A Multiple Sensors Approach To Wood Defect Detection

By

Xiangyu Xiao

Dissertation submitted to the Faculty of Bradley Department of Electrical Engineering of Virginia Polytechnic Institute and State University in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

in

Electrical Engineering

Richard W. Conners, Chairman

D. Earl Kline, Co-Chairman

Yilu Liu

Lynn Abbott

Anbo Wang

February 16, 1998

Blacksburg, Virginia

Keywords: Machine Vision System, Computer Vision, Multi-sensor, Data Fusion, Fuzzy Logic

A Multiple Sensors Approach To Wood Defect Detection

By

Xiangyu Xiao

Committee Chairman: Richard W. Conners

Electrical Engineering

Abstract

In the forest products manufacturing industry, recent price increases in the cost of high-quality lumber together with the reduced availability of this resource have forced manufacturers to utilize lower grade hardwood lumber in their manufacturing operations. This use of low quality lumber means that the labor involved in converting this lumber to usable parts is also increased because it takes more time to remove the additional defects that occur in the lower grade material. Simultaneously, labor costs have gone up and availability of skilled workers capable of getting a high yield of usable parts has markedly decreased. To face this increasingly complex and competitive environment, the industry has a critical need for efficient and cost-effective new processing equipment that can replace human operators who locate and identify defects that need to be removed in lumber and then remove these defects when cutting the lumber into rough parts. This human inspection process is laborious, inconsistent and subjective in nature due to the demands of making decisions very rapidly in a noisy and tiring environment. Hence, an automatic sawing system that could remove defects in lumber while creating maximum yield, offers significant opportunities for increasing profits of this industry. The difficult part in designing an automatic sawing system is creating an automatic inspection system that can detect critical features in wood that affect the quality of the rough parts. Many automatic inspection systems have been proposed and studied for the inspection of wood or wood products. But, most of these systems utilize a single sensing modality, e.g., a single optical sensor or an X-ray imaging system. These systems cannot detect all critical defects in wood.

This research work reported in this dissertation is the first aimed at creating a vision system utilizes three imaging modalities: a color imaging system, a laser range profiling system and an X-ray imaging system. The objective of in designing this vision system is to detect and identify: 1) surface features such as knots, splits, stains; 2) geometry features such as wane, thin board; and 3) internal features such as voids, knots. The laser range profiling system is used to locate and identify geometry features. The X-ray imaging system is primarily used to detect features such as knots, splits and interior voids. The color imaging system is mainly employed to identify surface features.

In this vision system a number of methodologies are used to improve processing speed and identification accuracy. The images from different sensing modalities are analyzed in a special order to offset the larger amount of image data that comes from the multiple sensors and that must be analyzed. The analysis of laser image is performed first. It is used to find defects that have insufficient thickness. These defects are then removed from consideration in the subsequent analysis of the X-ray image. Removing these defects from consideration in the analysis of the X-ray image not only improves the accuracy of detecting and identifying defects but also reduces the amount of time needed to analyze the X-ray image. Similarly, defect areas such as knot and mineral streak that are found in the analysis of the X-ray image are removed from consideration in the analysis of the color image. A fuzzy logic algorithm -- the approaching degree method-- is used to assign defect labels. The fuzzy logic approach is used to mimic human behavior in identifying defects in hardwood lumber.

The initial results obtained from this vision system demonstrate the feasibility of locating and identifying all the major defects that occur in hardwood lumber. This was even true during the initial hardware development phase when only images of unsatisfactory quality from a limited lumber of samples were available. The vision system is capable of locating and identifying defects at the production speed of two linear feet per second that is typical in most hardwood secondary manufacturing plants. This vision system software was designed to run on a relative slow computer (200 MHz Pentium processor) with aid of special image processing hardware, i.e., the MORRPH board that was also designed at Virginia Tech.

Acknowledgements

My sincerest appreciation goes to my advisor, Professor Richard W. Conners. I owe him a great deal for his guidance, his patience, his financial support throughout the course of my studies. I am especially appreciative of the effort he spent on editing my dissertation. Without his help, I would never have completed this degree. I would also like to express my thanks to Professor D. Earl Kline for his advice, his help and his encouragement during the course my research. And I would also like to thank Professor Yilu Liu, Professor Lynn Abbott, Professor Anbo Wang, and Professor John Bay for their suggestions and the time they spent as members of my committee.

The hardware support provided by students at the Spatial Data Analysis Laboratory is greatly appreciated. And special thanks goes to Mr. Thomas H. Dryer, Mr. Paul LaCase, Mr. William King, Mr. Frank Qiu, and Mr. Pannus Arvanitis for their help and support.

The financial support provided by Mr. Philp A. Araman of the U.S. Forest Service, Mr. John Zinn of the International Woodworking Fair, and Aristokraft, Inc. is also very much appreciated. And I would like to thank Mr. William Fortney for his providing me lumber samples and for his useful detailed descriptions of lumber defects.

I wish to thank my children Miaomiao and Donna for their love and understanding. And finally, I appreciate my wife, Junying Tang's support and patience throughout the period of this study.

Table of Contents

List of Notations

Images

$C_{RAW}(i, j)$	Raw color image obtained directly from the color camera
$L_{RAW}(i, j)$	Raw laser image directly from laser scanning system
$X_{RAW}(i, j)$	Raw X-ray image directly from x-ray imaging system
$C_U(i, j)$	Unregistered or original color image
$L_U(i, j)$	Unregistered or original laser image
$X_U(i, j)$	Unregistered or original X-ray image
$BW_U(i, j)$	Unregistered black/white extracted from image $C_U(i, j)$
$C(i, j)$	Registered color image
$BW(i, j)$	Registered black/white extracted from image $C(i, j)$
$L(i, j)$	Registered laser image
$X(i, j)$	Registered X-ray image
$BW_T(i, j)$	Thresholded black/white image
$L_T(i, j)$	Thresholded laser image
$X_T(i, j)$	Thresholded X-ray image
$BW_C(i, j)$	Connected-component labeled black/white image
$L_C(i, j)$	Connected-component labeled laser image
$X_C(i, j)$	Connected-component labeled X-ray image

Other notations

j	Coordinate for cross-board direction, with positive direction to the left

i	Coordinate for down-board direction, with positive direction downward
j_{UL}	j direction of Upper-left corner of the minimum inscribing rectangle (MIR) of a component
i_{UL}	i direction of Upper-left corner of the MIR
j_{LR}	j direction of lower-right corner of the MIR
i_{LR}	i direction of lower-right corner of the MIR
T_{TE}	Threshold for thick enough pixels in laser images
T_{TU}	Threshold for *unknown thickness* in laser images
T_{touch_edge}	Threshold to determine weather a component is on the edge.
D_i	Distance between MIR's of two components in i direction
D_j	Distance between MIR's of two components in j direction
$H_{xt}(k)$	Transmission level histogram of the whole X-ray image
$H_{xd}(k)$	Transmission level histogram of the defect regions (within MIRs of the defects) in the X-ray image
$H_{bwb}(k)$	Histogram of the whole board face in the black and white image
$H_{bwd}(k)$	Histogram of the defect regions in the black and white image

List of Tables

List of Figures

Chapter 1

INTRODUCTION

1.1 Motivation

1.1.1 Hardwood Secondary Manufacturing

In the forest products industry, trees are converted into wood products through several processing operations. The first operation, called logging, includes cutting trees and turning them into logs. A second processing operation converts logs into products such as lumber and plywood that can be used as raw material by others. This operation is known as primary processing or primary manufacturing. A third processing operation involves converting the products from the primary manufacturers into finished products such as doors, windows, cabinets, and furniture. This processing operation is called secondary processing or secondary manufacturing. Of the above three processing operations, secondary manufacturing is the operation that adds the most value to the product. Therefore, it is important that it be done efficiently and correctly.

In a secondary manufacturing plant, lumber is cut up into rough parts in a location called the rough mill. In a rough mill, sawyers perform the cutting of lumber into parts based on a set of given part specifications. The primary objective of the sawyers is to remove defects from the lumber while maximizing the volume of the parts produced. The defects are features in wood that will reduce the quality and value of the final products. Sawyers typically make hundreds of sawing decisions during a work day, each of which can and does affect the quality and the yield of the product. The location, type, and size of defects, together with the size of a rough part, largely

determine its grade and value. As one might see, the rough mill is the critical part of converting existing lumber into high-quality parts.

In a rough mill, a sawyer locates and identifies the wood features on lumber that are considered to be defects. The lumber is then cut into parts to eliminate those defects while maximizing the yield of parts obtainable from the lumber. Recognizing the features that need to be removed and formulating an optimal cutting strategy have proven to be a difficult job [CON97, FOR95]. The job is difficult for a number of reasons. Given plenty of time, people can do an excellent job at wood feature location and identification. However, in the manufacturing environment decisions need to be made very quickly. Even highly skilled sawyers have a difficult time working hour after hour at peak efficiency due to the dull and repetitive nature of the work [HUB92, MOO95]. Also when specifications for what constitutes a removable feature, i.e., defect, are changed, people have difficulty making the change rapidly. Not all surface features represent undesirable defects. Also, a feature is considered a defect depending on the product being manufactured and the quality of that product. Finally, secondary manufacturers are finding it more and more difficult to get skilled sawyers capable of making good cutting decisions. If one must start with an unskilled worker, management must invest up to two years training and experience to get a skilled sawyer.

In secondary manufacturing, hardwood lumber has been used to make many high value and high demand products such as doors, windows, cabinets, furniture, flooring, and other household fixtures. For many years, secondary manufacturers have used the higher grades of hardwood lumber, with little consideration being given to the utilization and processing of it. However, recent price increases in the hardwood lumber market, together with the reduced availability of high-grade timber resources have forced manufacturers to make their products from lower-grade hardwood lumber. The cost of the labor to convert lower-grade lumber into rough parts is greater due to the fact that: 1) lower-grade lumber has more defects and it is harder to locate clear areas and make cutting decisions that result in maximized yield recovery; and 2) there is a tight labor market for the skilled people who can do the conversion.

These increases in raw material cost and labor cost are making it difficult for manufacturers to remain profitable. To survive in the increasingly complex and competitive environment, the industry has to find new technologies that maximize the utilization of available lumber resources. Studies show that approximately 40 to 75 percent of the cost of producing defect-free parts in a furniture plant is the cost of lumber [KLI96]. Therefore, any increase in the volume of usable parts produced from a given volume of material input will markedly affect profit margins. Hence, the industry has a critical need for efficient and cost-effective new processing methods that convert existing lumber into high-quality parts.

Researchers from government, universities, and industry have recognized the need for automating the process of converting lumber to rough parts in the forest products industry for a number of years. Clearly, an automatic sawing system that can remove defects in lumber while creating maximum yield would provide the solution to the above mentioned industrial problem. Automatic saws, in fact, could offer other advantages such as the saving of labor cost, and the reduction in demand for highly skilled operators. The major difficulty in creating such an automatic sawing system is developing a machine vision system that can detect critical features in wood that can affect the value of the final product. The challenge to creating such a machine vision system comes from the extremely variable and heterogeneous nature of hardwood lumber and the image-processing speed required to inspect the lumber at production rates.

1.1.2 Hardwood Lumber

Some typical features in hardwood include holes, knots, mineral streak, splits, checks, wane, bird peck, decay, internal voids, and surface stains. These features (defects) are categorized into defect classes defined in Table 1.1. The features a manufacturer does not want in the finished product in order to meet a particular quality standard are subsequently referred to as defects or removable defects. To reduce manufacturing cost, most manufacturers will allow some features to appear in the final product. The type, size, and location of features can have a marked impact on the quality and cost of the finished product.

Hardwood is a very heterogeneous material, with both surface and interior features. Most of these features occur in random locations with variable shape, color, and size. Some interior features may become visible on the surface during the manufacturing process. Besides natural defects, manufacturing defects can be introduced during part handling, conditioning, and milling. State-of-the-art commercial systems still rely on operators to inspect lumber for such critical defects. To automatically detect these features reliably, accurately, and in real-time has proven to be a difficult task for the designers of machine vision systems.

Table 1.1 Hardwood lumber features that are often treated as defects

Features (defect) class	Description
Hole	Voids in the wood caused by worms, missing knots
Knot	Portion of a branch that is inside the trunk of a tree. Grain is generally perpendicular to the grain of the trunk.
Split/Check	A lengthwise separation of the wood
Wane	An area where the outer surface of the log is included in the board
Bird peck	A small hole or patch of distorted grain resulting from birds pecking through the growing cells of the tree
Decay	The disintegration of wood due to the action of wood destroying fungi
Internal void	A cellular separation that occurs in the interior of a piece of wood
Surface stains	A surface discoloration
Mineral streak	An olive to greenish-black or brown discoloration of undetermined cause

1.1.3 Machine Vision Systems for Hardwood Lumber Defect Detection

A significant amount of research conducted both at universities and in industry has gone into developing a machine vision technology for automatic defect detection. Hence a number of machine vision systems have been proposed by researchers for the inspection of wood or wood products [CON87, CHA89, CHO91, CON92, CON95, AST92, HAG93, REG92, KLI93, CON97]. While different sensing methods are used on these systems, most of these systems utilize a single-sensing modality. Generally, the sensing modality is either one or more black and white cameras or color cameras. Substantial progress has been made in developing both the hardware and the software of a system for processing black/white and color imagery in order to locate and identify surface features, and some internal features [CON95]. Since these systems are based on limited sensing methods, none of these developments have resulted in a machine vision system that can meet the secondary manufacturing industry's expectations.

The need for using a multiple-sensor approach for locating and identifying features in wood has been cited by several researchers [CON92, AST92, HAG93, KLI93]. Manufacturers not only want to accurately detect surface defects, they also want to detect shape, internal defects and other features that affect the quality of the parts needed in their operation. Several sensing techniques including optical, ultrasonic, laser-range profiling, and X-ray imaging have been studied. Of all the sensing modalities that have been studied, the following three sensing modalities seem the most appropriate for use in a multiple-sensor system [CON92, KLI93, CON93, CON95]:

1. Color camera--used to detect surface features;
2. Laser range profiling--used to identify shape features;
3. X-ray scanner--used to locate internal features.

