the information of interest is highlighted. Following this simplification process, perceptual higher-level rules are defined that interpret the visual content. Some rules are hardwired in the human physiology system; others are applied according to the human experience or subjectivity reasons.

In order to obtain a general and modular defect pattern detection technique, the proposed approach adopts a hierarchical perspective. A formal analogy has been drawn between the structure of defect patterns and the symptom of disease in clinical practice. The defect patterns to be recognized are viewed as decision made to a particular disease. Diagnostic rule (Chapter 3) generates screening of disease from symptoms, similarity rule (Chapter 3) generates a test report from screening, and logical rules generate the final decision of the disease from the test report. Thus, a large collection of complex defect patterns can be described by a small number of elementary sub-patterns and clinical rules. They are:

- **Diagnostic rule**

  This rule controls the association of pixels in perceptually uniform regions.

- **Similarity rule**

  The role of the similarity rule is to combine the regions into objects.

- **Logical rules**

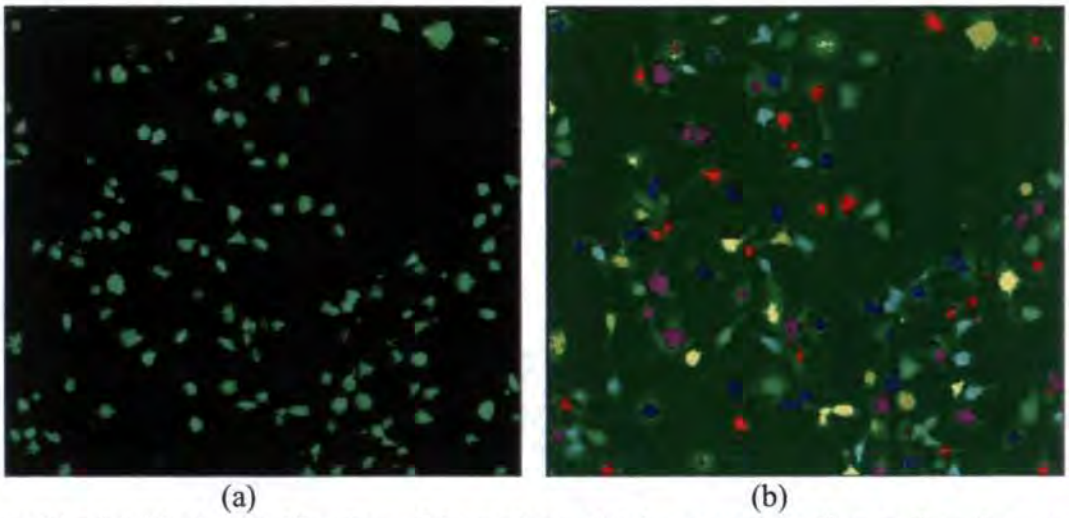
  Locate the (defective) regions based on features.

Defect pattern segmentation is an attempt to perform an analysis of the visual information contained in the error image by describing what is present in the scene and where it is located. This description makes defect pattern segmentation a process very similar to human vision way of segmenting defect patterns. Both try to discover from images what is present in the world, and where it is located. Also it provides information to decide what actions to make to interact or react to the visual input.

In this section it is proposed to a new defect pattern segmentation scheme that has been conceived to achieve a high degree of flexibility. The adopted strategy takes into account some of the known properties of the human visual system (HVS), which remains the optimum image analysis system in several applications. According to this model it is proposed to decompose the general defect pattern segmentation problem in several levels. The lowest levels have a high degree of generality. Their main role is to simplify the input data and to extract the perceptually meaningful information. The proposed process is sufficiently general to be applied on a wide class of applications and input data. The highest levels combine the information generated by the lowest levels according to the specific criteria of the application at hand. The objective of this section is to define the low levels and the high levels rules and the associated algorithmic solutions that provide a perceptually equivalent simplification of the input data.

6.3 Construction of Objects

After pixel classification, uniform regions are obtained using similarity rule, then the objects are built by merging neighboring regions of the similar class. The objects belonging to the black/neutral/white classes can be built or not, depending on the needs. A two-step process explained in section 3.6.3 is used to build objects from the image.



(a) (b)

Figure 6.1: (a) Pixel classification of input bacterial image using diagnostic rules, and (b) Objects built from the bacterial image using similarity rules.

Fig. 6.1 (a) shows the pixel classification of image using diagnostic rules. Fig. 6.1 (b) shows the objects built using the similarity rule. For the sake of computational efficiency, the objects are described as lists of runs. A run is a horizontal segment of pixels of the same class.

It allows working at the object level but also down to the runs level. This allows faster processing in critical cases to compute the object features.

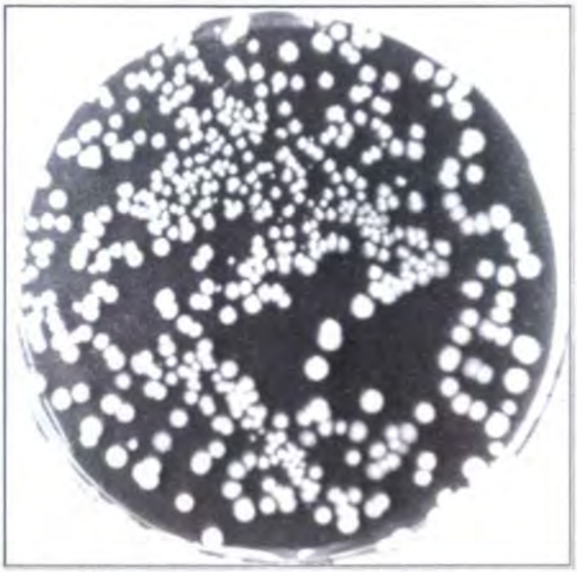
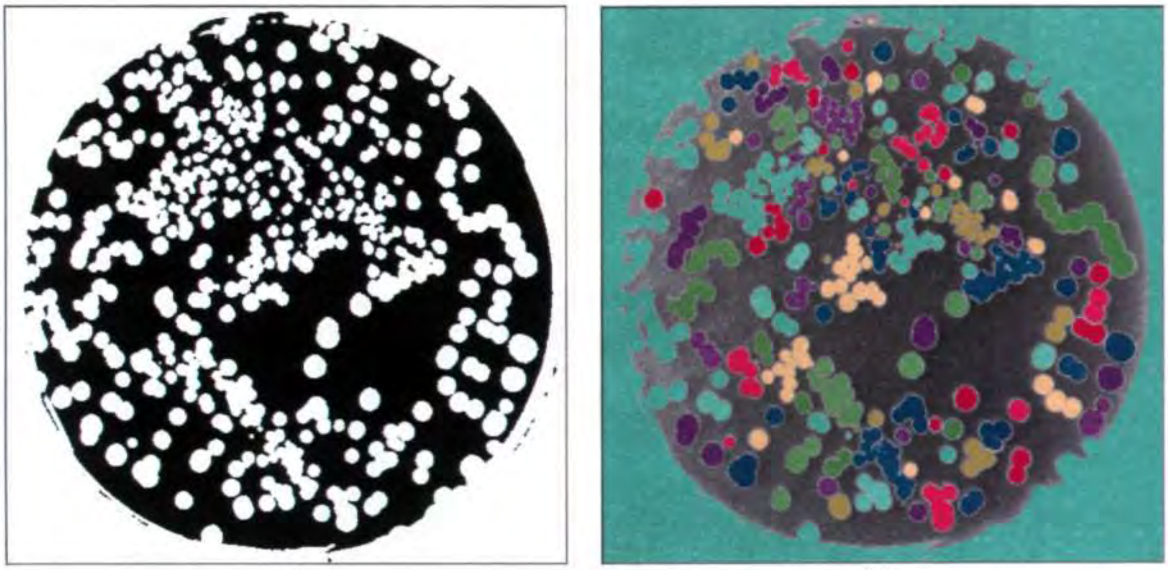


Figure 6.2: Input bacterial image.



(a) (b)

Figure 6.3: (a) Pixel classification of input bacterial image using diagnostic rules, and (b) Objects built from the bacterial image using similarity rules.

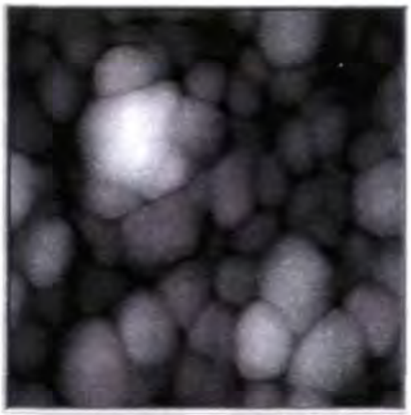
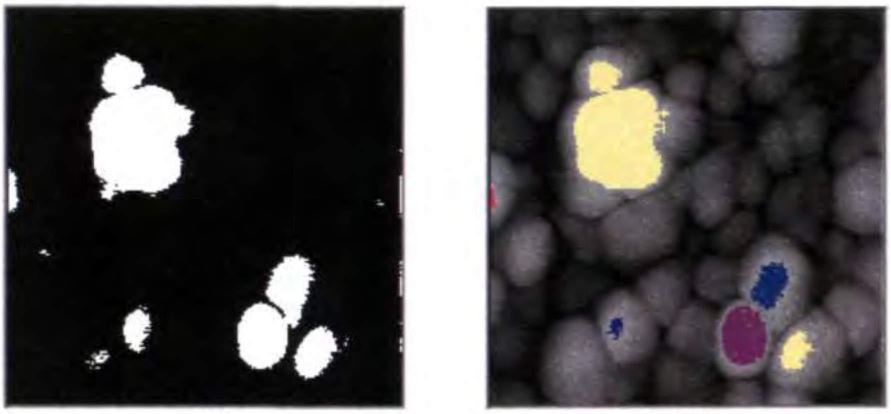


Figure 6.4: Input image containing diseased surface.



(a)

(b)

Figure 6.5: Pixel classification of diseased regions using diagnostic rules, and (b) Objects built for the diseased regions using similarity rules.

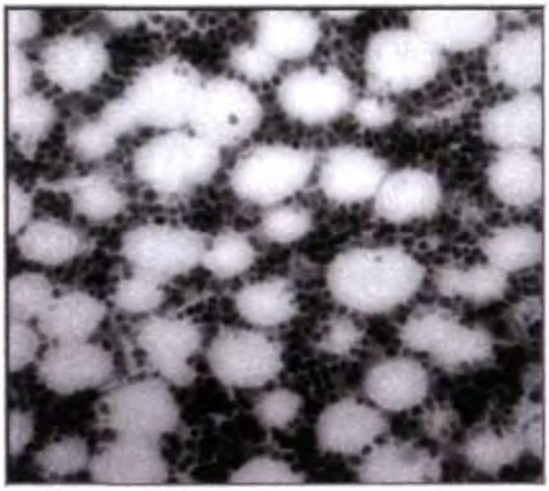


Figure 6.6: Input bone marrow image.

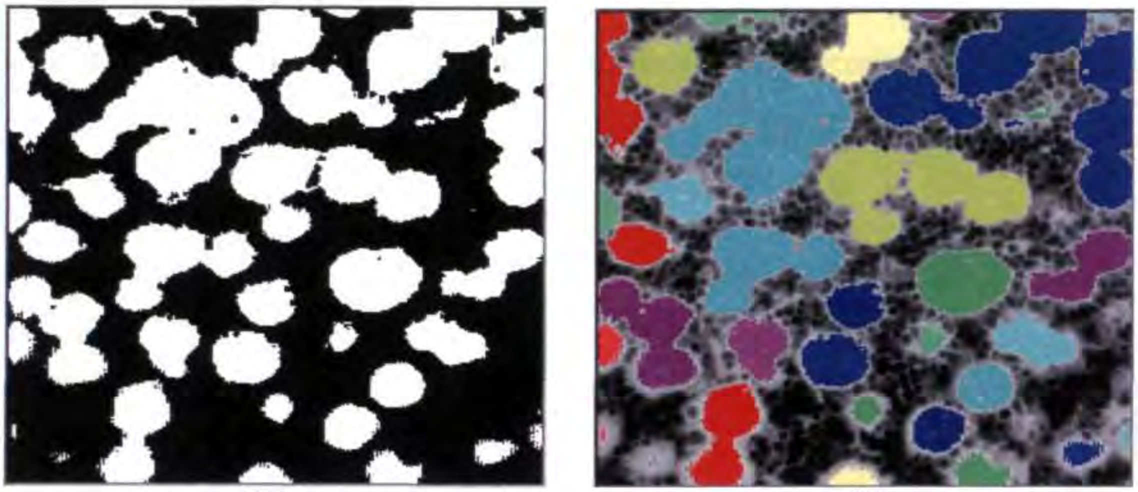


Figure 6.7: (a) Pixel classification using diagnostic rules of bone marrow image, and (b) Objects built using similarity rules for the bone marrow image.

6.4 Logical Rules

'The problem of classification is basically one of partitioning the feature space into regions, one region for each category' [175]. High discriminating features will lead to high and accurate classification rates. Color values (RGB-channels), as local features, are directly related with the images, so they were introduced to the system without any change. When the gray-level image is processed, the pixels are classed as black, neutral or white by using two thresholds, either constant or adaptive (threshold images). When a color image is processed, the pixels are classed as black or white by using two thresholds per color component. The black, neutral and white pixels are given a class number and are grouped to form connected components, or blobs. When the class number is 0, the object is not coded.

After the thresholding and segmentation parameters are set, processing the image generates a list of objects. Each object receives a distinct number starting from 0. The list of objects can be traversed sequentially forwards or backwards. An object can also be reached directly by its number. Sequential access is more efficient. When objects have been built, one can compute a series of geometric features, chosen among object features. Then the value of the computed feature(s) of any object can be retrieved at will. These objects are designated with numbers. A logical rule exists whenever there is a substantial difference in the objects feature of interest. Once an image has been segmented and the corresponding blobs have been built, the features of objects will be computed based on the following geometry.

Area is defined as the number of pixel count (P_i):

$$\text{Area } N = \sum P_i \tag{6.1}$$

Formula for Gravity Center X (N = Area):

$$\overline{X} = \frac{1}{N}\sum X_i \tag{6.2}$$

Formula for Centroid X (P_i= Pixel value):

$$\hat{X} = \frac{\sum P_i X_i}{\sum P_i} \tag{6.3}$$

The object perimeter can be measured indirectly by tracing the object contour by means of contouring methods, and counting the pixels. From the standard geometric features, other ones can be derived. For instance, object elongation is computed as the ratio of large to short ellipse axis or limit height over limit width. Object circularity is defined as the ratio of squared perimeter over four times P_i of the object area. With obtained object features one can count cells that have been labeled with two different dyes (red and green) as shown in Fig. 6.8 (a)

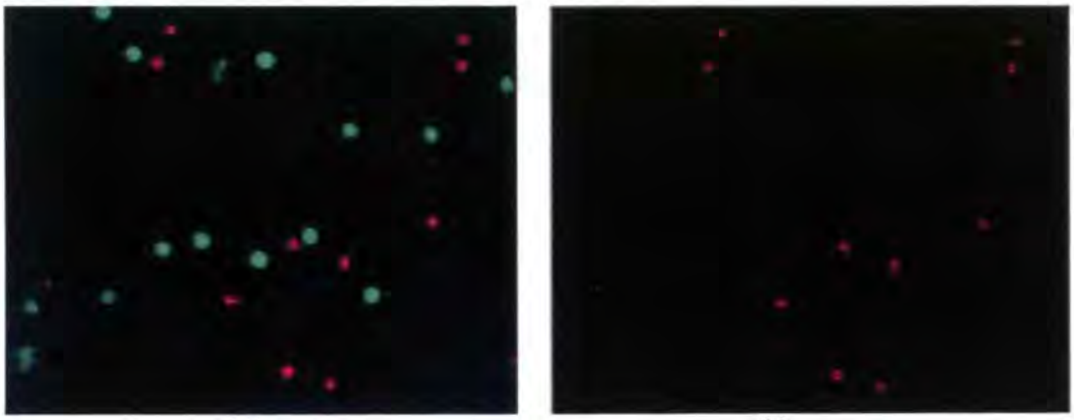

(a)
(b)

Figure 6.8: (a) Input image with live and dead cells, and (b) Objects built for the dead cells.