However, little work has been done on designing an overall machine vision system that effectively combines these sensing modalities.

For many years machine vision systems have been applied to a number of industrial inspection problems that might be applicable to the lumber inspection problem. A number of successful systems have been created for special problems. These include vision systems for the inspection of printed circuit boards [HAR88, DAR88], thin-film disk heads [SAN88], and machine parts inspection [BER91]. Other machine vision systems with X-ray, ultrasonic, and microwave sensing have been applied to the inspection of bread [BAT93], solder-joints [BAR88], integrated circuits [PAU83, CHO93], semiconductor wafers [JAI89a] and so on. Extensive surveys of machine vision systems for industrial inspection are cited in the literature [CHI88, BAY93]. Although some of the above processing and analysis technologies can be used in designing an automatic inspection system for hardwood lumber, these vision systems cannot be directly applied to hardwood lumber. The inappropriateness of existing technologies is due, in part, to the enormous variability in the nature and the condition of the wood materials [CON92, KLI96], and in the inflexibility of current machine vision technology [CHI86, CHO91, NG93]. In summary, there is a critical need for more fundamental research directed specifically in the area of machine vision for wood inspection.

1.1.4. Multiple Sensor Machine Vision Prototype for Automatic Lumber Inspection

To address the critical need for more research in automated lumber systems, special hardware to collect color image, X-ray image, and laser-range data in real-time has been developed in the Spatial Data Analysis Laboratory at the Virginia Tech [HOU93, DRA95]. A prototype multiple-sensor vision system for wood inspection has been created for collecting images with three imaging modalities: color imaging, X-ray imaging, and laser-range profiling. The hardware architecture of the system is illustrated in Figure 1.1. Also, a computer vision system for analyzing black and white images has been created [CHO91, GUO92].

To further extend the capability of the Virginia Tech's multiple-sensor machine vision system and fully integrate all the sensing modalities on the prototype requires that a computer vision system be designed that

1. Takes multi-sensor image data as its input,

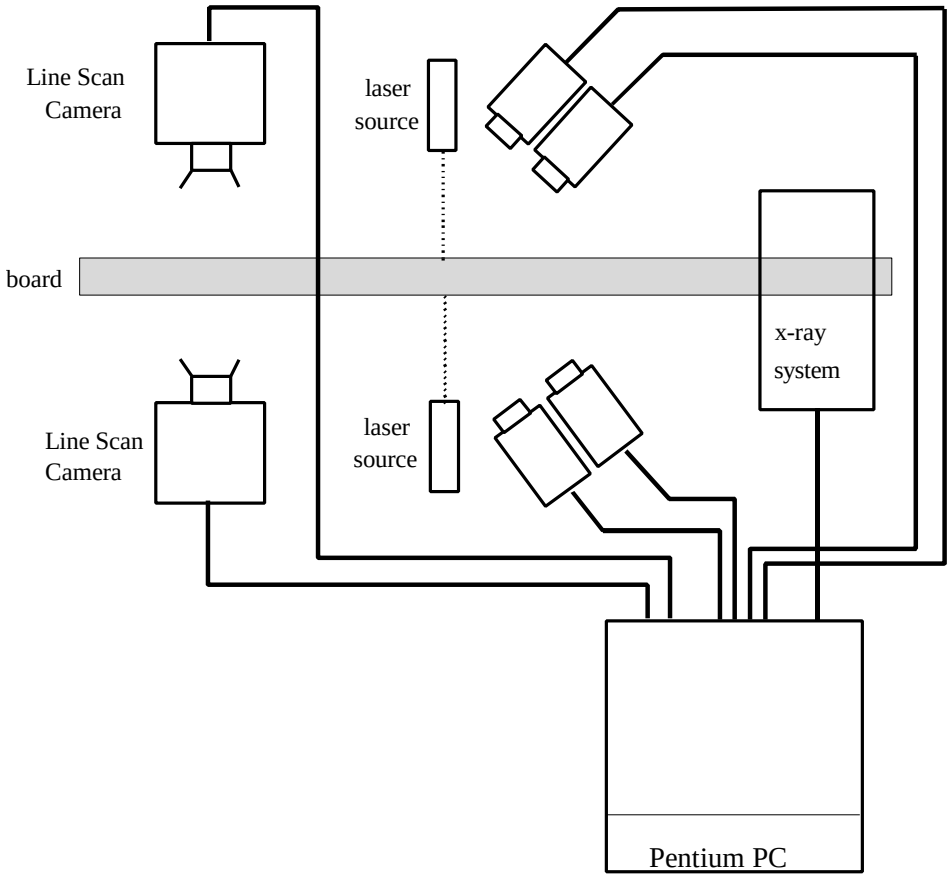


Figure 1.1 Multi-sensor Vision System

2. Detects various features in the input images, e.g., both surface and interior defects,
3. Works at a speed that matches the speed of production line,
4. Provides accurate and repeatable recognition results,
5. Provides a good deal of flexibility so that it can be easily tailored to other applications.

1.1.5. Sensor-Fusion Approach to Automatic Hardwood Lumber Inspection

As discussed earlier, different types of imaging sensors are required for a system to detect features in wood. To detect these features in wood, the system must combine or fuse information from all imaging modalities. How and when to fuse multi-sensor data has been the subject of much research since the 1980s. Many sensor fusion techniques and multi-sensor systems have been developed in recent years [PAU83, DUR87, NAK89, WAL90, LOU92, MUR96]. Most sensor fusion techniques are designed to address the problems in robot navigation, in automatic

target recognition, and in military command and control. However, little work has been done on fusing information from multiple image sources for the purpose of industrial inspection in hardwood lumber secondary manufacturing.

The combination (or fusion) of imagery from the multiple sensor mentioned previously can provide information about all the important features in lumber. However, the amount of data to be processed increases with the number of sensors used and makes real-time processing more difficult. Hence, research is needed to further develop sensor fusion methods applied to automatic lumber inspection that is both accurate and computationally efficient.

1.2 Objectives

This introduction has described several machine vision systems that have been of limited utility in automatic lumber inspection applications. The overall goal of this dissertation is to create a sensor fusion methodology and incorporate it into a machine vision system design for more robust defect detection in hardwood lumber. A novel sensor fusion approach was developed that provides a flexible knowledge representation framework, which can operate on multiple channel sensor input. The knowledge framework incorporates fuzzy logic concepts to extract and represent knowledge about how lumber features are seen in the various sensor images. The vision system outputs a "feature map", a complete description of features found in lumber including geometry, appearance, and structural information of lumber. This output lumber description can subsequently be used by applications software that controls the actual processing of lumber

To meet this overall goal requires that five tasks be accomplished. The first task is to develop image preprocessing algorithms and image registration algorithms for all three imaging modalities. The images preprocessing algorithms include functions such as image registration, edge detection, shade correction, crack-preserving filtering, and histogram extraction. Because some of these functions are computationally intensive, their implementation in special purpose preprocessing hardware is key to meeting the requirement of real-time processing. Therefore,

these algorithms are designed to be implemented in special purpose hardware that is currently being developed in the Spatial Data Analysis Laboratory at Virginia Tech.

The second task of the research is to develop low-level and intermediate-level vision algorithms. These algorithms are needed to extract image primitives (such as components or regions) and to characterize each of such image primitives with a measure list that could be used to describe lumber features (more commonly referred to as defects) effectively. The algorithms must be fast enough to meet the requirement for real-time processing and be able to extract measures of such components or regions reliably at the same time.

The third task is to design algorithms for high-level visions that identify a component (a penitential defect) found by the lower-level algorithms, and construct a consistent description of the component. The measures of the component extracted from one of the three images (color image, laser range image, and X-ray image) are primarily used to determine if it belongs to one of the defect classes in Table 1.1. But if additional information is needed, the algorithms will extract more measures from the other two images, and then combine the information from all three images and subsequently identify the component.

The fourth task of this work is to investigate the techniques used in sensor or data fusion, to design approaches to the computer vision system, to apply appropriate approaches to detect defects, and to reduce the amount of data that needs to be processed in the overall process of defect identification.

The final task is to integrate and test the low-level vision modules, high-level vision modules, domain knowledge sources, data fusion modules, and scene-analysis control modules into an integrated software system that will run in real-time on an architecture that has been selected for the vision system implementation.

Completion of the overall research goal will result in several key contributions to the basic science of machine vision systems. First, this work will complete an automated multiple-sensor

lumber scanning system prototype. The prototype machine vision system algorithms developed in this research will be the first such system that can accommodate multiple sensor image information to identify both surface and internal characteristics in hardwood lumber.

The second contribution will be the development and testing of a new and efficient approach to fusing the information from multiple imaging sensors. With this approach, a defect region is identified by systematically extracting and analyzing measures of components from multiple images. Efficiency of the approach is maintained by creating a unique system that locates and labels components of interest from spatially registered images and extracts component measures only from those smaller components of interest.

The development of a novel histogram-based global thresholding technique followed by a connected-component labeling algorithm will be the third contribution of this work used for dynamic image segmentation of the various multiple-sensor images. The connected- component labeling algorithm is a new procedure developed for processing and labeling multiple-valued thresholded images.

A fourth and final contribution of this research will be a fuzzy logic approach for defect identification of labeled regions-of-interest. This new approach is the first developed for lumber image processing and defect recognition. The results of this research will show how fuzzy logic is an efficient approach to handling the heterogeneous nature of defects in hardwood lumber.

1.3 Assumptions

The following assumptions have been made in the design and development of the vision system, and will be tested whenever possible during the course of this research work:

1. The sensing modalities used in this study, i.e., color cameras, laser profiling, and x-ray imaging, span the breadth of wood inspection problems. It is generally believed that improvements of classification accuracy can only be achieved by

incorporating additional independent measurements, the type of measurements provided by additional sensing modalities [PAU82, PAU90].

2. The increased hardware complexity of the multiple-sensor system will be more than offset by improved accuracy in both the location and identification of important features. In addition, the increased cost of system hardware will be more than offset by the reduced complexity of the computer vision software and computing machine. Moreover, supplementary sensing hardware is also needed to increase the robustness and reliability of the system and decrease its probability of failure in an industrial environment.
3. While specific measurements were taken and various parameters were estimated for red oak lumber, the overall methodologies presented in this study can be naturally extended to other wood species as well. Red oak is generally considered one of the most difficult wood species in terms of automatic feature recognition.
4. The surface area where features are located represents a small percentage of the total board surface. This assumption is generally true for the hardwood lumber used in secondary wood manufacturing operations.
5. Wood defects (or features) can be classified into a number of defect classes. Each defect class can be pre-defined to have a unique set of measures that differ from other defect classes. A defect is assigned to a class which possesses those measures closest to a pre-defined class.

1.4 Limitations of the Study

There are a number of limitations to this research. First, the validity of this research was demonstrated primarily on red oak lumber. While specific defect classes and measures defining those classes may vary, the basic principals and procedures developed in this research can be easily applied to automatic feature recognition of other hardwood lumber species. As mentioned earlier, red oak is one of the more difficult species to accurately demonstrate automatic feature recognition algorithms. Therefore, it is anticipated that the performance of the system based on

inspection of red oak demonstrates this system's potential ability to inspect other hardwood lumber with relatively minor modifications to the algorithms and software.

Another limitation of the research is that only dry-planed lumber was used. Dry lumber ensures a uniform moisture content, which minimizes feature variability associated with moisture content in wood. Planed lumber ensures a "clean" surface which reduces the amount of feature variability that can arise from surface roughness, soil marks, and other factors. This limitation can be justified in many practical applications such as secondary manufacturing where lumber material is normally dried and planed before the inspection process. Note that applications exist where it would be advantageous to automatically inspect green (undried) rough (unplaned) lumber. While the results of this research may serve as a good starting point for automatic inspection of such lumber, further research is required to overcome some of the unique feature variability encountered in green rough lumber.

1.5 The Dissertation Outline

This dissertation is organized into 10 chapters. The Chapter 2 introduces the concept of computer vision systems, sensor fusion, and fuzzy logic. This chapter provides a background into various technologies and the theory needed to build a modern computer vision system. A variety of computer systems are also reviewed, including systems proposed or developed by researchers, patented systems, and commercial systems. Chapter 3 provides the focus and definition of the inspection problem to be addressed in this study and provides an overview of the computer vision system for automatic lumber inspection. It also provides a comparison of the proposed system with some existing systems and how the proposed system overcomes some of the key challenges in developing automatic lumber inspection systems. Chapter 4 describes the preprocessing performed by the MORRPH board and by the data collection software. Chapter 5 provides the algorithm for doing foreground/background differentiation. Chapter 6 presents a detailed description of algorithms in the analysis of laser range images. Chapter 7 describes the analysis of X-ray images and how information from other images is combined and used for more robust defect identification. Chapter 8 describes the analysis of color images to identify the smaller,

more subtle surface defects in wood. Chapter 9 will discusses experimental results that demonstrate the ability of the system to detect various hardwood lumber features. Finally, the summary and conclusions are provided in Chapter 10.

Chapter 2

Computer Vision Systems and Sensor Fusion: A Review

Computer vision is the science and technology of understanding a scene from its projected images. The goal of research in computer vision is the understanding of complex visual processes and the construction of effective computer-based visual systems that can give an explicit, meaningful description of physical objects from input images. The field involves such diverse disciplines and areas as cognitive psychology, neurophysiology, mathematics, pattern recognition, image processing, computer systems hardware and software, computer graphics, electrical engineering, robotics, and knowledge engineering in artificial intelligence. However, in general, work in computer vision roughly falls into two main categories of research. The first seeks to develop a coherent theory of visual perception and to develop an understanding of how the human vision system works (computational vision). Researchers in this group often build computational models of human vision processes. The second category does research and development on applying the science and technology of computer vision to real world problems (machine vision). Researchers in this group focus on developing machine vision systems that represent economical solutions to industrial, medical, and military inspection problems.