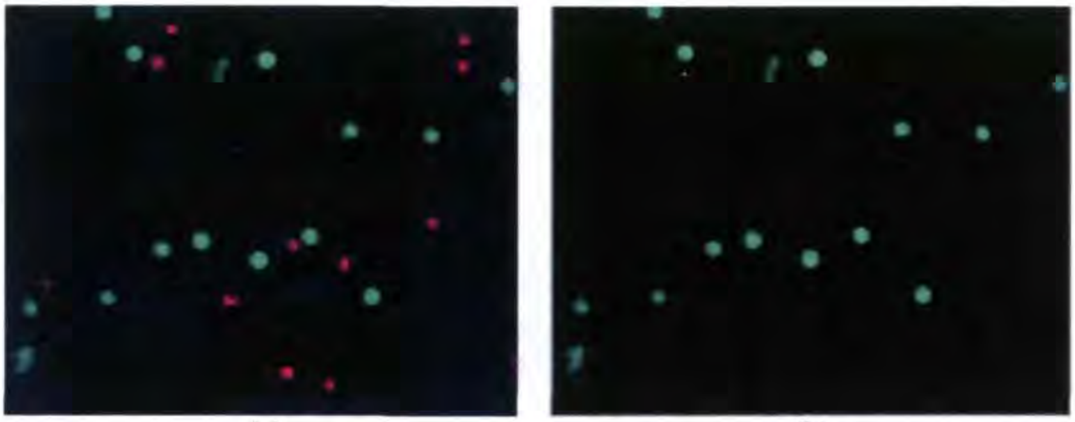
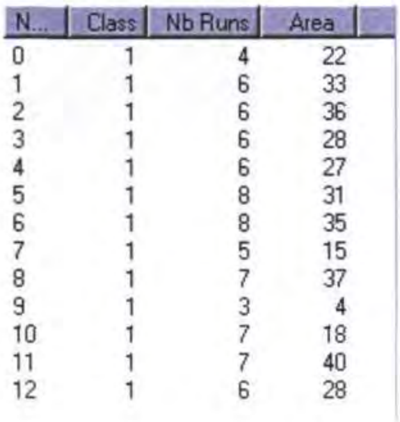

(a) (b)

Figure 6.9: (a) Input image with live and dead cells, and (b) Objects built for the live cells.

It is optimized to distinguish between live (green) and dead (red) cells based on their appearance after staining with a live/dead or bacterial staining kit. Measurement of the two types of cells will then be processed based on detection of discrete cell boundaries, using logical rules of selected area features. Fig. 6.8 (b) shows the objects built for the dead cells and Fig. 6.9 (b) shows the objects built for the live cells.

For the sake of simplicity in the experiment it is consider only the area feature of the dead and live cells of the input image. Table 6.1 summarizes the area features of the dead cells present in the input image. Fig. 6.10 shows the object features built for the dead cells. Similarly, Table 6.2 summarizes the area features of the live cells and Fig. 6.11 shows the object built for the live cells.

Table 6.1: Object features for the dead cells



N...	Class	Nb Runs	Area
0	1	4	22
1	1	6	33
2	1	6	36
3	1	6	28
4	1	6	27
5	1	8	31
6	1	8	35
7	1	5	15
8	1	7	37
9	1	3	4
10	1	7	18
11	1	7	40
12	1	6	28

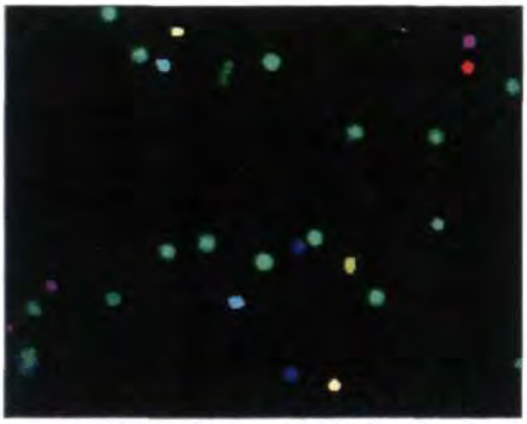
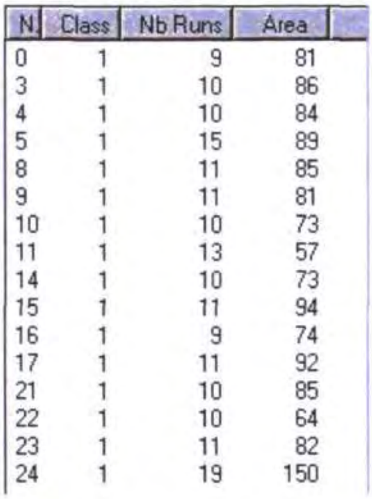


Figure 6.10: Object features built for the dead cells.

Table 6.2: Object features for the live cells



N	Class	Nb Runs	Area
0	1	9	81
3	1	10	86
4	1	10	84
5	1	15	89
8	1	11	85
9	1	11	81
10	1	10	73
11	1	13	57
14	1	10	73
15	1	11	94
16	1	9	74
17	1	11	92
21	1	10	85
22	1	10	64
23	1	11	82
24	1	19	150

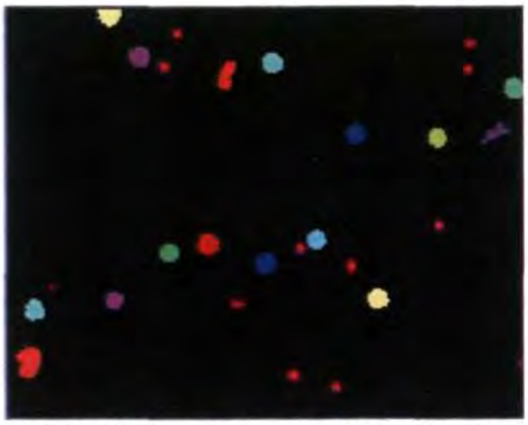


Figure 6.11: Object features built for the live cells.

6.4.1 Objects selection and sorting

Each object can be set in a selected/unselected state. Only the features of selected objects are computed to save processing time. Objects can be selected individually or collectively by a geometric criterion (intersection with a given rectangle) or based on a feature (smaller or larger than a specified threshold). Successive selections interleaved with feature computations are allowed. The selected objects can also be sorted using a feature as comparison criterion. Note that the object list is organized so that the selected objects always precede unselected objects in the sequential traversal.

6.4.2 Selection of the objects

The object segmentation process is "blind" in the sense that it considers any defect pattern as an object. Most of the time, this includes noisy pixels appearing as tiny objects, as well as foreign features. Selection provides an elegant way to get rid of the unwanted blobs and keep only the relevant ones. Selection can be based on the position of objects, i.e. when their bounding box interferes with a given rectangle. In particular, objects touching the image edges can be discarded. Selection can also be based on the value of one feature at a time, i.e. when a given feature value falls in or out of a given range. Selections can be cascaded at will. For the sake of efficiency, object features are computed only on the currently selected objects. This allows computing the features only when strictly required.

6.4.3 Enumeration of the objects

Once selection has led to the elimination of unwanted objects, true object processing can take place. The list of selected objects can be traversed and each object can be queried for the values of the computed features. Optionally, the objects can be sorted on the value of a computed parameter so that the list is traversed in that order. All the objects are kept in a single list. The range of the feature is fixed from 0 to 500 in our experiment. Usually the selected ones appear before the unselected ones, so that traversing from the beginning of the list gives efficient access to all selected objects.

6.5 Summary

In this Chapter an automated method is described to find and interpret the cells of interest in a given biological images using rule-based approach. Our system includes a set of rules to analyze the given input image to generate features of interest. The experimental results proven to give good results statistically for several biological images of interest as explained above and for several species of the input image. These results need to be validated quantitatively, however, in future work and it is also planned to further characterize the sensitivity of the features to different methods and cell types, but also to different specimen preparation methods (e.g., live vs. fixed cells) and different cell states.

Chapter 7

Conclusions and Recommendations

7.1 Conclusions

In this thesis, a new rule-based approach is proposed to segment defect image. Several segmentation techniques already exist but they often focus on the constraints of a specific application and therefore they lack of generality and flexibility. This limits the use of computer vision in all those tasks where the visual data content and the purpose of the defect analysis are not known a priori. Moreover, the limited generality increases the costs for the design of unsupervised image analysis systems.

In order to achieve a higher degree of autonomy, generality and a more flexible solution, the proposed defect image segmentation approach adopts a hierarchical perspective: the visual information in a frame is viewed as being composed of patterns, referred to as objects, which are built from simpler sub-patterns, referred to as regions, which are themselves built from yet simpler primitives corresponding to the pixels in the defect image.

A set of features, which encode the color and texture information by taking into account for the mechanisms of the human visual system, describes the perceptual appearance of the pixels in the image. The selected features constitute the symptom of the visual language. They are used to classify the pixels into perceptually uniform regions according to what havę been defined to as diagnostic rules. The obtained perceptually uniform regions correspond to the screening of the visual language.