A machine vision system or a computer vision system generally refers to a system consisting of both hardware and software components that are designed to solve one or more real world problems. Machine vision systems, computer vision systems, and robot vision systems have been used interchangeably in the literature. The definition of the major components of a machine vision system has also varied in the literature [KAH93, NG93]. The hardware components include imaging sensors (and/or image acquisition equipment) and image processing hardware (usually a computer). The hardware components may, in addition, consist of illumination sources, optics that couples the optical image to a sensor, a material handling system and so on. The hardware

components of a computer vision system usually depend on its application. The components of a vision system used for industrial inspection may differ substantially from the components of a machine vision system used for robot guidance. The software components generally consist of computer vision software that performs image processing and scene analysis functions, a domain-dependent knowledge base, a user interface that receives commands, a graphical display for showing processing results, and control software that controls the imaging sensors and/or materials handling.

During the past several decades, there has been an increasing need for faster and more effective image processing hardware. The bottleneck in machine vision throughput caused by limited hardware speed has now been largely overcome by the rapid evolution of new architectures and high-speed processors. Despite significant hardware advances, the construction of effective general-purpose computer vision systems has still proven to be exceedingly difficult if not impossible to do. However, many special-purpose computer vision systems have been developed and successfully applied to a variety of industrial applications [CHI86, BAT97], such as the real-time industrial inspection of planer web materials and robot guidance.

The focus of this research work is to develop a computer vision system designed for industrial inspection, more specially, the inspection of hardwood lumber used by the secondary manufacturing forest products industry. For the purpose of this work, the term "machine vision system" refers to an overall system that consists of both hardware and software components. The term "computer vision system" specifically refers to the software system that performs image processing and scene analysis.

The purpose of this chapter is 1) to identify the major components of current computer vision systems used for industrial inspection; 2) to survey the literature to determine the current state-of-the-art in machine vision system technology, with a focus on systems that are applicable for wood inspection and their software components; more specifically, the algorithms, implementation, and performance of the computer vision software; 3) to identify limitations of current computer vision systems; and 4) to demonstrate that the largest limitation common with

most computer vision systems designed to inspect wood products is ultimately the result of employing a single sensing modality.

2.1 Classical and Modern Computer Vision Systems

The most popular computer vision system paradigm consists of image preprocessing and segmentation followed by interpretation. Horn [HAK92] suggested that, in general, a computer vision system is composed of the following three processes:

1. Low-level processing modules that perform image processing functions.
2. Intermediate-level processing modules that deal with the task of extracting and characterizing objects (e.g., regions) in an image that has undergone low-level processing.
3. High-level processing modules that perform the scene analysis functions, i.e., identifying relations among patterns and more abstract scene identification tasks.

The definition of these three levels of processing varies in the literature and there are no clear-cut boundaries between the above processes, however these subdivisions provide a useful framework for categorizing the various processes in computer vision systems.

Computer vision systems can be divided into two classes: classical (conventional) and modern (intelligent) computer vision systems [KAH93]. Perhaps the major difference between the classical and modern vision systems is how domain knowledge is structured and then applied to interpret objects found by lower-level processing functions. Modern vision systems tend to incorporate more and more artificial intelligence and knowledge-based technologies [BAT93].

2.1.1 Low-level Processing Technologies

Low-level processing roughly refers to those operations that are performed directly on the numeric arrays (usually two-dimensional arrays, an element $C[x, y]$, of these arrays represents a pixel) of sensory data that represent digital images. A detailed discussion about the representation and modeling of digital images can be found in [GON92, JAI89]. These operations include such

functions as image filtering, image shading correction, and image segmentation. The purpose of image filtering is to remove or at least to suppress noise generated by the image sensing device or the data collection electronics. Image filtering techniques include operations such as low-pass filtering, Gaussian smoothing, or median filtering that can be found in image-processing literature [BAL82, ROS82, YOU86, JAI89, PRA91, GON92].

Shading correction [SAW77] removes distortion of gray levels caused by a non-uniform illumination source and unequal distances from different parts of the image to the focal plane of the camera. The camera output (image data) can be modeled as a nonlinear function of camera input:

$$g(x, y) = N(f(x, y)) \tag{2.1}$$

where $g(x, y)$ represents the camera output to a given input $f(x, y)$, (x, y) denotes the pixel position, and N denotes the nonlinear function. The input $f(x, y)$ may be viewed as the product of a reflectance component $r(x, y)$ and an illumination component $i(x, y)$. Sawchuk [SAW77] showed that g(x, y) can be approximated by a linear function of $f(x, y)$ with two constant coefficients to be determined:

$$g(x, y) = a_0(x, y) + a_1(x, y) f(x, y) \tag{2.2}$$

With a uniform black surface $f(x, y) = 0$ (represents the minimum pixel value in the image) and a uniform white surface $f(x, y) = 255$ (represents the maximum pixel value), and the measured outputs from the camera, the two values a_0 and $a_1$can be determined. Then the image without shading distortion can be obtained from equation (2.2) on a pixel-by-pixel basis since both coefficients are dependent of pixel location (x, y). Since the implementation of the above operation has to be performed on a pixel-by-pixel basis, the shading correction is generally computationally intensive and, therefore, time consuming, if it is done by software.

Image segmentation partitions an image so regions that might potentially contain features of interest are explicitly marked. A region generally refers to a group of pixels that share similar measures such as color, intensity, texture, etc. Segmentation is regarded as a process of pixel classification [ROS82]. Segmentation is the first important step in the analysis and understanding of an image. To date, numerous segmentation techniques have been proposed and developed

[OHL78, OTH80, ROS82, HAR85, POS92, CHU93, BHA95]. These techniques can be roughly grouped into two major approaches: boundary-based and region-based segmentation [NEV86].

In boundary-based segmentation methods, edges in an image are detected with an edge operator applied to each pixel in the image. The edge points that correspond to object boundaries are those that have an abrupt gray-level change. These edge points are then connected to form, hopefully, complete boundaries of regions. The problems with this approach usually are: 1) sample edge detectors such as Roberts, Prewitt, Sobel operators [BAL82, PRA91] often generate meaningless or false edges due to image noise, while advanced edge detection methods such as LoG (Laplacian of Gaussian) [BAL82, MAR80], Canny [GON92], and zero-crossing [HAR84] are computationally complex; and 2) to form complete boundaries by resolving incomplete or false edges can be very difficult and time-consuming. Usually a well-defined model of the world is required to group these line segments into objects. Thus in real-time computer vision systems that are applied to industry inspection, such boundary-based approaches should be used only in those applications where edge information is absolutely needed and where the edges are well-defined.

Region-based methods can be divided into two groups: 1) methods based on finding the regions directly such as region growing [AGG77, FU81, CON92], region splitting, and merging [OHL78, NEV86, JAI89, PRA91]; and 2) methods accomplished via thresholding based on features or distribution of pixels [NEV86]. As to the methods in the first group, region growing technology [AGG77, GON92], an image is divided into *atomic regions* [AGG77, JAI89] of constant gray level or color. Similar adjacent regions are merged sequentially until the adjacent regions become sufficiently different based on some similarity measure, e.g., gray level, texture, or color. In practice, individual pixels are often used as atomic regions. The region-growing methods generally start with a set of "seed" points and from these seed points regions are grown. Some fundamental problems with region growing are: 1) how to select initial seeds that will represent regions of interest; 2) how to choose suitable measures for including points in the various regions during the growing process; and 3) how to formulate the reasonably complex constraints that must be placed on the growth pattern to achieve acceptable results. Using region-

based image segmentation methods, reasonably accurate results have been achieved on simple scenes but not on complex scenes [PRA91]. With region splitting and merging approaches, an image is initially divided into a set of arbitrary, disjointed regions. These regions are then merged or split-based on their measures such as intensity, texture, or color. In general, techniques in the first group have a computational complexity that is proportional to the image size.

For methods in the second group, segmentation often consists of two steps of processing:, image thresholding followed by connected-component labeling. Thresholding techniques can consist of either a single thresholding or multi-level thresholding [GON92]. In single thresholding, a thresholded image $BW_T(i, j)$ is defined as

$$BW_T(x,y) = \begin{cases} 1 & \text{if } BW(i,j) > \mathrm{T} \\ 0 & \text{if } BW(i,j) \leq \mathrm{T} \end{cases}$$

where T is the threshold that is used to separate objects from the background. Obviously, this method should be used on images that have a distinct bi-modal histogram in which a valley point divides the image into two classes. Unfortunately most real world histograms seldom have this idealized shape. Hence, in practice, multi-level thresholding methods are often needed. In these methods, thresholds T_0, T_1, …, T_k are used, and pixels with gray levels falling between the two thresholds T_i and T_{i+1} are labeled with label L_i. In using techniques based on thresholds, one must develop an algorithm for selecting the thresholds. Numerous threshold selecting techniques have been developed by researchers, which include simple global thresholding [HAR85, PRA91], multi-level thresholding [PRA91], optimal thresholding [GON92], adaptive thresholding [SAH88], and dynamic thresholding [SAH88, GON92, BAY93].

The step following image thresholding is connected-component labeling. Connected-component labeling is performed on a thresholded image to form regions. The operation assigns pixels that are connected (either 4-connected or 8-connected, as defined in Figure 2.1) the same region label. Many methods for performing connected-component labeling have been proposed

(a) (b)

Figure 2.1 Connectivity on a grid. A pixel is said to be 4- or 8- connected if it has the same properties as one of its nearest four or eight neighbors. Pixel A and its (a) 4-connected and (b) 8-connected neighbors

and studied [SHI87, JAI89, GON92]. Unfortunately most of these methods deal with binary images [BAL82, SHI87, JAI89, GON92]. Little research work has been reported on connected-component labeling algorithms that can be applied to multiple-thresholded images.

The computational complexity of thresholding-based methods are determined primarily by the complexity of the threshold selection process used and the connected-component labeling scheme that is employed. Numerous techniques for image thresholding and connected-component labeling have been developed during the past two decades and some of these methods have been implemented using a variety of different processing hardware. These techniques will not be discussed in detail here since they can be readily found in the literature [OTH80, OHL78, HAR85, NEV86, SAH88, POS92].

After more than two decades of intensive research [JAI95, BHA95], segmentation has proven to be the most difficult problem of all the low-level processing operations. Although a large number of segmentation techniques have been developed [SAH88], no general segmentation technique has been developed that can perform adequately on a diverse set of images. Numerous modifications to an algorithm's control parameters are usually required to adapt the algorithm to a new problem domain. Even when these parameters are optimally adjusted there is no guarantee the procedure will provide satisfactory performance. Besides the above two major approaches, other segmentation methods have been proposed by researchers, which include segmentation using genetic algorithms [BIR95], segmentation with variational approaches [SCH96], and segmentation with mathematical morphology [ZUE88]. These methods have generally been reported to achieve better results on certain kinds of images, but they are often computationally intensive [ZUE88].

The above discussion has indicated that more robust segmentation algorithms often require more complex pixel-by-pixel based operations, and these operations are generally computationally demanding. For industrial inspection, a basic requirement for any region segmentation method is that its implementation must be fast enough to match production speed. Therefore a tradeoff between processing time and segmentation accuracy must be considered

when one selects a segmentation algorithm. Appropriate algorithms should provide reasonably accurate results at near real-time speeds, where real-time is defined by production speed.

2.1.2 Intermediate-Level Processing

The objective of intermediate-level processing is to characterize the image primitives, such as edges and components (or regions) found during the segmentation operation, with a set of measures extracted from these primitives. These measures are then combined together to form a measurement vector. This measurement vector is then used by the high-level process to label objects in the scene. Many types of measures have been proposed by investigators [BAL82, YOU86, CHO91, GON92, HAN93]. These popular measures include the following:

1. Area, the number of pixels in region R;
2. Average gray level, which may represent brightness, density, or thickness;
3. Center of mass,

$$(\bar{r},\bar{c}) = \left(\frac{1}{N} \sum_{(r,c)\in R} r, \frac{1}{N} \sum_{(r,c)\in R} c \right);$$

4. Minimum inscribing rectangle (MIR), described by its upper left vertex, (r_1,c_1) and lower right vertex point (r_2,c_2), where

$$(r_1,c_1) = \left(\min_{(r,c)\in R} r, \min_{(r,c)\in R} c \right),$$

$$(r_2,c_2) = \left(\max_{(r,c)\in R} r, \max_{(r,c)\in R} c \right);$$

5. Compactness,

$$CP = \frac{(perimeter)^2}{4p \cdot (area)};$$

6. Ratio of height to width,

$$R = \frac{height}{width};$$

7. Perimeter, the pixel length of the boundary;

8. Color [OHT80, BRU90], represented by either red, green, and blue components (RGB color model) or hue, saturation, and intensity (HIS color model);
9. Elongatedness, defined by

$$E = \frac{e_1}{e_2}$$

where e_1 and e_2 are eigenvalues of the covariance matrix of the distribution of ($r - \bar{r}$, $c - \bar{c}$) and $e_1 > e_2$;

10. Orientation measure, or the direction of the major axis, which is defined as the angle θ that minimizes the following equation

$$I(q) = \sum\sum\left\{(y-\bar{y})\cos q - (x-\bar{x}\sin q)^2\right\}$$

Orientation may also be defined as

$$Orientation = \frac{1}{2}\tan^{-1}\left[\frac{2m_{1,1}}{m_{2,0} - m_{0,2}}\right]$$

where $m_{i,j} = \sum_{(x,y)\in R}\sum (x-\bar{x})^i (y-\bar{y})^j$.

11. Texture measures, higher-order moments, and curvature have also been used to describe regions [GON92, HAR92].

Ideally a measure should describe each class uniquely. Unfortunately, for most real world problems in industry inspection, defining an ideal measure is usually impossible since the number of defect classes is either large and/or the shape of defect is quite complex. In general, more than one measure is needed in defect classification and recognition. The selection of measures usually depends on a specific application. One often tries to extract measures with as little redundancy or overlap as possible that can be used to distinguish one defect from others. Often, different classes of defects are overlapped in the measure space, which makes classification difficult.

For industrial inspection systems, another difficulty in measure selection comes from the requirement for real-time processing. The extraction of measures is performed only on regions of interest and, if the number of regions is small, the resulting computational complexity is often manageable. Nevertheless, the extraction of numerous measures or complex measures such as

texture and orientation measures can still be time-consuming. Thus the number and complexity of measures should be selectively used only in applications where they are critical for the accuracy of intermediate-level processing.

2.1.3 High-level Processing

The high-level processing of images attempts to interpret the relevant components (regions) of an image in the context of the goals and *a priori* knowledge of the domain. The purpose of high-level processing is to interpret the objects present in an image and analyze the complete scene. To accomplish this purpose, the high-level processing performed by most modern computer vision systems usually incorporates artificial intelligence (AI) methods to direct the reasoning and control processes. Many high-level processing techniques have been developed by computer vision researchers [BAL82, CON87, DAR88, NTU89, CHO91, KAH93]. These technologies include model matching, neural network, and syntactic analysis approaches.

In a model matching approach, objects of interest are often first modeled by a template image (reference image), or a set of measures that uniquely describe objects in a class [LON94, FOR97]. An object (or region) is recognized by determining where the measures of the object sufficiently match the measures of a reference image. The performance of model matching approaches can be affected by the class distribution in measure space and/or the accuracy of object modeling.

Neural networks have been used to classify and recognize objects in computer vision research [WAN90, ARG90, CHO91]. In this approach, a multi-layer feed-forward neural network is first trained with measures of known objects; then the neural network is used to classify a given object with a set of measures. In neural network approaches, sufficient amounts of training data are required to train the neural networks. Oftentimes, such sufficient training data is not available. Even after training, there is no guarantee a neural network will provide satisfactory performance [COX94] for real world problems.

In syntactic analysis approaches, an object is described as a string, a tree, or a graph of image primitives and their relationships. The syntactic methods constitute a language theory of vision, which includes sets of image primitives and a formal image grammar [COH82]. Objects in an image are regarded as sentences in a language defined by a formal image grammar. Just as an English sentence might consist of a noun followed by a verb phrase, an object might contain a top followed by a side. Recognition of an object in an image is analogous to the parsing of an English sentence.

A problem with syntactic methods is how to define appropriate primitives to represent objects of interest, and to find mechanisms capable of detecting and extracting these primitives from an image. Simple primitives are easy to detect, but a more complicated grammar is often needed. While more complicated primitives can reduce the complexity in grammar, they are difficult to extract. Syntactic methods are usually applicable only when the image contains a small set of well-defined, easily recognized primitives and when the relationships between primitives are simple [COH82]. For complex scenes, designing sophisticated grammars becomes more difficult and sometimes impractical.

As one might see from the above review, despite considerable research and development over the past few decades, the computer vision field is still in its infancy. The research and development of the field is often driven by concrete and well-defined applications, and as a consequence, the resultant techniques or systems are very specific solutions to the specific applications, working satisfactorily only in a very restricted environment [KRO96]. Literature on computer vision-related fields is vast. Detailed discussion and comparison of numerous algorithms or methods used in the computer vision field are beyond the scope of this research, and can easily be found in the literature.

2.1.4 Scene Analysis Strategies

The previous sections have discussed various techniques for the three main processing levels involved in computer vision applications. For effective applications, these processing levels

must be systematically controlled and coordinated. Three basic control approaches to scene analysis have been developed [BAL82, CON87, ARG90]: 1) bottom-up type approaches in which the analysis starts from pixels to image regions to object labeling; 2) top-down type approaches in which analysis begins with object models to image regions to pixels; and 3) combined or mixed approaches that use a combination of bottom-up and top-down strategies.

Bottom-up approaches have been used in computer vision research for several decades [ROB65, CLO71, CHI88, JAI95]. With this type of approach, image data is processed by low-level processing followed by intermediate-level processing where various measures are extracted, measures that describe regions (or other image primitives) found by the low-level processing. These measures are then matched to measures of known object models, which the system anticipates or is designed to recognize. A region is assigned to an object where the region's measures most closely resemble it. Bottom-up approaches are data-directed methods. The advantage of the data-directed approach is that little prior knowledge concerning the context of the region is required. The disadvantage of these approaches is that it lacks direction: it does not know where in the image to look for objects. Therefore, an extensive and time-consuming search process is required to locate regions of interests. Another problem with bottom-up approaches is that they are sensitive to noise since errors caused by noise at a lower level of processing propagates to subsequent levels of processing. Since bottom-up approaches are entirely data-driven there is no way to effectively prevent the propagation of noise.

Top-down approaches, on other the hand, are basically goal-directed methods. In a top-down approach, one starts with a hypothesis about what objects are in the image and what are their features. Each hypothesis is then decomposed into sub-hypotheses that support the hypothesis, and so on. Each hypothesis is backed up by hypotheses at one level lower in the hierarchy. The lowest level hypotheses must be verified by extracting image features with image processing and analysis methods. If a particular hypothesis is found to be invalid, a new hypothesis is formulated and, subsequently, attempts are made to verify this new hypothesis. The generation of the new hypothesis is independent of previous processing as well as the information obtained from the image. The advantage of top-down approaches is that known measures of

various objects can be used to direct the search for objects. However, top-down approaches depend very much on high-level control of low-level processing, and therefore are domain-dependent [COH81]. It seems that top-down approaches should only be applied to highly structured scenes. It has been argued that a pure top-down approach is not suitable for images with a large amount of variability in their structure [ARG90].

Combined or mixed approaches combine the advantages of both bottom-up and top-down approaches. Such approaches generally start with a bottom-up approach that is used to generate a list of possible objects. Then top-down approaches are applied to verify hypotheses formulated from the list of objects generated during the bottom-up processing.

Of the three kinds of approaches, the combined approaches are generally more efficient, effective, and robust [COH82, CHO91] particularly for natural scenes that can have considerable variability. As discussed in next section or Section 2.2, images involved in lumber inspection have immense variability in their structure. Therefore, the proposed computer vision system developed in this research uses a combined approach in the high-level processing.

2.2 Machine Vision System for Wood Inspection

The demand for premium wood species has increased in both the domestic and international markets [BUS90, LUP91] and the availability of this material is in limited supply [LUP94, BEC91]. These trends could ultimately result in supply shortages and price increases in hardwood lumber [LUP95, LUP91]. A major user of this lumber is the furniture, cabinet, and dimension industry which uses an estimated 36.5 percent of the hardwood lumber in the U.S. [LUP93]. With increasing shortages and prices, the wood-processing industries such as these are spending more time and money than ever to inspect lumber and carefully remove those features that have a negative impact on the quality and value of the final product.

Considerable work has been done concerning the ability to automatically detect features in wood using various non-destructive evaluation methods [SZY81, POR92]. Many methods have

been shown to be promising in detecting defects in wood including optical, ultrasonic, microwave, nuclear magnetic resonance (NMR), and x-ray sensing techniques. Most of the machine vision systems developed for wood inspection applications that have been reported in the literature are designed for the surface inspection of a range of products including lumber and panels [CON89, HAT89, CHO90b, CHO90c, CIC92, BAY93, POE95, MOO95]. While research on such machine vision systems has produced marked results in recent years, commercial applications based on these methods have not been able to meet the industry's expectations. Accurate and reliable feature detection in wood has been a challenge mainly due to the enormous variability found in wood materials processed by the industry.

Wood is a natural material and, therefore, has variability both within and between species. This variability can lead to undesirable characteristics and features that are considered to be defects such as knots, stain, tension wood, and pitch pockets. Defects can also be created throughout the handling, conditioning, and manufacturing process. Internal checking, surface checking, wane, and various machining and handling damage can be caused in this process. These defects are of economic concern to all producers and users of lumber because the grade and ultimately the price of the material is based upon defect size, location, and type. Since manufacturers in the wood furniture industry are concerned with those features that affect the *visual appearance* as well as the structural integrity and stability of their products, they must consider practically every feature in wood.

As demonstrated earlier in Table 1.1, there are many important features that affect the value and quality of wood. These features are generally considered as "defects" that must be removed before the wood material can be used in the final products. The severity of these defects depends on the type of product being manufactured. The most critical features in hardwood lumber can be classified into three main categories. These are: 1) visual surface features such as knots, holes, splits/checks, color surface stains, etc.; 2) geometric or shape features such as wane, crook, thickness variations, voids, cup, etc.; and 3) internal features such as density variations, internal splits, honeycomb, and decay. No single-sensor machine vision system in existence has

been able to reliably detect all of the above feature categories in wood. Ideally, each category of features requires a different detection mechanism and sensor.

There are a number of inspection systems that have been specifically designed to identify surface features in wood. Conners, et al. [CON83, CON89] studied the automatic inspection of rough lumber. In this approach, the basic strategy was to divide the collected digital image of a board into a number of disjointed rectangular regions and to classify each independently using a pair-wise Bayesian classifier. This procedure lends itself to a parallel processing implementation. However, there are a number of problems with this approach. First, it is not suitable for detecting small surface defects like splits and checks. Second, the evaluation of the texture measures used to characterize each region is computationally demanding.

Conners and Cho [CON89, CHO90a, CHO91] proposed and implemented a knowledge-based machine vision system for automated industrial web inspection. The objective of this system is to automatically locate and identify surface defects of the web material being inspected. The vision system conceptually consists of two modules: a low-level segmentation module and a high-level recognition module. The low-level module performs image segmentation and performs component measurement extraction. The high-level module does the scene analysis, i.e., recognizes defect components. The high-level module can further be decomposed into three parts: 1) a focus of attention mechanism that is triggered by confidence values of defect areas; 2) defect-detection procedures where each of these procedures is designed to detect a special defect using domain knowledge; and 3) verification procedures that employ contextual constraints. The module employed a rule-based approach to perform defect label verification [CHO91]. The rule-based component employed a blackboard system to control the reasoning. The system was tested on some 100 gray level images (with typical size of 512x480) of rough hardwood lumber and achieved about a 79 percent overall correct classification rate on all defects it was trained to identify. However, this system was specifically designed to detect surface features but not internal or geometric features.

Polzleitner [POE95] described a method for identifying surface defects on wooden boards using syntactic pattern recognition. In this method, raw-image data was reduced to a considerably smaller number of syntactic symbols by symbolic extraction and region segmentation. The syntactic method was found to give correct classification results on a number of test images, and fulfilled the accuracy requirement for the inspection problem it was designed to address. But it was not possible to provide a real-time implementation of this algorithm.

Pietikainen, et al., [PIE91] developed a vision system for industrial inspection that employed an experimental hybrid architecture. This system was applied to the wood surface inspection problem. In the system, MaxVideo (from Datacube Inc.) machine vision cards are used to perform low-level vision operations, whereas a general-purpose multiprocessor (MIMD type) consisting of several transputers is used for high-level analysis. A SUN workstation was used to control the operation of the system and provided the user interface. One advantage of the hybrid architecture is that it allows new algorithms and more computational power to be added in when necessary. A hierarchical-connected component (HCC) algorithm was used for segmentation. The basic version of the HCC algorithm consists of two stages: the first one merges pixels, and the second one merges regions, according to their edge contrast in color space or gray-level. The first stage contains two passes that are similar to those of the conventional connected-components labeling. The image is scanned, row by row, from top to bottom, left to right, and pixels are assigned labels by comparing them with the two (4-connectivity) or four (8-connectivity) adjacent labeled pixels in the first pass. In the second pass, equivalent labels are resolved. In the second stage, two adjacent regions with an average edge contrast smaller than a threshold (ε) are merged, and the region adjacency graph is updated. The high-level analysis is performed on the transputers with an object-oriented image processing library; however, no detailed description of this library was given. The experimental results showed that the use of color information substantially reduces the ambiguities in classifying potential defects as compared to the results obtained from gray-level images. The hardware support for real-time HCC was under development. No information was given as to whether the system used any knowledge-based approaches in doing its analysis, nor was there any information provided about the accuracy of the system as it ran in real time.

Ciccotelli and Portala [CIC92] described a vision system for automatic defect detection to optimize the cross-cutting of lineals, i.e., relatively narrow strips of wood produced during a gang-rip operation. The system was equipped with a matrix camera and a set of mirrors. It was designed to analyze three sides (top, left and right sides) of pine lineals simultaneously. The analysis results are used to determine the optimum cutting plan for the cross-cut saw. A 25 percentage improvement was cited but no detailed information about the nature of the inspection software was provided. Apparently, this system cannot be adapted to detect all features in hardwood lumber due to the difference between softwood and hardwood, as well as limits of sensors used in the system. Due to greater variability present in hardwood lumber, lineals, or parts, is known to be a much more complicated problem than the inspection of softwood lumber.

Blackman [BAL95] described a vision system that was to be part of a ripsaw optimizer. The total system was called RipScan. The vision system is designed to detect only surface features such as wane, grain, knots, and splits in pine lumber. It employs a single sensing modality--multiple color cameras. The author mentioned that this vision system utilizes a "massive parallel neural net" to identify features in wood. No other details about the system are given [BAL95]. However, it takes the system's Pentium-powered computers close to 40 seconds to analyze an oak board and find the best cutting solution before the board is ripped. Real-time operation of this vision system is in question without the use of multiple computers.

Moore [MOO95] patented an optical scanning system for the detection of surface defects on lumber. The system utilizes a single optical sensing modality which first scans the top face, then the bottom face and sides of lumber through a set of cameras that can collect images of two faces at a time. A board-rotator is used to rotate a board 180 degrees so that these same cameras can be used to scan the other surface of the board. The scanned images are then analyzed by the computer vision system to look for surface defects. Finally, the surface defects with size, type, and location are evaluated by the system according to a set of grading rules, and then the grade of the lumber is determined. No details about the system's defects-identification algorithms are given [MOO95]. There are several problems that must be overcome to apply this system to hardwood lumber inspection. First, the system cannot locate and identify internal defects.