A set of descriptors is then associated to each perceptually uniform region. These descriptors characterize the texture properties of the region. Based on this information, a set of general purpose and application dependent similarity rules can be defined in order to merge regions into the final objects of interest. These are interpreted as the test result of the visual language.

Object finding procedures are used to define logic rules that provide more insights on the visual content of the defect image. In particular, logic rules are used to put in correspondence the objects found in successive frames. This process allows us to progressively build objects in the error image. The objects found are interpreted as the logic relationships between test results of a disease.

The increasing number of image segmentation techniques underlines the need for evaluation procedures of the methods and their results. The evaluation of image segmentation techniques is a complex task. In the investigations, it is confronted to the need of evaluating the performances of the proposed image segmentation method. In Chapters 4, 5 and 6 have selected existing analytical and empirical procedures in order to provide objective figures on the quality of the segmentation results.

The proposed rule-based approach to image segmentation has achieved the following properties:

a) Compliant with the human visual system: The human visual system is certainly the best pattern recognizer. Moreover, several computer vision applications aim at providing a description of the scene similar to the one a human being would provide. For these reasons, the proposed image segmentation technique directly takes into account for the mechanisms of human vision and perception. Models of these mechanisms have been selected from the ones proposed in literature and have been used in order to define the texture features.

Each feature and the associated distance measure are defined according to models that adapt the accuracy of the description to the reliability of the available information. The selection of features and descriptors that are compatible with the human visual system achieves in extending the application areas of the proposed technique.

b) Modular: The proposed image segmentation technique has a modular two-phase structure:

First, a fully automatic image simplification phase takes place. This first phase is based on the diagnostic rules and the general-purpose similarity rules. It does not need any parameter adjusting procedure and has demonstrated to be sufficiently general to be successfully applied on several defect images. The result of this phase is a segmentation mask composed of a limited number of perceptually homogeneous regions.

Second, an application dependent phase takes place. The scene is already decomposed in perceptually homogeneous regions. In the second phase, these regions are combined (merged) into objects according to the constraints of the application. In order to perform this merging process, application dependent similarity rules have to be defined. These rules have access to a complete set of descriptors for each region defined in the error image. It is thus easy to set up the correct set of rule that satisfies the need of the specific application. Moreover, the application dependent similarity rules are computationally efficient, since they evaluate a limited number of elements (regions) compared to the total number of pixels in the image. Examples of the simplicity and flexibility of the application dependent similarity rules have been provided in Chapter 3. This modular structure achieves in providing a flexible technique, which easily adapts to different application scenarios.

In order to implement the proposed image segmentation procedure, several state-of-the-art techniques have been selected and eventually improved to better fit the needs of the project. In this conclusive chapter it is outlined the contributions introduced to the following techniques:

Clustering: A multi-feature unsupervised clustering technique is used to group pixels into perceptually uniform regions. An evaluation of existing techniques has suggested the use of spatially unconstrained partitional algorithms. In particular it is found that the algorithm offering the best compromise between complexity and performances. In order to adapt the algorithm, it has proposed improvements in its standard initialization, association and update steps.

Regions refinement: A local refinement procedure using priority rules has been introduced in order to improve the accuracy of the region's borders. The proposed technique only takes into account the texture properties of the pixels in the image.

Object finding: A new feature-based object finding has been proposed. The results of the object finding procedure are used to initialize the analysis of the following input error images and to integrate defect information. This will increase the stability of the description of the error data.

C) Important contributions

The contributions of this work are the following:

1. A new rule-based approach to image segmentation is developed. It is based on the definition of a visual symptom on which a set of diagnostic, similarity and logic rules are successively applied. The proposed approach achieves the final results through a two-step hierarchical strategy. First, a fully automatic and general-purpose simplification procedure takes place, where the redundant information is simplified, while the information of interest is highlighted. Second, an application dependent process is defined on the simplification results. It integrates all the information and assumptions available for the specific application. It uses accurate and complex models to describe the visual information but remains fast and efficient because it is applied on a reduced set of significant regions generated by the simplification step.
2. A discussion of different texture models for segmentation purposes. Accurate texture models are characterized by a high number of parameters that can amplify the curse of dimensionality problem [6]. The comparison in this thesis underlines the limits of

accurate texture models when no a priori hypothesis is available in the choice of significant texture features on which defect features are computed.

3. The proposed initialization technique of the clustering algorithm automatically defines the number of initial classes and their properties according to the distribution of the input data. It evaluates at the same time an arbitrary number of input features and can be successfully extended to other partitional algorithms.
4. The association step of the clustering algorithm has been improved by using distance measures directly related to the properties of the chosen features. The proposed distance measure combines the contribution of several groups of features describing the texture information.
5. The update step of the clustering algorithm has been extended with a new dynamic weighting procedure. This procedure defines the relative weight of the different group of features in the definition of the distance measure. The weighting is fixed according to the reliability of the estimation of each group of features. The proposed solution limits the influence of artificial features such as those generated in estimating texture. Moreover, it reduces the curse of dimensionality problem [4] by associating low weights to less significant features.
6. A set of similarity rules especially the priority rules evaluates the perceptual distance between adjacent regions according to their properties.
7. An object finding procedure solves the correspondence problem and controls the quality of the segmentation results for successive frames of the input error image.
8. Evaluation procedures to analytically and empirically measure the quality of the segmentation results.

9. The identification and segmentation of defects by automated means is tried out for the first time in the semiconductor industry after wafer dicing using a rule-based approach.
10. A new rule-based approach is developed to recognize a wide range of defects.
11. A new rule-based approach is investigated to categorize the defect patterns to its specific structure i.e., chip out, bridging defect, metal lifting, and scratches, etc., so far done only by manual means. The rules can be easily modified without the need of expert designers.
12. A new wafermap is developed for the first time with the in-house defects using wafer map editor provides a final solution to in-line inspection.

7.2 Recommendations

The proposed rule-based approach to image segmentation has a modular structure. Thus, it is particularly convenient to extend its capabilities and performances. Each level that composes the proposed rule-based approach can be independently extended and improved. The following outline few ideas for each level that will consider useful in improving the proposed system.

The Symptom: The proposed visual symptom exploits the current knowledge of the mechanisms of human vision in defining those features that better approximate our perception of the visual information. This knowledge is subject to changes and in the future new and more accurate models of the human visual systems will be available. In particular better models of color, and texture perception could improve the overall performances of the system by providing more perceptually uniform descriptors. These descriptors could simply replace the ones defined in the current implementation without the need for re-engineering the entire system.

The Diagnostic Level: One of the most interesting extensions to this work would be to exploit edges information. In particular, the principles related to edges and their perceptions have not been taken into account. These can be taken into account in the definition of visual features. They could also help in a more accurate definition of the spatial supports on which to compute homogeneous texture features.

The Similarity Level: Among the different levels of the proposed clinical language, the similarity level is certainly the one that is the easiest to extend. At this level, the image has already been simplified into a small set of perceptually homogeneous regions. Several application dependent similarity rules can be defined in addition to the ones proposed in this thesis so as to adapt the method to the constraints of new applications. When a set of application dependent similarity rules is available, an interesting extension would be the automatic selection of the most appropriate set of rules. The proposed clinical approach would then be able to tackle different scenarios and applications.

The Logical Level: In the currently proposed method, the logical level is useful in obtaining a stable analysis of the scene. However, it is not introduced any artificial intelligence analysis of the error image. A deeper understanding of the meaning of the objects detected in the error image can be extremely useful to avoid misclassification. Artificial intelligence techniques should be integrated at the logical level in order to interpret the visual content of the error image.

In addition to the possible extensions of the proposed rule-based approach, further efforts on the evaluation of the image segmentation performances are required. Finally, the proposed method should be applied on a larger set of applications and video sequences. Rule-based approach is currently going through a major evolution. Hence, there are many research issues that need to be addressed. Several research issues are given below.

1. The proposed technique can be converted as a user-friendly programming environment. For example, an integrated development environment can aid medical personnel to train the system to detect newer pattern of the same disease under investigation, which is not available in the manufacturer list. Similar is the case for textile, chemical and biotechnology industries to identify newer pattern of the same kind without user and manufacturer coordination.
2. Due to aging most of the inspection systems malfunction by considering wrong reference patterns and identifying wrong patterns thereby giving wrong information to the user. Adopting the proper self-tuning technique will avoid user, supplier coordination to tune the system.
3. Further integration of this method will show insight in implementing 3-D object recognition at a fast rate.
4. Implementation of this technique into various stages of IC manufacturing (dicing, bonding, etc) and printing industries with suitable reference database (training signals) will lead to 100% yield and thereby saving operational cost of fabrication.

References

1. S. Theodoridis and K. Koutroumbas, Pattern recognition, Academic Press Incorporated, New York, 1998.

2. A. Jain, R. Duin, and J. Mao, "Statistical Pattern Recognition: A Review," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 22, pp. 4-37, Jan. 2000.

3. D. Marr, Vision, San Francisco, W. H. Freeman and Co., 1982.

4. R. Watt, Understanding Vision. Academic Press, New York. 1991.

5. H. G. Barrow and J. M. Tenenbaum, "Computational vision," Proceedings of the IEEE, vol. 69, pp. 572-595, May 1981.

6. H Miyashita et al., "Particle Measurements in Vacuum Tools by In-Situ Particle Monitor," *Journal of Vacuum Science and Technology* A 17, no. 3, pp.1066-1070, 1999.

7. L Milor, G Hill, and Y Peng, "Layer Yield Estimation Based on Critical Area and Electrical Defect Monitor Data," in *Proc. International Symposium on Semiconductor Manufacturing*, pp. 99-102, 1999.

8. "2002 Update, International Technology Roadmap for Semiconductors," Semiconductor Industry Association, http://public.itrs.net/, viewed on 11/09/2004.

9. The National Technology Roadmap for Semiconductors, Semiconductor Industry Association, pp. 163-171, Washington, DC, 1997.

10. Zhong, Z. W., and K. S. Goh, "Flip Chip on FR-4, Ceramics and Flex," Journal of Electronics Manufacturing, Vol. 10, No. 2, 2000, pp. 89-96.

11. Zhong, Z. W., and K. S. Goh, "Analysis and Experiments of Ball Deformation for Ultra-Fine-Pitch Wire Bonding," Journal of Electronics Manufacturing, Vol. 10, No. 4, 2000, pp. 211-217.

12. K.W. Tobin et al., "Using Historical Wafermap Data for Automated Yield Analysis," *Journal of Vacuum Science and Technology* A 17, no. 4 (1999): 1369–1376.

13. T. P. Karnowski, S.S. Gleason and K.W. Tobin, "Fuzzy Logic Connectivity in Semiconductor Defect Clustering," SME Technical Paper MS00-240, Society of Manufacturing Engineers, Dearborn, MI, 2000.

14. X. L. Dai, M. A. Hunt, and M. A. Schulze, "Automated image registration in the semiconductor industry: A case study in the direct to digital holography inspection system," *Machine Vision Applications in Industrial Inspection XI, Proc. SPI*E, vol. 5011, Santa Clara, California, 23-24, January 2003.

15. Shaun S. Gleason, Regina K. Ferrell, Thomas P. Karnowski and Kenneth W. Tobin, "Detection of semiconductor defects using a novel fractal encoding algorithm," Design, Process Integration and Diagnostics in IC Manufacturing, Proc. SPIE Vol. 4692, pp.61-71, 2002.

16. Karnowski, T.P., Tobin, K.W., Arrowood, L.F., Ferrell, R.K., Goddard, J.S., and Lakhani, F., "Field-test results of an image retrieval system for semiconductor yield learning", SPIE's 13th International Symposium on Electronic Imaging: Science and Technology, San Jose Convention Center, January 2001.

17. P. Xie, S. Guan, "A Golden-Template Self-Generating Method for Patterned Wafer Inspection," Machine Vision and Applications, vol. 12, no. 3, pp. 149-156, 2000.

18. Ganesha Udupa, B. K. A. Ngoi, H. C. Freddy Goh and M. N. Yusoff, "Defect detection in unpolished Si wafers by digital shearography," Measurement Science and Technology, Vol.15, pp.35-43, 2004.

19. K. Kameyama, Y. Kosugi, T. Okahashi and M. Izumita, "Automatic Defect Classification in Visual Inspection of Semiconductors Using Neural Networks," IEICE Trans. Inf. & Syst., vol. E81-D, no.11, pp.1261-1271, November (1998).

20. N.G. Shankar, and Z.W. Zhong, Rule-based Inspection of Wafer Surface, The Fourth International Conference on Control and Automation (ICCA'03), pp. 752-755, Montreal, 2003.

21. N.G. Shankar, and Z.W. Zhong, Rule-Based Classification of Defects on Semiconductor Wafers, ICMAT, Symposium L, Singapore, December 2003.

22. P.B. Chou, A.R. Rao, M.C. Sturzenbecker, F.Y. Wu, V.H. Brecher, "Automatic Defect Classification for Semiconductor Manufacturing", Machine Vision and Applications, Vol. 9 (4), pp. 201-214, 1997.

23. T.S. Newman, and A.K. Jain, "A Survey of Automated Visual Inspection", Computer Vision and Image Understanding, Vol.61, No.2, March 1995, pp. 231-262.

24. Don BE, Brecher V (1995) Recent advances in the automatic inspection of integrated circuits for pattern defects. Mach Vision Appl 8: 5-19.

25. Chin RT (1988) Survey Automated Visual Inspection: 1981 to 1987. Computer Vision Graphics Image Process 41: 346-381.

26. Chen CH, Cheng TH, Wu WT, Driscoll S (1998) Machine Vision Algorithms for Semiconductor Wafer Inspection: A Project with Inspex. Proc of SPIE 3521: 221-228

27. Babian F (1986) Optical defect detection limits in semiconductor wafers and msks. Ph.D. thesis, Stanford University, Stanford, California.

28. Newman TS, AK (1995) A survey of automated visual inspection. Comput Vision Graphics Image Process 61(2): 231-262.

29. Moganti M, Ercal F, Dagli CH, Tsunekawa S (1996) Automatic PCB inspection algorithms: a survey. Comput Vision Image Understanding 63(2): 287-313

30. Klaus Wiltschi, Axel Pinz, Tony Lindeberg, An automatic assessment scheme for steel quality inspection, Machine Vision and Applications (2000) 12: 113-128.

31. Dom BE, Brecher VH, Bonner R, Batcelder JS, Jaffe RS (1988) The P300: a system for automatic patterned wafer inspection. Mach Vision Appl 1(3): 205-221.

32. Khalaj BH, Aghajan HK, Kailath T, Digital image processing techniques for patterned wafer inspection. Proceedings of the SPIE - The International Society for Optical Engineering, vol.1926, pp.508-516, 1993.

33. A. Paulraj, R. Roy, and T. Kailath. Estimation of signal parameters via rotational invariance techniques - ESPRIT. In *Proc. 19th Asilomar Conf. on Circuits, Systems and Comp.*, pp. 83-89, Asilomar, Pacific Grove, CA, November 1985.

34. P. Xie and S.U. Guan, "A golden- template self-generating method for patterned wafer inspection", Machine Vision and Applications (2000) 12: 149-156, May 2000.

35. Lilly Spirkovska and Max B. Reid, Coarse-Coded Higher-Order Neural Networks for PSRI Object Recognition, IEEE Trans. on Neural Networks, Vol. 4(2), pp. 276-283, March 1993.

36. Ji-Chien Lee, Bing J. Sheu, Wai-Chi Fang, and Rama Chellappa, " VLSI Neuroprocessors for Video Motion Detection", IEEE Transactions on Neural Networks, Vol. 4, no. 2, pp. 178-191, March 1993.

37. H.Kobayashi, J.L.White and A.A.Abidi, An Active Resistor Network for Gaussian filtering of Images, IEEE Solid-State Circuits, Vol. 26, no. 5, pp. 738-748, May 1991.

38. A. Moore, J. Allman and R.M. Goodman, A Real-Time Neural System for Color Constancy, IEEE Trans. On Neural Networks, vol. 2, no. 2, pp. 237-247, March 1991.

39. W.C. Lin, F.Y. Liao, C.K. Tsao and T. Ligutla, A Hierarchical Multiple – View Approach to Three – Dimensional Object Recognition, IEEE Trans. On Neural Networks, vol. 2, no. 1, pp. 84-92, 1991.

40. Bernard Widrow, Rodney G. Winter and Robert A. Baxter, Layered Neural Nets for Pattern Recognition, Speech, Signal Processing, Vol. 36, no.7, pp. 1109-1118, 1988.

41. H. Singh, F. Lakhani, P. Proctor, and A. Kazakoff, "Defect-Data Management System at SEMATECH", *Solid State Technology*, pp. 75-80, December 1995. Penwell Publishing.