Second, the system cannot determine the three-dimensional profile of lumber. Third, the system can only process and grade one segment or part of a board at a time, hence the real-time operation is in question.

The X-ray imaging techniques, especially computer tomography (CT), have been primarily applied in log inspection to locate and identify internal defects such as knots and rot [FUN85, ZHU91, ZHU93]. Funt and Bryant [FUN85] described an approach in detail on how internal log defects can be detected by analysis of a log cross-section histogram of a density map developed by computer tomography using x-rays. The method is designed to detect internal defects such knot and rot only, and cannot be used to detect any surface defects. Zhu studied the feasibility of hardwood log inspection using CT images [ZHU93]. He described a viable automated technology for locating and identifying log defects. In his research, he first established a database of imagery and ground truth information to determine how the various defects manifest themselves in the X-ray images. Then he analyzed the characteristics of defects in CT images of hardwood logs. Subsequently, he developed an image segmentation module that extracts regions that might contain internal defect such knots, decay, and holes. Finally, these regions were analyzed by a knowledge-based scene analysis module.

Several investigations have been underway to develop more accurate wood inspection systems by combining different sensing methods into one system. Hagman and Grundberg proposed multivariate image analysis methods to classify features on Scotch Pine [HAG93]. The proposed methods required images of Scotch pine from different sensors (microwave, optical, x-ray). Data from these sensors are calibrated by multivariate methods to obtain a model for the predication and classification of wood features such as knots, bark pockets, and decay. The methods are computationally demanding, and the models that were developed were not tested on any material other than the Scotch Pine images used to generate them.

Aune, et al., [AUN89] patented a lumber optimizer system that detects wane in lumber with an optical profiler and identifies other defects such as knots, holes and rot with an X-ray imaging system. Knots are located and identified based on sudden changes of density, since there

exists a relatively sharp boundary between a high-density knot area and the density of clear wood. Holes, decay and voids usually have a lower density than surrounding clear wood. Pixels of defect areas are determined by the analysis of density map (x-rays image) of lumber. However, the precise way in which defects other than knots are found is not described in any level of detail. The defect information is then transferred to a computer where an optimal cutting plan is formed. Note that the profile data and the x-ray data are processed completely independently of one another and the results of both are passed to the computer that formulates the cutting plan. The limitations of this inspection system are: 1) it cannot determine whether a defect is purely a surface defect or purely an internal defect, an important differentiation in some applications, and 2) it cannot detect any defect that does not affect the density of wood such as surface stains. No performance information about the system was given nor was it indicated whether a commercial version of the system is available on the market.

Floyd, et al., [FLO89] described a multiple sensor method for determination of pith location relative to softwood lumber surfaces in U.S. Patent No. 4,831,545. The method uses a slope-of-grain detection sensor [MAT86] and an additional optical sensor employed to determine the presence of wane. Both faces of the lumber are first scanned with an optical sensor to locate and identify any wane that might be present. Then both faces are scanned by the slope-of-grain detector to determine the location and size of knots on each face. Clearly, the method cannot be applied to the detection of internal defects in lumber. Also, the method cannot detect surface discoloration. The inability to detect surface discoloration and internal defects severely limits the utility of this method to the hardwood lumber inspection problem.

Aune and Peter [AUN91] described another multiple sensor system for inspecting softwood logs in U.S. Patent No. 5,023,805. The system employs a laser profiler to determine three-dimensional log shape and three x-ray sources looking at the log from three different directions to locate internal defects such as knots and rot. Knots and rot are determined by sharp changes of log density. This system was designed to inspect logs, and log inspection is a completely different problem from lumber, lineal, or part inspection. Hence, this system may have a number of problems if it is applied to general hardwood lumber inspection. First, it cannot

detect surface defects such as surface stains. Second, it is hard for the system to locate small defects such as small holes and checks, which have an impact to product quality in some applications.

In summary, the open literature on wood product inspection indicates that there are no commercial inspection systems that can detect all three categories of wood features in wood, i.e., surface, internal, and geometric features in wood. Although the need for multiple sensors has been identified by researchers, no complete inspection system has yet been developed to detect all features in hardwood lumber in a practical manner, i.e., systems that can locate and identify enough of the wood features to be of commercial use and that can be operated at production speeds.

2.3 Other Industrial Inspection Systems and Vision Systems

The literature on machine vision systems for other industrial inspection is very rich. Consider, for example, the extensive and systemic survey of machine vision systems for industrial inspection that was done by Bayro-Corrochano [BAY93]. Special systems have been developed for various industrial inspection applications that include printed circuit board (PCB) inspection [DAR88, HAR88, BEN90, ITO91], solder joints inspection [BAY93], IC wafer inspection [EJI89], metal inspection [MOR94], food inspection [BAT93], leather surface inspection [POL94], film inspection [BAY93], and the inspection of many other products [BAY93]. As mentioned before, these machine vision systems cannot be directly applied to the inspection of hardwood lumber, lineals, or parts, but processing and analysis technologies might be adapted to help design a machine vision system for hardwood inspection. Hence some industrial inspection systems will be reviewed in this section. The review of these systems will demonstrate that they cannot be directly applied to hardwood lumber inspection. Another purpose of the review is to reveal the state-of-the-art of the current vision systems and technologies used in these systems that may be most useful in hardwood lumber inspection applications. It is anticipated that this review will also bring additional insight into the vision problem and its applications to hardwood inspection.

Darwish and Jain [DAR88] developed a rule-based approach for visual inspection. This system was applied to the inspection of PCBs. In this system, domain knowledge is represented by a set of rules governing the geometry and topology of primitives of input images. They found that this approach was better than other techniques because it removes the burden of precise alignment required by traditional approaches to image comparison. Hence false alarms normally caused by rotation and translation can be avoided. However, the rules that govern the geometry and topology of image primitives are hard to formalize. This difficulty is especially true for complex scenes. Therefore, the system cannot be directly applied to hardwood lumber inspection without major modifications.

Smyrniotis and Dutta [SMY88] described a knowledge-based vision system for recognizing and classifying man-made objects in aerial images. The system consists of three stages of processing: low-level, intermediate-level, and high-level processing. At the first stage, domain knowledge is used to guide low-level processing such as select image segmentation methods. The system incorporated a flexible framework for the inclusion of a number of low-level vision algorithms (such as edge detectors) and for the selection of such algorithms based on the given context. Researchers have realized that even at low-level vision processing, domain knowledge can help one select the appropriate algorithms [NAZ84]. One can obtain much better results by applying specific techniques to segment a given image than using other techniques. A classical split/merge technique was implemented as the region segmentation method, and reasonable results were achieved for an airport scene image. Little detail is given as to how many other segmentation methods were built into the system and how many images were used to test the system in [SMY88]. The intermediate-level processing is responsible for the creation of region profiles and providing an interface between low-level and high-level vision. In the high-level processing stage for each region found earlier, object profiles are created (one for each man-made object), and the object attributes of the region are compared with model attributes. The system assigns the label to the region whose model attributes have the best match to the object profiles of the region. Again, little detail is given about the performance of this vision system. The knowledge-based vision system illustrates a framework of modern vision systems, but it has limited applicability to lumber inspection for several reasons. First, its low-level processing

algorithms might not be fast enough for real-time operation. Second the system can only analyze black/white images. Finally, defects in lumber are very hard to model using generalized object attributes.

Brzakovic, et al. [BRZ93], described an expert system for analyzing and classifying defects in uniform web materials. The system consists of four sub-systems: 1) detection, 2) characterization, 3) feature analysis, and 4) classification. The function of the detection sub-system is to locate regions that contain defects. Unfortunately, the detection sub-system is not described in any degree of detail. The function of the characterization sub-system is to compress image data by extracting pertinent information, i.e., pattern vectors, from regions found by the detection sub-system. Invariant moments, spatial-domain intensity signatures, and wavelet-based signatures are used as features to describe the regions. But none of the three characterization approaches resulted in clustering in feature space consistent with classification objectives. The feature analysis sub-system consists of modules that perform single feature and pattern vector analysis. This sub-system evaluates class representation and determines whether classes can be separated in a feature space. The classification sub-system is composed of classifiers such as non-parametric classifiers. The classification sub-system performs defect recognition. The authors showed that using invariant moments or spatial-domain intensity signatures is not sufficient for classification of defects in a specific web material that has a large amount of variation within a given defect class and overlap between classes in appearance of defects. In hardwood lumber inspection, defects generally have a large variation with a same defect class. Furthermore, defects in different classes can have a similar appearance.

A further literature search and patent search only reveal some similar industrial inspection systems [LON94, FOR97]. Although these state-of-the-art vision systems achieve some reasonable performance in certain restricted environments, most of them are inflexible, special-purpose computer vision systems [CHI86, CHO91, NG93, KRO96, BAT97]. Clearly, none of these existing inspection systems for other industrial inspections can be directly applied to the inspection of hardwood lumber due to enormous variability in the nature and the condition of the wood materials [CON92, KLI96]. However, the conceptual processing and analysis methodology

used in other industrial inspection systems cited in this section can be adapted and employed to help design a machine vision system for hardwood lumber inspection. Methods that look most promising for lumber and wood inspection include: 1) the methods that employ the geometry and topology of image primitives to identify defects [DAR88] where no precise image registrations are required; and 2) the methods that use domain knowledge to guide low-and/or intermediate-level processing of images therefore increasing the speed of level processing.

2.4 Sensor Fusion and Its Applications

The fusion of multiple sensor data is a fundamental function observed in human and other biological systems. Humans can synergistically use the information obtained from the body's multi-sensory inputs (eyes, ears, hands, etc.) with prior knowledge to recognize complicated objects, to make inference, and to handle the events occurring around them. A simple example of human multi-sensor fusion is that when a person walks across a street, his eyes and ears are used to sense the environment and deal with changes in the environment. Humans clearly have the ability to use a flexible body of knowledge and the ability to synergistically fuse the information from different modalities gathered by body senses. A knowledge-based expert system, to some extent, is an attempt to capture the first ability, while the current research in multi-sensor fusion (or integration) is an attempt to emulate aspects of the second ability.

In the past decade, there has been a growing interest in the use of multiple sensors in many areas such as robotics, manufacturing automation, automatic target recognition, pattern recognition systems, active vision systems, and military control [WAL86]. Numerous research papers are published each year in the *IEEE Transactions on Systems, Man and Cybernetics, the International Journal of Robotics Research, Proceedings of SPIE-The International Society for Optical Engineering*, etc.

In industrial inspection applications, multiple sensors can provide information about features that are impossible to gauge with a single sensor. In addition, the use of multiple sensors offers the possibility to both provide more accurate information and increase system reliability in

the case of sensor failure or error. In fact, these benefits are the motivation for using multiple sensors together with sensor fusion technologies.

There is no authoritative definition for the field of sensor fusion yet. In the literature, sensor fusion, data fusion, multi-sensor integration, multi-sensor correlation and data association all generally refer to the same concept. LOU [LOU92] defined multi-sensor fusion as referring "to any stage of actual combination (or fusion) of different sources of sensory information into one representation format." But Lou also argued the distinction of sensor fusion from sensor integration. In some literature, the systematic use of information obtained from multiple sensors to assist in the overall function of a system is also referred to as sensor fusion. Sensor fusion may be carried out in low-level processing or in high-level processing.

Many fusion methods have been proposed by researchers for combining multiple sensory data at different processing levels [PAU83, LOU92]. These methods can generally be classified into two categories [LOU92]: 1) feature-based methods which include weighted average methods, Kalman filtering methods [FAU86, NAK89], statistical pattern-recognition methods [YOU92], Bayesian estimation using consensus-sensors methods [LOU88], multi-Bayesian approaches [DUR87], Dempster-Shafer-based evidential reasoning methods [SHA76, WAL86, BOG87], neural network based methods [JAK88, HAB93], adaptive learning methods [MIL87, CHE88], fuzzy logic [ROG89, ABD93], and the like; and 2) cognitive-based methods which include techniques such as production rule-based methods [KAM85] and other knowledge-based methods [HAN88].

Feature-based fusion may be implemented at the signal level, the pixel level, the feature level, or the symbol level. Signal- and pixel-level fusion is actually direct data fusion in which data (signal, pixel) from a group of sensors are combined to provide the system with the same data but of greater quality. This is common in multi-spectral fusion applications. Pixel-level fusion is especially useful in image processing operations such as segmentation, but it is computationally demanding. Feature-level fusion refers to the fusing of features where each feature is extracted from one type of sensory data. The goal of feature fusion is to provide

additional composite features for the system. Symbol-level fusion allows information (e.g., symbolic descriptions) derived from the individual sensor data to be combined together at a high-level of abstraction. Obviously, fusion at the feature level or symbol level is the choice for this research work because fusing at either of these levels reduces the computational complexity of the fusion task.

At the feature level of sensor fusion, Dempster-Shafer methods have drawn considerable interest from the research community. Garvey, Lowrance, and Fischler [GAR81] introduced the idea of using Dempster-Shafer evidential reasoning in multi-sensor fusion. Bogler [BOG87] and Waltz, et al. [WAL90] have explored employing Dempter-Shafer evidential reasoning to multi-sensor based target identification, and military command and control. Dempster-Shafer evidential reasoning is an extension of the Bayesian approach [ABI92]. Dempster-Shafer evidential reasoning allows each sensor to contribute information on its own terms or more formally at its own level of abstraction or detail. For example, one sensor may be only able to provide information whether an object belongs to a defect class (to a certain degree), and another sensor may be able to confirm (or negate) this decision or identify individual objects with additional information.