42. M.M. Slama, M.H. Bennett, P.W. Fletcher, "Advanced In-line Process Control of Defects", Integrated Circuit Metrology, Inspection, and Process Control VI, SPIE Vol. 1673, pp. 496-505, June 1992.

43. Gleason, S. S., Hunt, M. A., Sari-Sarraf, H., "Semiconductor Yield Improvement Through Automatic Defect Classification," CRADA Final Report for CRADA Number ORNL92-0140, September 1995.

44. B. Trafas, M.H. Bennett, M. Godwin, "Meeting Advanced Pattern Inspection System Requirements for 0.25-mm Technology and Beyond", *Microelectronic Manufacturing Yield Reliability, and Failure Analysis*, SPIE Vol. 2635, pp. 50-55, September 1995.

45. R. Sherman, E. Tirosh, Z. Smilansky, "Automatic Defect Classification System for Semiconductor Wafers", Machine Vision Applications in Industrial Inspection, SPIE Vol. 1907, pp. 72-79, May 1993.

46. K. W. Tobin, S. S. Gleason, F. Lakhani, M. H. Bennett, "Automated Analysis for Rapid Defect Sourcing and Yield Learning," Future Fab International, issue 4, vol. 1, pp. 313-320, 1998.

47. Tobin, K.W., Gleason, S.S., Karnowski, T.P., Cohen, S.L., "Feature Analysis and Classification of Manufacturing Signatures on Semiconductor Wafers", Machine Vision Applications in Industrial Inspection V, Proc. SPIE, vol. 3029, pp. 12-25, April 1997.

48. Tobin, K.W., Gleason, S.S., Karnowski, T.P., Cohen, S. L., and Lakhani, F., "Automatic Classification of Spatial Signatures on Semiconductor Wafermaps",

Metrology, Inspection, and Process Control for Microlithography XI, Proc. SPIE, vol. 3050, pp. 434-444, July, 1997.

49. Tobin, K.W., Gleason, S.S., Karnowski, T.P., Bennett, M.H., "An Image Paradigm for Semiconductor Defect Data Reduction", Metrology, Inspection, and Process Control for Microlithography X, Proc. SPIE Vol. 2725, pp. 194-205, May 1996.

50. S.S. Gleason, K.W. Tobin, T.P. Karnowski, "Spatial Signature Analysis of Semiconductor Defects for Manufacturing Problem Diagnosis", Solid State Technology, PennWell Corp., Tusla OK, July 1996.

51. Gleason SS, Tobin KW, and Karnowski TP, "Rapid Yield Learning through Optical Defect and Electrical Test Analysis," presented at Metrology, Inspection, and Process Control for Microlithography XII session at SPIE's 23rd International Symposium on Microlithography, pp. 57-63, Santa Clara, CA, February 1998.

52. J.T. Tou, R.C. Gonzalez, Pattern Recognition Principles, Addison-Wesley Publishing Company, Reading Massachusetts, 1974.

53. R.O. Duda, P.E. Hart, Pattern Classification and Scene Analysis, John Wiley & Sons, New York, New York, 1973.

54. H. Bunke, A. Sanfeliu, Syntactic and Structural Pattern Recognition, Theory and Applications, World Scientific, New Jersey, 1990.

55. K.W. Tobin, S.S. Gleason, and T.P. Karnowski, "Adaptation of the Fuzzy K-Nearest Neighbor Classifier for Manufacturing Automation", Machine Vision Applications in Industrial Inspection, Proc. SPIE, Vol. 3306, pp. 122-130, 1998.

56. S. Haykin, Neural Networks: A Comprehensive Foundation, Macmillan College Publishing Company, New York, 1994.

57. V.N. Gudivada and V.V. Raghavan, "Content-Based Image Retrieval Systems", *IEEE Computer Magazine*, 0018-9162, pp. 18-23, September 1995.

58. V. Sankaran, C.M. Weber, K.W. Tobin "Inspection in Semiconductor Manufacturing", Webster's Encyclopedia of Electrical and Electronic Engineering, Vol. 10, pp.242-262, Wiley & Sons, NY, 1999.

59. CCD Image Sensors Databook, DALSA Corporation, Ontario, Canada, March 1996.

60. Limin Fu, Neural Network's In Computer Intelligence, McGraw Hill, New York, 1994.

61. Nalini K. Ratha, Computer Vision Algorithms on Reconfigurable Logic Arrays, Ph.D thesis, Michigan State University, Department of Computer Science, 1996.

62. K.G. Shin and P. Ramanathan. Real-time computing: a new discipline of computer science and engineering. Proc IEEE, 82, pp. 6-24, January 1994.

63. C.M. Brown and D. Terzopoulos, editors. Real-time computer vision. Cambridge University Press, Cambridge, 1994.

64. O. Firs Hein, Defense applications of image understanding. IEEE Expert, vol. 10, issue. 5, pp. 11-17, October 1995.

65. J. Onuki, T. Maenosono, M. Shibata, N. Lijima, H.Mitsui, and Y. Yoshida, ANN accelerator by parallel processor based on DSP. In Proc. Intl. Joint Conference on Neural Networks: 1913-1913, Nagoya, Japan, October 1993.

66. C. C. Weems, Architectural requirements of image understanding with respect to parallel processing. Proc IEEE, 79(4), pp. 537-547, April 1991.

67. C. C. Weems, S. P. Levitan, A.R. Hanson, E. M. Riseman, D. B. Shu, and J.G. Nash. The image understanding architecture, International Journal of Computer Vision, 2(3): 251-282, 1989.

68. A. N. Choudhary, J. H. Patel, and N. Ahuja. NETRA: A hierarchical and partitionable architecture for computer vision systems. IEEE Trans. on Parallel and Distributed Systems, 4(10): 1092-1104, October 1993.

69. G. D. Micheli. Computer-aided Hardware-Software codesign. IEEE Micro: 10-16, August 1994.

70. K.W. Tobin, S.S. Gleason, F. Lakhani, and M.H. Bennett, "Automated Analysis for Rapid Defect Sourcing and Yield Learning", Future Fab International, vol. 1, Issue 4, Technology Publishing Ltd., pp.313-320, (1997).

71. K.W. Tobin, T. P. Karnowski, R. K. Ferrell, "Image Retrieval in the Industrial Environment ", SPIE's 11th Annual Symposium on Electronic Imaging Science & Technology, San Jose, CA, January 23-29, 1999.

72. Vision Inspector, TOPCON Vi – 1200, 2200 Series Manual, Version: 1999 – 12M, Tokyo, Japan.

73. A. Rangarajan, R. Chellapa, and Y.T. Zhou, A model-based approach for filtering and edge detection in noisy images, IEEE Trans. On Circuits and Systems, 37, pp. 140-144, 1990.

74. N.G. Shankar, N. Ravi, and Z.W. Zhong, On-Line Defect Detection in Web Offset Printing, The Fourth International Conference on Control and Automation (ICCA'03), pp. 794-798, Montreal, 2003.

75. M. Anderberg, Cluster Analysis for Applications. Academic Press, 1973.

76. A. Jain and D. A. Zongker, "Feature selection: Evaluation, application, and small sample performance," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 19, no. 2, pp. 153-158, 1997.

77. R. Setiono and H. Liu, "Neural network feature selector," IEEE Transactions on Neural Networks, vol. 8, no. 3, pp. 654-662, 1997.

78. P. Zamperoni, "Analysis of some region growing operators for image segmentation," in Proceedings of the International Conference of Advances in Image Processing and Pattern Recognition, vol. 2, (Amsterdam, Netherlands), pp. 204-208, Jan. 1986.

79. M. Kocher and R. Leonardi, "Adaptive region growing technique using polynomial functions for image approximation," Signal Processing, vol. 11, no. 1, pp. 47-60, 1986.

80. L. C. Yian and L. Xiaobo, "Adaptive image region-growing," IEEE Transactions on Image Processing, vol. 3, pp. 868-872, Nov. 1994.

81. R. Adams and L. Bischof, "Seeded region growing," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 16, pp. 641-647, June 1994.

82. X. Jianhua, T. Adali, and W. Yue, "Segmentation of magnetic resonance brain image: Integrating region growing and edge detection," in Proc. IEEE International Conference on Image Processing, vol. 3, pp. 544-547, Sept. 1995.

83. A. Baraldi and F. Parmiggiani, "Single linkage region growing algorithms based on the vector degree of match," IEEE Transactions on Geoscience and Remote Sensing, vol. 34, pp. 137-148, Jan. 1996.

84. C. Revol and M. Jourlin, "A new minimum variance region growing algorithm for image segmentation," Pattern Recognition Letters, vol. 18, pp. 249-258, Mar. 1997.

85. A. Tremeau and N. Borel, "A region growing and merging algorithm to color segmentation," Pattern Recognition, vol. 30, pp. 1191-1204, July 1997.

86. J. P. Thiran, V. Warscotte, and B. Macq, "A queue-based region growing algorithm for accurate segmentation of multi-dimensional digital images," Signal Processing, vol. 60, pp. 1-10, July 1997.

87. A. Mehnert and P. Jackway, "An improved seeded region growing algorithm," Pattern Recognition Letters, vol. 18, pp. 1065-1071, Oct. 1997.

88. S. A. Hojjatoleslami and J. Kittler, "Region growing: a new approach," IEEE Transactions on Image Processing, vol. 7, pp. 1079-1084, July 1998.

89. J. Maeda, T. Iizawa, T. Ishizaka, C. Ishikawa, and Y. Suzuki, "Segmentation of natural images using anisotropic diffusion and linking of boundary edges," Pattern Recognition, vol. 31, pp. 1993-1999, Dec. 1998.

90. P. C. Chen and T. Pavlidis, "Segmentation by texture using a co-occurrence matrix and a split-and-merge algorithm," Computer Graphics and Image Processing, vol. 10, pp. 172-182, June 1979.

91. U. Ranft, J. M. S. Prewitt, and K. S. Fu, "Segmentation of microscopic transverse section pictures of muscle tissue using a split-and-merge technique," in Proc. of the Sixth International Conference on Pattern Recognition, vol. 2, (New York, USA), pp. 626-628, June 1982.

92. R. H. Laprade, "Split-and-merge segmentation of aerial photographs," Computer Vision, Graphics, and Image Processing, vol. 44, pp. 77-86, Oct. 1988.

93. T. T. Yih and H. T. Wen, "Model-guided attributed string matching by split-and-merge for shape recognition," International Journal of Pattern Recognition and Artificial Intelligence, vol. 3, pp. 159-179, June 1989.

94. S. Kasif and A. Rosenfeld, "Pyramid linking is a special case of ISODATA," IEEE Transactions on Systems, Man and Cybernetics, vol. 13, pp. 84-85, Jan. 1983.

95. A. Postolache and J. Trecat, "Pyramidal approach to license plate segmentation," Journal of Electronic Imaging, vol. 5, pp. 402-409, July 1996.

96. V. Cantoni, L. Lombardi, G. Manzini, and L. Cinque, "Page segmentation using a pyramidal architecture," in Proc. Fourth IEEE International Workshop on Computer Architecture for Machine Perception. CAMP'97, vol. 23, (Los Alamitos, CA, USA), pp. 195-199, Mar. 1997.

97. F. Ziliani and B. Jensen, "Unsupervised segmentation using modified pyramidal linking approach," Proc. IEEE International Conference on Image Processing, October 1998.

98. T. Pun, "A new method for grey-level picture thresholding using the entropy of the histogram," Signal Processing, vol. 2, pp. 223-237, July 1980.

99. J. N. Kapur, P. K. Sahoo, and A. K. C. Wong, "A new method for gray-level picture thresholding using the entropy of the histogram," Computer Vision, Graphics, and Image Processing, vol. 29, pp. 273-285, Mar. 1985.

100. M. T. Du and H. C. Ying, "A fast histogram-clustering approach for multi-level thresholding," Pattern Recognition Letters, vol. 13, pp. 245-252, Apr. 1992.

101. F. Albregtsen, "Non-parametric histogram thresholding methods: Error versus relative object area," in Proceedings of the 8th Scandinavian Conference on Image Analysis, vol. 1, (Tromso, Norway), pp. 273-280, 1993.

102. C. H. Li and P. K.S.Tam, "Robustness analysis of histogram thresholding algorithms," in Proceedings of the Second International Conference on Mechatronics and Machine Vision in Practice, vol. 1, (City Univ. Hong Kong, Hong Kong), pp. 203-208, 1995.

103. L. Liyuan, G. Jian, and C. Winan, "Gray-level image thresholding based on Fisher linear projection of two-dimensional histogram," Pattern Recognition, vol. 30, pp. 743-749, May 1997.

104. T. H. Do and M. Installe, "A fast clustering procedure based on ISODATA algorithm with application to remote sensing," in Proceedings of the 4th International Joint Conference on Pattern Recognition., vol. 1, (Kyoto, Univ, Kyoto, Japan), pp. 326-328, July 1978.

105. V. F. R. Dias, "Thresholding using the ISODATA clustering algorithm," IEEE Transactions on Systems, Man and Cybernetics, vol. 10, pp. 771-774, Nov. 1980.

106. T. Y. Phillips and A. Rosenfeld, "An ISODATA algorithm for straight line fitting," Pattern Recognition Letters, vol. 7, pp. 291-297, June 1988.

107. C. S. Carman and M. B. Merickel, "Supervising ISODATA with an information theoretic stopping rule," Mar. 1990.

108. N. B. Venkateswarlu and P. S. V.S. K. Raju, "Fast ISODATA clustering algorithms," Mar. 1992.

109. K. Takahashi, H. Nakatani, and K. Abe, "Color image segmentation using ISODATA clustering method," in Proceedings of the Second Asian Conference on Computer Vision ACCV '95., vol. 1, (Nanyang Technol. Univ, Singapore), pp. 284-290, July 1995.

110. T. V. Ravi and K. C. Gowda, "An ISODATA clustering procedure for symbolic objects using a distributed genetic algorithm," July 1999.

111. E. Forgy, "Cluster analysis of multivariate data: efficiency versus interpretability of classifications," Biometrics, vol.21, p. 768, 1965.

112. S. Z. Sleim and M. A. Ismail, "On the local optimality of the fuzzy isodata clustering algorithm," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 8, pp. 284-288, Mar. 1986.

113. M. J. Sabin, "Convergence and consistency of Fuzzy C-means/ISODATA algorithms," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 9, pp. 661-668, Sept. 1987.

114. A. K. Jain, Fundamentals of Digital Image Processing. Information and System Sciences, Prentice Hall, 1989.

115. W. K. Pratt, O. D. Faugeras, and A. Gagalowicz, "Visual discrimination of stochastic texture fields," IEEE Transactions on Systems, Man and Cybernetics, vol. 8, pp. 796-804, Nov. 1978.

116. S. Theodoridis and K. Koutroumbas, Pattern recognition. Academic Press Incorporated, Nov. 1998.

117. A. K. Jain and R. C. Dubes, Algorithms for Clustering Data. Prentice Hall, 1988.

118. CIE, "Colorimetry." CIE Publ. No. 15.2. 1986.

119. A. A. Michelson, Study in Optics. University of Chicago, 1927.

120. Soile. P., Morphological Image Analysis, Springer-Verlag, Berlin, 1998.

121. Pattern Classification and Scene Analysis, Duda R.O., Hart P. E., Wiley & Sons, Canada, 1973.

122. Haralick, R.M., Sternberg, S.R., and Zhuang, X., "Image analysis using mathematical morphology," *IEEE Trans. Pattern Analysis and Machine Intelligence,* PAMI-9, 532-550 (1987).

123. J. Bullier, "Architecture du systeme visuel," in Proc. of the NSI'97, La perception visuelle, Aspects Multi-Disciplinaires, (Aussois, France), May 1997.

124. K.W. Tobin, F. Lakhani and T.P. Karnowski, "An Industry Survey of Automatic Defect Classification Technologies, Methods and Performance," Design, Process Integration and Diagnostics in IC Manufacturing, Proc. SPIE Vol. 4692, pp.46-53, 2002.

125. P. Brodatz, Texture: A Photograph Album for Artists and Designers. Dover, 1956.

126. G. Cross and A. Jain, "Markov Random Field Texture Models," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 5, pp. 25-39, 1983.

127. A. Jain and K. Karu, "Learning Texture Discrimination Masks," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 18, pp. 195-205, 1996.

128. J. K. Hawkins, Picture Processing and Psychopictorics, ch. Textural Properties for Pattern Recognition, pp. 347-370. New York: Academic Press, 1970.

129. A. Bovik, M. Clarke, and W. Geisler, "Multi-channel Texture Analysis Using Localized Spatial Filters," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 12, pp. 55-73, 1990.