The Dempster-Shafer methods can be illustrated by an example from lumber surface defect recognition. Assume that the analysis of the output of a color-imaging sensor (color CCD camera) indicates that a given object (dark, round area) is consistent only with defect types D1 (knot) and D2 (hole). Further, assume the confidence of this indication is 0.9. Based on this finding, a mass of 0.9 is assigned to the disjunction of D1 and D2 (either D1 or D2), i.e., $M_1(D_1 \cup D_2) = 0.9$. Also a mass of 0.1, corresponding to one minus the confidence of 0.9, is assigned to θ, i.e., $M_1(\theta) = 0.1$, where θ represents the disjunction of all propositions, i.e., all possible defect types. Next, assume that a laser-range profile does not show a change in thickness of the object in question. This implies that a knot (or at least not a hole)-type defect is present, but, since other types of defects may have similar shape, the confidence on the hypothesis of a knot defect is only 0.7 (i.e., $M_2(D1) = 0.7$) and the remainder is assigned to θ, $M_2(\theta) = 0.3$.

Dempster's rule then can be used to combine sensor information by effectively determining the intersection of the sensor statements. This is usually accomplished by a probability mass vector. In the above example, taking the product of $M_1(D_1 \cup D_2) = 0.9$ and M_2(D1) = 0.7 results in a mass declaration of 0.63 to the proposition that the defect type is D1(a knot). Similarly, the product of $M_1(\Theta)$ and M_2(D2) results in a declaration of 0.07 to be D2(a hole). The resulting probability mass vector is:

$$\mathbf{M} = \begin{bmatrix} M(D_1) = 0.63 \\ M(D_1 \cup D_2) = 0.27 \\ M(D_2) = 0.07 \\ M(\Theta) = 0.03 \end{bmatrix} \tag{2.3}$$

The mass vector given in Equation (2.3) illustrates the Dempster-Shafer concept of a probability interval, defined by the support and plausibility for a given proposition. The support for a given proposition is that mass assigned directly to that proposition. In the example, the support for the proposition that the defect type is D1 is 0.63. In this way, Dempster's rule combines information from two sensors and shows that this is a strong belief that the object is a knot.

Dempster's rule also conveniently handles the condition of inconsistency between sensor declarations. For example, if the analysis of color camera outputs indicates that an object is a D1 (knot) with a degree of support of 0.7, and, the laser-range profile analysis indicates the object is D2 (hole), with a degree of support 0.9. What degrees of support does the combined evidence provide? Clearly, we have:

M_1 (Knot) = 0.7, $M_1(\Theta)$ = 0.3

M_2 (Hole) = 0.9, $M_2(\Theta)$ = 0.1.

The result product of $M_1 * M_2$ can be computed by:

M(Knot $\cap$ Θ) = 0.07

M(Hole $\cap$ Θ) = 0.27

$M(\Theta \cap \Theta)$ = 0.03

The above result can be normalized as:

$M(\text{Knot} \cap \Theta) = 0.19$

$M(\text{Hole} \cap \Theta) = 0.73$

$M(\Theta \cap \Theta) = 0.08$

After normalizing, the result indicates that there is a strong belief that the defect is a hole. This means that even though the two sensors provide somewhat controversial information, Dempster's rule can still resolve the conflict.

Cognitive-based methods include rule-based and knowledge-based methods. Hanson, et al. [HAN88], described a knowledge-based constraint approach to uniformly combining information from multiple sensors with different representations in their multiple sensor system for scene analysis. Pau [PAU89, PAU90] proposed a formal knowledge representation architecture to fuse the sensory information at the feature and symbol levels. The architecture consists of a hierarchy of processes, with the feature vectors at the lowest level, feature graphs at the next level, and rules invoked by the nodes in feature graphs at the highest level. Fusion can take place at each level in the hierarchy through transformation, match, and projection operations. Examples are given that show the application of the method for the tasks of integrated circuit inspection and object recognition. Kamat [KAM85] used rule-based systems for object recognition using multi-sensor fusion. Production rules, each associated with a confidence factor to indicate a degree of uncertainty, are used to symbolically represent the relations between sensory information and attributes/features that can be inferred from the corresponding information. Fusion takes place when two or more rules, referring to the same object, are combined during logical inference to form one rule. The advantage of using production rules in sensor fusion is that production rules provide modularity for the system since additional sensors can be added to the system without requiring the modification of existing rules. The major problem in using rule-based methods for fusion is that the confidence factor of each rule is dependent upon other rules in the system, making it difficult to alter the system when conditions change.

In general, technologies based on multiple sensor fusion are applied to the applications where no single sensor appears capable of providing the information needed for the applications.

When multiple imaging sensors are combined, both sophisticated image processing techniques and imaging analysis techniques are often necessary; and sensor registration issues also need to be addressed if the spatial resolution of each sensor is different or each sensor has a different viewpoint [LOU92]. The amount of sensory data and the processing time needed also increases as the number of sensors increase, and the processing time is also affected by what processing level the sensory information is fused. Some debate remains [MAR89] as to whether information should be fused at the sensor data level, at the feature level, or at the symbolic level. It seems that the best fusion level and methods are often application dependent. Of all the sensor fusion techniques, Dempster-Shafer's evidential reasoning and fuzzy logic have been successfully used to combine information from multiple sensors in many applications [LOU92, BLO96]. Fuzzy logic will be discussed in more detail in the next section.

2.5 Fuzzy Logic and Its Applications

In the early 1970's, the inventor of fuzzy sets and fuzzy logic, Lotfi Zadeh, stated [ZAD73]:

> "As the complexity of a system increases, our ability to make precise and yet significant statements about its behavior diminishes until a threshold is reached beyond which precision and significance (or relevance) become almost mutually exclusive characteristics."

Real world problems or systems are often complex. Complexity generally stems from uncertainty in the form of ambiguity that is addressed subconsciously by humans. A complete description of a real system is often so complicated that it is not practical to provide such a description with data available. Yet, humans can reason effectively about the real system under some uncertainties. This means that humans have the capacity to reason approximately with only a generic understanding about the system. "Fuzzy set" theory provides a means for representing uncertainties and "fuzzy logic" provides a method to formalize a mechanism for imprecise reasoning.

During the past several decades, researchers have applied the theory of fuzzy sets and fuzzy logic to a variety of areas such as robotics control, processing control, image understanding, information system modeling, and pattern recognition [WAN92, CHE94, LEE94, POP94, KOC96]. These applications are attempts to better model the decision-making process under uncertainty by mimicking human reasoning. .

This section presents a brief review of the relevant concepts of fuzzy mathematics and some applications of fuzzy logic to industrial systems.

2.5.1 Fuzzy Sets and Fuzzy Logic

Fuzzy set theory, introduced by Lotfi Zadeh in the 1960's, is a generalization of crisp (classical) set theory. A fuzzy set consists of objects and their respective grades of membership in the set. The value of the grade of the membership of an object can range from 0 to 1 where the value zero is used to represent non-membership, and the value one is used to represent full membership. Fuzzy logic is a superset of Boolean (conventional) logic that has been extended to handle the concept of partial truth--truth values between "completely true" and "completely false." Fuzzy logic was introduced by Lotfi Zadeh as a means to model the uncertainty of natural language.

Mathematically, a fuzzy set is defined as a set of ordered pairs

$$A = \left\{(x, m_A(x)) \mid m_A(x) \in [0,1], x \in X\right\}$$

Where X is a classical set of objects, x denotes the generic elements of X, and $m_A(x)$ is a membership function which indicates a degree of the membership x in A. The membership function establishes a one-to-one correspondence between an element in the domain and a truth value indicating its degree of membership in the set. Thus,

$m_A(x_1) = 0$ means that x_1 is not a member of A,

$m_A(x_1) = 1$ indicates that x_1 is a member of A,

$m_A(x_1) = 0.7$ denotes that x_1 is a member of A with degree of 0.7.

Clearly, the closer the value of $m_A(x_1)$ is to 1, the more x_1 belongs to A. When $m_A(x_1)$ is restricted to $\{0,1\}$, the fuzzy set simplifies to a crisp (ordinary) set. For an element $x \in X$ that typically belongs to a fuzzy set A, one may demand its membership value to be greater than a threshold $a \in [0,1]$. The ordinary set of such elements is called the a cut of A and usually denoted by A_a:

$$A_a = \{(x, m_A(x)) | m_A(x) \geq a\}$$

Basic set operations on fuzzy sets, which were initially defined by Zadeh are:

Intersection $m_{A \cap B} = \min(m_A[x], m_B[y])$

Union $m_{A \cup B} = \max(m_A[x], m_B[y])$

Complement $m_{\sim A} = 1 - m_A[x]$.

During the past several decades, researchers have defined alternative set operations, and practical experience in constructing fuzzy models shows that such alternative forms are necessary [COX94]. Two commonly applied alternative operators for intersection are mean and weighted-mean operators. For example,

Intersection $m_{A \cap B} = (m_A[x] + m_B[y]) / 2$

Intersection $m_{A \cap B} = (m_A[x] + 2 \cdot m_B[y]) / 3$

A major motivation for fuzzy sets is to provide the need to represent such propositions as:

John is tall.

A knot area is round.

Both propositions involve fuzziness and are fuzzy propositions. Interpreting the first proposition depends on defining the degree of tallness. Six feet two inches can be considered tall. Six feet can also be considered tall but to a lesser degree. The second proposition involves the degree of roundness of a knot. With fuzzy set theory, *tall* and *round* (linguistic values) can be characterized numerically using membership functions.

Fuzzy logic is an extension of classical logic when partial truths are included to extend two-value or binary logic (true or false) to a multi-valued logic (degrees of truth between true and false). A proposition in binary logic is assigned to an ordinary set [0, 1], where 0 typically represents false and 1 represents true. While fuzzy propositions are assigned to fuzzy sets, partial truth is represented by the membership degrees of fuzzy sets. Therefore, fuzzy propositions have the ability to express in numerical format the amount of imprecision and ambiguity in human thinking and decision-making. The goal of fuzzy logic is to model the imprecise modes of reasoning that play an important role in the remarkable human ability to make rational decisions with inexact, incomplete, or not totally reliable knowledge.

Literature in applications of fuzzy sets is vast. The applications of fuzzy sets and fuzzy logic could be found within the range from process control, image recognition, to medical diagnosis and expert systems. Many algorithms and methodologies have been developed during the past decades for fuzzy process control and pattern recognition. But in the next sub-section, only some applications of fuzzy sets related to pattern recognition will be reviewed. The algorithms and methodologies that can be utilized for this research will be discussed in later chapters.

2.5.2 Applications of Fuzzy Sets and Fuzzy Logic

Luria, et al., [LUR94] described a fuzzy logic expert system for automatic defect classification of semiconductor wafer defects. The defect identification includes five parts: 1) Image extraction--a defect image is extracted using image comparison techniques (the image is compared to a reference image). 2) Feature extraction--both spatial and textural visual features are extracted from the defect image, using image-processing techniques. 3) Fuzzification of visual predicates--visual predicates are fuzzy terms which describe the characteristics of the wafer defect. Circular, elongated, and scattered are examples of visual predicates. The truth value of each visual predicate is then determined from the features extracted previously, using trapezoidal numbers and S-functions. 4) Rule application--each of the rules, which is designed to identify one of the defect classes, is evaluated using the visual predicates computed earlier and a truth value is

returned. 5) Defect classification decision--the rule with the highest truth value is selected and hence the corresponding defect is identified. It is not clear how this system handles the situation when two rules return similar truth values. The classification accuracy was under test.

Popovic and Liang [POP94] proposed a fuzzy logic approach to pattern recognition along with the corresponding algorithm for model-based object recognition in intelligent robotics with a limited number of well-defined objects. In this approach, all pattern classes of the objects to be expected in the "world" are first stored in their image and their features called AOS (angle-of-sight) signature form. Then Popovic and Liang defined the grade membership functions to represent the membership within the fuzzy sets mapped as referent image clusters. The larger the value of the membership function, the closer the object pattern is to the corresponding fuzzy set (referent image). Therefore, an object image is finally classified as a member of a fuzzy set (corresponding to a pre-defined pattern class) at which maximum value of the membership function is assumed. If there is more than one maximum value, then the object is classified according to the priorities of these fuzzy sets. This is accomplished by adding knowledge-based rules to the algorithm. The simulation results showed that the fuzzy approach is efficient if the number of pattern classes is small and the shapes of pattern are relatively simple. The authors further suggested that the approach could be improved by better feature selections. It seems that this approach could possibly be adapted to defect detection in hardwood lumber processing since the defects to be expected can be categorized into pre-defined classes. Moreover, the defect classes could be described by their features although these features are much more complicated than the objects in the robot world.

Although fuzzy sets and logic appear promising for hardware lumber inspection utilizing multi-sensor information, little work has been done to test its applicability. With the very heterogeneous nature of hardwood and many variations with the same defect class, some rather vague but robust object identification criteria would be necessary to detect the defects in the hardwood lumber. Moreover, fuzzy logic approach has generally proved to be highly useful pattern recognition [ZIM87] since the fuzzy sets have become the adequate theoretical background for modeling the cognitive processes of humans related to object recognition.

Therefore, one should be able to use the fuzzy sets to represent the features of each defect class and to use related fuzzy logic approaches to identify the defects in hardwood lumber.

2.6 Summary of Review

The discussion in this chapter has shown that a current computer vision system for industrial inspection is composed of three levels of processing: low-level segmentation, intermediate-level feature extraction and high-level recognition. Many inspection systems have been proposed or developed by researchers during the past decade to address automatic inspection of wood products. Most of these systems utilize a single-sensing modality to detect either surface defects or interior defects. Although much progress has been made in recent years, commercial systems have not been able to completely meet industry's expectations. Yet, to date, published information showed that there is no proposed or developed vision system that is capable of detecting all features in hardwood lumber. Researchers have identified the need for multiple-sensor machine vision systems that can accurately identify all important lumber features in real-time.