130. B. Chaudhuri, N. Sarkar, and P. Kundu, "Improved Fractal Geometry Based Texture Segmentation Technique," IEE Proceedings, vol. 140, pp. 233-241, 1993.

131. O. Faugeras and W. Pratt, "Decorrelation Methods of Texture Feature Extraction," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 2, pp. 323-332, 1980.

132. M. Pietikainen, T. Ojala, and Z. Xu, "Rotation-invariant texture classification using feature distributions," Pattern Recognition, vol. 33, pp. 43-52, Jan. 2000.

133. T. Randen and J. H. Husoy, "Filtering for texture classification: A comparative study," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 21, pp. 291-310, Apr. 1999.

134. W. K. Pratt, Digital image processing, John Wiley Interscience, 1991.

135. R. M. Haralick, K. Shanmugam, and I. Dinstein, "Textural features for image classification," IEEE Transactions on Systems, Man and Cybernetics, vol. SMC 3, no.6, vol. 66, pp. 610-621, Nov 1973.

136. S. Aksoy and R. M. Haralick, "Textural features for image database retrieval," IEEE Workshop on Content Based Access of Image and Video Libraries, pp. 45-49, 1998.

137. J. M. H. du Buf and P. Heitkamper, "Texture features based on Gabor phase," Signal Processing, vol. 23, pp. 227-244, 1991.

138. A. K. Jain and F. Farrokhnia, "Unsupervised texture segmentation using gabor filters," Pattern Recognition, vol. 24, no. 12, pp. 1167-1185, 1991.

139. S. Fischer and J. Bigün, "Texture boundary tracking with gabor phase," in Proc. of the 9th Scandinavian Conference on Image Analysis (G. Borgefors, ed.), vol. 2, (Uppsala, Sweden), pp. 877-884, June 6-9 1995.

140. N. W. Campbell and B. T. Thomas, "Automatic selection of gabor filters for pixel classification," in 6th Int. Conf. on Image Processing and its Applications, vol. 2, (Dublin, Ireland), pp. 761-765, July 1997.

141. S. G. Mallat, "A theory for multi-resolution signal decomposition: The wavelet representation," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 11, pp. 674-693, July 1989.

142. S. Fukuda and H. Hirosawa, "A wavelet-based texture feature set applied to classification of multi-frequency polarimetric SAR images," IEEE Transactions on Geoscience and Remote Sensing, vol. 37, pp. 2282-2286, Sept. 1999.

143. A. Laine and J. Fan, "Texture classification by wavelet packet signatures," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 15, pp. 1186-1190, Nov. 1993.

144. T. Chang and C. C. J. Kuo, "Texture analysis and classification with tree-structured wavelet transform," IEEE Transactions on Image Processing, vol. 2, no. 4, pp. 429-441, 1993.

145. I. Ng, T. Tan, and J. Kittler, "On local linear transform and Gabor filter representation of texture," in Proc. of the International Conference on Pattern recognition, pp. 627-631, 1992.

146. T. Randen and J. H. Husoy, "Multichannel filtering for image texture segmentation," Optical Engineering, vol. 33, pp. 2617-2625, Aug. 1994.

147. O. Pichler, A. Teuner, and B. J. Hosticka, "A comparison of texture feature extraction using adaptive Gabor filtering, pyramidal and tree structured wavelet transforms," Pattern Recognition, vol. 29, pp. 773 - 742, May 1996.

148. Meyer Y, "Wavelets Algorithms and Applications", Philadelphia: SIAM, 1993.

149. Kolaczyk E, "WVD Solution of Inverse Problems", Doctoral Dissertation, Stanford University, Dept. of Statistics, 1994.

150. Porter R, Canagarajah N, "A Robust Automatic Clustering Scheme foe Image Segmentation Using Wavelets", IEEE Trans. on Image Processing, 5, pp.662-665, 1996.

151. Unser M, "Texture Classification and Segmentation Using Wavelet Frames", IEEE Trans. Image Processing, 4, pp.1549-1560, 1995.

152. Ryan T. W, Sanders D, Fisher H.D, Iverson A.E, "Image Compression by Texture Modeling in the Wavelet Domain", IEEE Trans. Image Processing, 5, pp. 26-36, 1996.

153. Antonini M, Barlaud M, Mathieu P, Daubechies I, "Image Coding Using Wavelet Transform," IEEE Trans. Image Processing, 1, pp.205-220, 1992.

154. Donoho D.L, Interpolating Wavelet Transforms, Technical Report 408, Dept. of Statistics, Stanford University, 1992.

155. R. Castagno, Video Segmentation based on multiple features for interactive and automatic multimedia applications. PhD thesis, Swiss Federal Institute of Technology, Lausanne, CH, 1998.

156. J. Serra, Image analysis and mathematical morphology. Ecole Supérieure de Mines de Paris: Academic Press, 1982.

157. R. Haralick and L. Shapiro, Computer and Robot Vision. New York: Addison Welsey, 1992. Chapter 10.

158. IEEE Broadcast Technology Society, IEEE Std 208-1995, IEEE Standard on Video Techniques: Measurement of Resolution of Camera Systems, 1993 Techniques, IEEE, (1995).

159. Image Processing Toolbox, MATLAB, Version 3.2, Release 13, June 28, (2002).

160. International Standards Organization. ISO 12233: 2000, Photography – Electronic Still-picture Cameras – Resolution Measurements, ISO, (2000).

161. Draper, N.R and H. Smith, Applied Regression Analysis, 3rd Ed., John Wiley & Sons, New York, 1998.

162. D. J. Purll, "Solid State Image Sensors in Automated Visual Inspection", B.G. Batchelor, D.A. Hill and D.C. Hodgson, Eds. Elsevier Science Publishing Company, Inc., New York, NY, pp. 255-293, 1985.

163. D. Zhu and A.A. Louis Beex, "Robust Spatial Autoregressive Modeling for Hardwood Texture Analysis", Journal of Visual Communication and Image Representation, vol. 5, no.1, March, pp. 41-51, 1994.

164. P. Araman, D. Schmoldt, T.H. Cho, D. Zhu, R. Conners, and D. Earl Kline, "Machine Vision Systems for Processing Hardwood Lumber and Logs", AI Applications, 6, no. 2, April 1992, pp. 13-26.

165. C. W. Kim and A.J. Koivo, "Hierarchical Classification of Surfaced Defects on Dusty Wooden Boards", Proc. of 10th. Int'l. Conf. On Pattern Recognition, pp. 775-779, Atlantic City, New Jersey, 1990.

166. J. E. Graf, S.T. Enright, and S.I. Shapiro, "Automated Web Inspection Ensures Highest Quality Nonwovens", Tappi Journal, 78, no. 9, September 1995, pp. 135-138.

167. J. P. Bernie and W.J. Murray Douglas, "Local Grammage Distribution and Formation of Paper by Light Transmission Image Analysis", Tappi Journal, 79, no. 1, January 1996, pp. 193-202.

168. M. V. Boland, M. K. Markey, and R. F. Murphy, "Classification of Protein Localization Patterns Obtained via Fluorescence Light Microscopy," presented at the 19th Annu. Intl. Conf. IEEE Eng. Med. Biol. Soc., Chicago, IL, USA, 1997.

169. M. V. Boland, M. K. Markey, and R. F. Murphy, "Automated recognition of patterns characteristic of subcellular structures in fluorescence microscopy images," *Cytometry*, vol. 33, pp. 366-375, 1998.

170. R. F. Murphy, M. V. Boland, and M. Velliste, "Towards a Systematics for Protein Subcellular Location: Quantitative Description of Protein Localization Patterns and Automated Analysis of Fluorescence Microscope Images," *ISM*B, vol. 8, pp. 251-259, 2000.

171. M. V. Boland and R. F. Murphy, "A Neural Network Classifier Capable of Recognizing the Patterns of all Major Subcellular Structures in Fluorescence Microscope Images of HeLa Cells," *Bioinformatic*s, vol. 17, in press, 2001.

172. Detection of bilateral symmetry in complex biological images has been reported, Perception, 2000, volume 29, pp. 31- 42.

173. M. Kass, A. Witkin, and D. Terzopoulos, "Snakes: Active Contour Models," Intl. J. Computer Vision, 1(4), 321-331, 1987.

174. D. Terzopoulos, and T. McInerney, "Deformable Models in Medical Image Analysis: A Survey," Medical Image Analysis, 1(2), 91-108, 1996.

175. Duda R.O., Hart P. E., Pattern Classification and Scene Analysis, Wiley & Sons, Canada, 1973.