The vision systems designed and developed for other industrial inspections have been used to inspect products such as printed circuit boards and integrated circuit wafers. But these systems are very application-specific and hence inflexible in their structure. These vision systems cannot be easily tailored to inspect hardwood lumber. Although substantial research progress has been made and the speed of computer hardware has been increasing rapidly, creating a computer vision system that can address a fairly general class of problems has proven to be a very difficult task for the computer vision research community.

The techniques used in general sensor fusion and fuzzy logic were described. These techniques are useful in creating a vision system that can detect all the features in hardwood lumber where multiple sensors have to be used to unambiguously differentiate between the various different feature types. Given the heterogeneous nature of hardwood lumber, a rather "sophisticated" knowledge base is also needed in a computer vision system for hardwood lumber

inspection. The objective of this research is to design a multiple-sensor vision system that detects all features in hardwood lumber.

Chapter 3

Overview of the Proposed Multiple Sensor Vision System

In Chapter 2, various computer vision systems for wood inspection were reviewed. The review of the literature reveals that present commercial systems cannot yet detect all critical features in hardwood lumber and optimize yield in the rough mill. Current state-of-the-art commercial systems for hardwood inspection still rely on operators to inspect lumber for critical defects [KIL96, FOR96]. In industry, a completely automatic lumber inspection system must not only accurately detect the size and shape of lumber, but also accurately find the location and type of defect with sufficient precision and resolution. Moreover, the inspection system must be fast enough to match the speed of the production line. Of course, the system must be realizable at a reasonable cost. To more fully automate lumber inspection, most of the existing proposed systems are based on a single-sensing modality, which limits the abilities and performance of these systems on detecting wood features. Ultimately, multiple-sensor scanning systems will be required to reach expected levels of accuracy and precision.

To effectively process the wealth of information that can be generated by these systems, a novel machine vision system approach is proposed. This approach must be able to address the variability and inconsistency that can be present from multiple sources of data. Furthermore, the approach must be robust enough to detect all features (defects) in hardwood lumber and efficient enough to be implemented in real-time. This chapter provides an overview of the proposed machine vision system and a comparison of the proposed system with other existing inspection systems.

3.1 Overview of the Lumber Inspection System Hardware

An overall lumber inspection hardware system has been developed at Virginia Tech to generate combined color, laser-profile, and X-ray images for one side of a board (see Figure 3.1). While the imaging hardware can be individually set to test different resolutions, this research generated all images with a cross-board resolution of 1.2 pixel/mm (30 pixel/inch) and a down-board resolution of 0.63 pixel/mm (16 pixel/inch). This resolution was selected because it was the highest resolution that the X-ray imaging system could achieve on a target moving at the speed of 0.61 m/s (2 linear ft/s). The laser profile, color, and X-ray imaging subsystems were calibrated so that each generated image had identical spatial resolution. The lumber inspection hardware consists of 1) two high-speed laser ranging systems, one for the top side, one for the bottom side; 2) two color-line scan imaging systems, one for each side; 3) an X-ray imaging system; 4) a processing computer with special image processing hardware; and 5) a material handling control system. In this research, only one high-speed laser ranging system and one color imaging system is used to scan one side of the lumber at a time. Hardware systems depicted in the dotted line in Figure 3.1 are not used.

3.1.1 Laser Profiling System

A high-speed laser-based ranging system, designed and built at Virginia Tech [HOU93], uses a 16-mW helium-neon gas laser with a 632.8 nm wavelength. A 24-facet polygon scan mirror, rotating at about 30,000 rpm, sweeps a point of laser light across the board. The image is captured by two EG&G black-and-white 128 x 128 array cameras at the rate of 384 frames per second. At 0.61 m/s (2 linear ft/s), the cameras were set with a cross-board resolution of 1.2 pixel/mm (30 pixel/inch) and a down-board resolution of 0.63 pixel/mm (16 pixel/inch). Because the mirror is rotating at such a high rate, the point of laser light sweeps across the board surface several times in one video frame. As such, the laser appears as a line falling across the board.

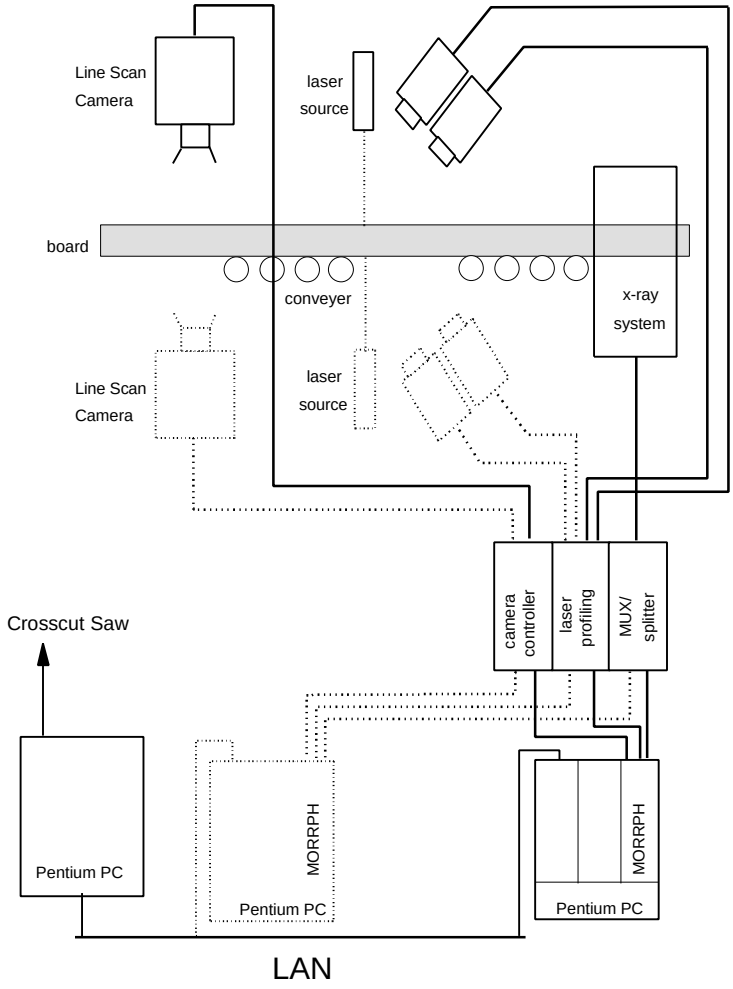


Figure 3.1 Overview of the hardware configuration for the proposed machine vision system

The cameras then capture the displacement of the laser line and variations in its light intensity. This data, as an analog signal, is transferred to a special-purpose electronics board. The board converts the data to an 8-bit digital image, determines the position of the laser line in each row of data, and then enters that pixel location into a lookup table. The table generates 7 bits of range data, which are transferred to a computer interface on the image-processing computer. Through this process, the laser-based ranging system can measure board thickness and various board indentations to within 0.4 mm (1/64 inch).

3.1.2 Color Imaging System

Color images were collected using a Pulnix TL-2600 RGB line scan camera with a resolution of 864 pixels and a data collection rate of 2.5 MHz. With a processing speed of 0.61 m/s (2 linear ft/s), the color imaging system is configured with a cross-board resolution of 2.4 pixel/mm (60 pixel/inch) and a down-board resolution of 1.26 pixel/mm (32 pixel/inch), double that of the final system resolution. This higher resolution is necessary to detect finer resolution details in the wood such as cracks and splits. With a crack-preserving filter applied to the image data, the resolution can be later cut in half, allowing the data resolution to match that of the laser-ranging and X-ray sensing devices while preserving crack and split information.

The camera was mounted perpendicular to the wood surface and Tungsten-halogen incandescent light bulbs were used to illuminate the board surface. The light was carried to the board surface and background using linear Fostec fiber-optic light lines. The Pulnix camera has three sensor arrays filtered with red, green and blue interference filters respectively. This camera configuration gives a slight spatial offset of the three color images, but no undesired effects were observed for this experiment. The camera was fitted with two color-balancing filters (Schott filter numbers FG-6 and BG-34).

3.1.3 X-ray Imaging System

Scanning with X-rays gives an averaged image of the wood density throughout the lumber thickness. This average wood density can be useful for detection of wood features that have a higher or lower density than normal wood such as knots, voids, and decay. The X-ray system employed an EG&G Astrophysics X-ray source with the radiation energy set to 100 kV and 0.6 mA. The X-ray sensor was a 256-pixel line array manufactured by FISCAN. As mentioned earlier, the X-ray imaging system is the limiting sensor in terms of resolution that can be collected. With image geometry constraints and processing speeds of 0.61 m/s (2 linear ft/s), the maximum cross-board and down-board resolution that can be achieved with the X-ray imaging system is 1.2 pixel/mm (30 pixel/inch) and 0.63 pixel/mm (16 pixel/inch), respectively.

3.1.4 Image Processing Hardware

The image-processing computer uses a 200-MHz Pentium PC with 64 Mbytes of main memory running Windows NT. The computer incorporates the use of special purpose hardware to perform various low-level image processing functions and DMA transfer of the collected images into computer memory. This special purpose hardware was designed to handle the large amount of image data in the collected images. For example, the initial color image data alone from one side of a 4.9 m long, 34.3 cm wide (16-foot, 13-1/2-inch) board alone will require almost 16 Mbytes of computer memory. Furthermore, low-level functions require more than 70 percent of the computational time [CHO92]. Therefore, the performance of the system can be significantly improved by running them on special purpose hardware.

Special-purpose hardware employed in this research uses the MORRPH-ISA board [DRA96, DRA95]. This special image processing hardware--the MOdular Re-programmable Real-time Processing Hardware (MORRPH) is a general-purpose reconfigurable unit, designed primarily for real-time low-level image processing. The MORRPH is implemented in reconfigurable hardware employing FPGA chips. The image preprocessing functions that were developed in this research (see Section 3.2.1) were designed to be implemented on the MORRPH board. The output from these low-level functions is then passed to computer memory using a high-performance direct memory access interface board for the PCI bus [DRA95]. This DMA interface board transfers data from the MORRPH into system memory without using the system processor. Through this hardware configuration, real-time processing of large amounts of machine vision data is possible.

3.1.5 Material Handling Control System

Boards need to pass through the various imaging systems at a constant rate, without bouncing vertically or shifting laterally. This system uses drive rollers beneath the board to move the board, and pneumatic rollers above the board to keep it in place. A 286 PC controls the entire

prototype system, sending control signals to the materials handling system and the image-processing system computer. As lumber moves from left to right, it triggers the control circuits that control the image collection of the three imaging systems. The laser range, color, and X-ray images are collected simultaneously as the lumber travels through the conveyor. The collected images are first preprocessed by the special MORRPH hardware. The preprocessing hardware performs low-level image processing operations, including shading correction, background extraction, histogram extraction, and crack-preserving filtering as the images are being collected. The preprocessed images are then transferred to the memory of the processing computer for further analysis. When the trailing end of the board leaves the system, the material handling control signals the processing computer to cease collection of images.

3.2 Software Architecture

The proposed machine vision system developed in this research is responsible for processing these three images (laser range, color, and X-ray) to locate and identify defects that can be seen from the scanned lumber face. The final output of the machine vision system is a "defect map" that includes the size, location, and type of every defect. The machine vision software developed in this research uses a novel data-fusion approach to first preprocess the images, segment the image into regions of interest, and then to employ fuzzy logic to determine to which defect class the various regions belong. The machine vision software architecture consists of four major modules: 1) image preprocessing module, 2) laser-image analyzing module, 3) X-ray image processing module, and 4) color-image analyzing module as illustrated in Figure 3.2. The preprocessing module performs image pre-processing and image registration and then outputs the laser, X-ray, and color images to the other modules. The module for laser image analyzing first processes the laser image and passes its results to the other image analyzing modules. Subsequently, the X-ray module analyzes the X-ray image and transfers its results to the color module. Finally, the color module performs image analyzing on the color image and forms the final results.

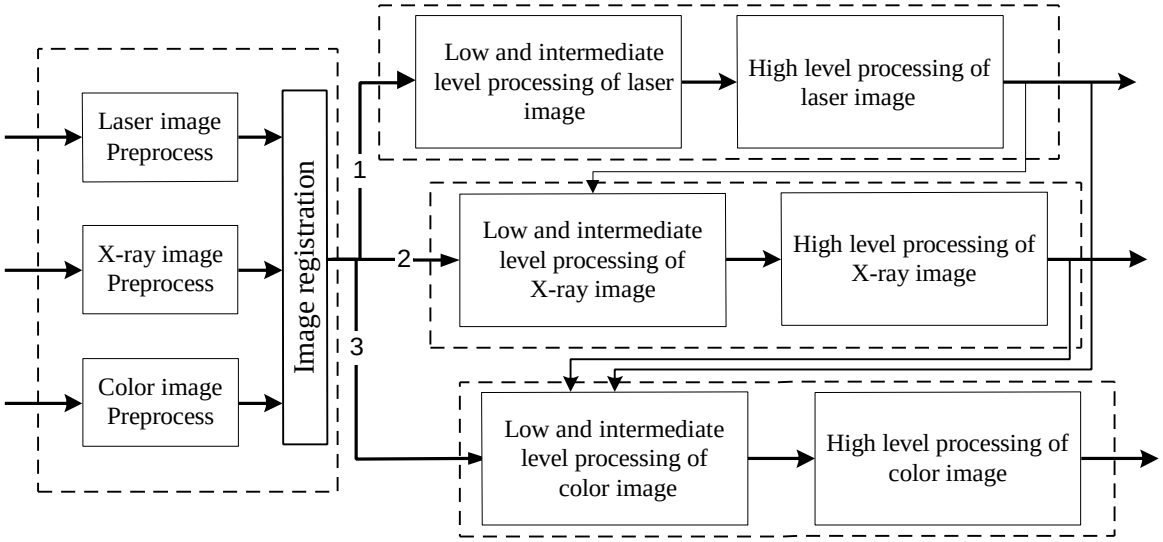


Figure 3.2 Software modules of the machine vision system

3.2.1 Image Preprocessing Module

The purpose of image preprocessing is to provide higher quality (for example, uniformly illuminated) images to the computer vision system and reduce the amount of image data required to detect critical defects during the image analysis in the vision system. The image preprocessing used in this research includes the following operations: shading correction, background extraction, histogram extraction, a crack-preserving filtering and image registration. Shading correction is applied only to color images (shading correction for laser/X-ray images is performed by image collection hardware) as detailed in Section 6.1. Background extraction is used to detect the lumber edge in each of the three images. Histogram extraction operations are applied to both color and X-ray images to extract histograms for further analyses such as image thresholding. A pre-determined threshold can be used (see Section 6.2 for details) to threshold the laser images; therefore, histogram extraction is not performed on laser images.

The "crack-preserving" filter is an image filter that reduces the amount of image data fed into the vision system, and in the meantime preserves the system's ability to detect small cracks and other fine defects. A sample 1 x 2 crack-preserving filter is applied to disjoint 1 x 2 sub-arrays of the color image, sub-arrays that completely cover the color image. The filter finds the color pixel in each sub-array that has the darkest color. This pixel is then chosen to represent the 1 x 2 sub-array. In this way, the filter reduces the number of pixels that are sent to the vision system by a factor of 2, while preserving the pixel level of details needed to detect small cracks that usually are very dark in a color image. The crack-preserving filter will be discussed in detail in Section 4.1. The software coding of this filter was developed so that it could be easily implemented in the MORRPH hardware.

As with most multiple sensor systems, the problem of image registration must be addressed before image analysis. Image registration generally refers to any of the methods (e.g., geometrical transformations) used to make the images from each sensor commensurate in both its spatial dimensions. That is, for instance, pixel $C_{i,j}$ in color image, pixel $L_{i,j}$ in the laser-range image and pixel $X_{i,j}$ in the X-ray image refer to the same location (i, j) on the lumber surface. In the proposed vision system, the image edges are used as the corresponding reference objects for image registration. Since the majority of lumber to be inspected has a long and straight edge and the lumber is placed tightly against a fence on the material handling system, each of the three images collected has one common straight edge. Hence, the registration is simplified to image shift operations if all images have the same resolution. Detailed discussion on image registration used in this research can be found in Section 4.1.

After all three images are pre-processed and registered, they are fed into three image analyzing modules. Subsequently, the laser image is analyzed first.

3.2.2 Laser Image Processing Module

The laser-range profiling image is mainly intended to detect defects such as wane and thin regions in the board. The processing of a laser image consists of three major operations: low-

level segmentation, intermediate-level extraction of component measures, and high-level defect identification.

The laser image provides an unambiguous measure concerning the thickness variation of the lumber to be inspected, and each pixel value represents a relative thickness of the corresponding area on the lumber surface with a thickness resolution of 0.4mm (1/64 inch). A pixel of low value generally represents a corresponding thinner area on the lumber surface. Defect areas that are thinner than normal lumber such as wane usually manifest themselves in laser images. These areas generally have lower pixel values compared to normal lumber in the laser image. However, due to some hardware limitations of the laser profiling system, it cannot resolve thickness for very dark regions such as dark surface stains, dark knots, deep holes, and it cannot detect *very* thin areas either. Pixels in these defect areas have a very low value determined by the hardware. The images of these defects are classified as "*unknown*" areas since what is inside these areas cannot be determined from the laser-range image alone.

Based on the properties of the laser-range image, one could use a threshold to separate pixels in "*unknown*" areas from pixels in thin areas and another threshold to separate pixels in thin areas from pixels that represent normal lumber during operation of laser-image segmentation. Obviously, a multiple thresholding operation is then needed for the laser-range image after the above two thresholds are selected. The operation separates pixels which possibly represent defects. Subsequently, a connected-component labeling operation is applied to the image to form initial regions that contain possible defects. A 4-neighbor connected-component labeling algorithm has been developed in this research to extract components from the multiple thresholded images. Details of laser-image segmentation are discussed in Section 6.2. The low-level segmentation will finally result in one or more of the following three initial components: components of normal thickness, thin/wane components, and unknown components.

The extraction of component measures is then performed on the initial components found earlier. But these initial components usually contain many small disjointed components (e.g., 1 or 2 pixel components) that are considered to be noise and, consequently, are first eliminated. Basic

component measures are then extracted from each of the segmented initial components. These measures, as discussed in Section 6.3, include area, minimum inscribing rectangle (MIR), average range value, and height/width ratio. Similar nearby adjacent components (e.g., within 1 or 2 pixels) are then merged when certain component measures match. Once similar components are merged, additional component measures that are numerically intensive such as compactness are subsequently extracted. The output of intermediate-level processing includes a list of components that contain possible defects and a set of component measures for each component on the list.

Two defect identification expert procedures are applied to each component found by the low- and intermediate-level laser image-processing module. One procedure is designed to identify the wane defect, the other procedure is used to identify areas that are too thin. These laser-image defect identification procedures will be described in greater detail in Chapter 6. Knowledge-based approaches are used in each procedure to determine the type of a given defect area. Any component that cannot be identified by one of the two procedures will be marked as an *unknown* component and more information will need to be collected from the X-ray and color-image data before they can be identified. Defect components that can be identified are added to a final defect list. Unknown components are put into a tentative defect list for further consideration in the analysis that follows.

3.2.3 X-ray Image Processing Module

The X-ray image is mainly used to detect defects such as knots, holes, mineral streaks, large splits/checks, low-density areas such as internal voids or decay, and high-density areas such foreign metallic objects, based on the density variation (measured by X-ray transmission value) of these defects. Basically, the X-ray image provides a density map of the lumber. Some defects such as knots and mineral streak have a higher density compared to normal clear wood, while some other defects such as holes, cracks/splits, internal voids and decay have a lower density.

The processing of an X-ray image consists of two processing stages: 1) initial/large-defect area detection and 2) secondary/small-defect area detection. The purpose of these two processing

stages is to remove the effect of the large-defect areas from the histogram of the X-ray image so that the sensitivity to finding smaller defect areas is enhanced. Each of these two processing stages consists of three operations: low-level segmentation, intermediate-level component measure extraction, and high-level defect identification.

Before the low-level segmentation of the X-ray image, pixels within the minimum inscribing rectangle of each defect area found earlier in the laser-image processing are removed from the histogram of the X-ray image as described in Section 7.1. This reduces processing time and increases the system's ability to detect defects.

A typical histogram of an X-ray image has one large peak that represents the clear wood area. The histogram may also have one or two smaller peaks, one to the right of the large peak for areas that have a higher density, the other one to the left of the large peak for areas that have a lower density (see Figure 3.3). Both valley points and inflection points may exist between two

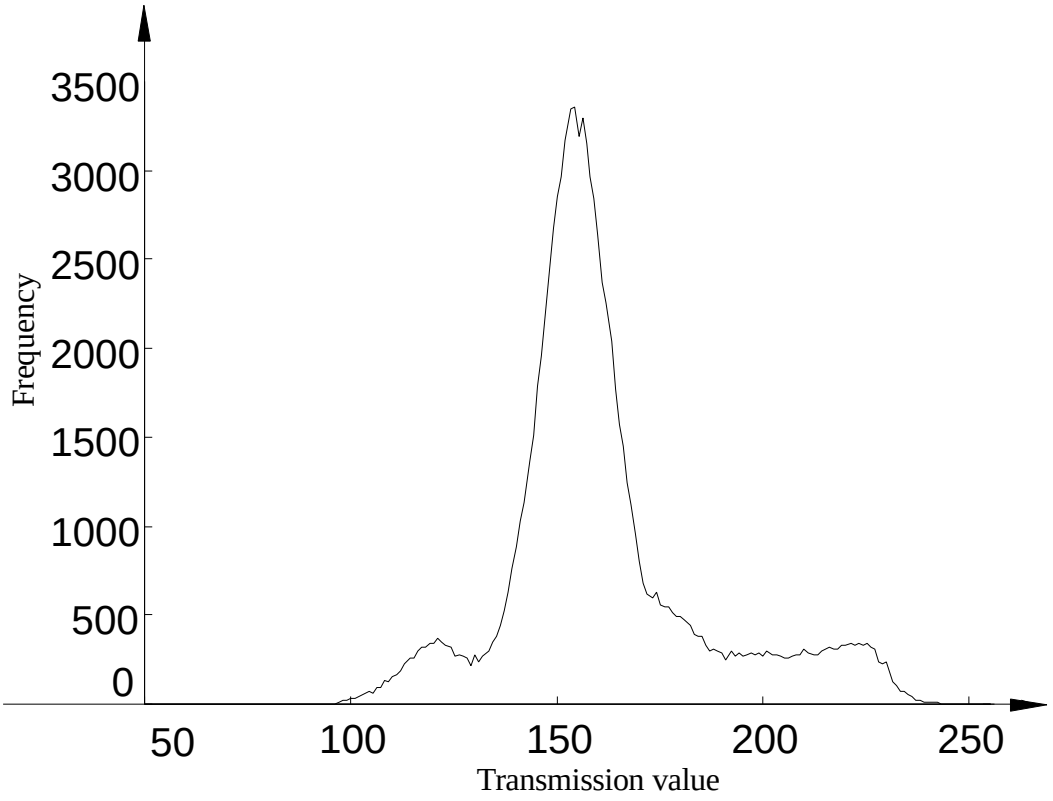


Figure 3.3 A typical histogram of X-ray images

adjacent peaks since the pixels in the clear wood areas, high-density defect areas, and low-density defect areas are approximately normally distributed. These valley or inflection points can be used to find the threshold levels to separate pixels representing areas of clear wood with normal density from pixels that might be in defect areas. Thresholding separates image pixels into three pixel area classes: 1) higher density-, 2) normal density-, and 3) lower density-areas respectively. The operation of connected-component region labeling (as discussed in Section 6.2.2) is then applied on the initially segmented image to form regions of connected pixels belonging to the same density class. This ends the low-level segmentation and results in initial components that contain possible defects. Details of initial large-area segmentation for X-ray images can be found in Section 7.2.

At the intermediate-level processing, basic component measures such as area, MIR, and average gray values are then extracted, and small noisy components are eliminated. Similar adjacent components are then merged, based on these basic component measures that are relatively simple to calculate. The component-merging algorithm is derived from a rule-based scheme [HAN88]. Once the operation of component merging is finished, additional component measures that are numerically intensive such as compactness are subsequently extracted. Components whose areas are greater than a certain threshold (as described in Section 7.3) are forwarded to the high-level processing stage. Since small components that contain only several pixels are considered to be noise, they are discarded at the intermediate stage of analysis. Details of intermediate-level processing of X-ray image are given in Section 7.3.

At the high-level processing stage, each component found by the intermediate stage is then processed with defect identification modules (e.g., knots, mineral streaks, holes, splits, and low-density region). Each of these modules utilizes a fuzzy logic approach called weighted-approaching degree method [ROS95] to determine the similarity between a specific defect class and the given component, based on a number of component measures. Each module outputs a fuzzy similarity measure-approaching degree [ROS95], gauging how close a component resembles a specific defect type. Once the component is analyzed for every defect type, the maximum value of approaching degree is determined, and the corresponding defect class label is

assigned to the component since the component most closely resembles the defect class. The detailed descriptions of the algorithms for identifying each type of defects are discussed in Section 7.4.

The fuzzy logic approach is considered more appropriate here than the traditional Bayesian-based approach for one major reason. The *a-priori* probability of a defect occurring within a specific board is not required in the defect identification process. As a matter of fact, this *a-priori* probability is rather difficult to determine even with a large number of lumber samples available. For example, suppose there are 500 board samples and the following defects are observed: 1400 knots, 400 holes and 200 wane areas, a total of 2000 defects. Therefore, the *a-priori* probabilities for defect knot, hole, and split to occur are approximately: Pr(Knot) = 0.7, Pr(Hole) = 0.2 and Pr(Wane) = 0.1. However, for a specific board, the above *a-priori* probabilities often don't hold since most defects occur at different patterns. For example, if wane areas appear in the board, often there exist a number of wane areas, and as a consequence, there are more wane areas than knots. Similarly, holes sometimes also appear in clusters (many holes can appear in one area) in some boards, but do not occur in most of the samples. Hence for a given board, the *a-priori* probability of a specific defect type occurring is difficult to determine, which makes Bayes's approach an inappropriate method for the task of defect identification.

To detect some defects such as low-density or high-density areas, information (additional measures of the components) from other sensing modalities (color imaging and laser-range imaging) is often needed. The information provided by the other two imaging sensors is extracted only when it is necessary and is subsequently fused by defect identification modules. For example, both a high-density area and a knot appear as a dark area (corresponding to high density areas) in an X-ray image. Both defects could have a similar shape, too. Therefore, a high density area in an X-ray image cannot often be determined from the X-ray image alone, since it can be a knot or a high-density area. However, the color image can provide surface color in the corresponding area, which can be used to discriminate between a knot and a high-density area. Given the location of the area in the X-ray image, the color information in the area can be quickly and easily extracted from the color image. One can then combine the color feature with other features extracted in the

X-ray image and determine the defect type. This approach substantially reduces the time needed to fuse the information from all three imaging modalities for defect detection compared to traditional sensor fusion methods.

To detect small area defects, a new segmentation approach is implemented. In this approach, big defect regions are first identified as discussed above and then removed from the histogram of the X-ray image as described in Section 7.4. The major reason for removing large-area defects such as splits, big holes, mineral streak and knots from the histogram of the X-ray image is to obtain a better-shaped histogram that is more sensitive to smaller defects. Since in a histogram of an X-ray image, the areas such as interior splits and small holes may fall into the same interval as those big ones. As such, the algorithm that determines thresholds cannot easily detect the inflection points in the histogram caused by small knots or decay regions.

Once pixels inside these identified large defect areas are removed from the histogram, the same algorithms in the three stages of X-ray image processing are employed again to detect small defects. Defects that are detected and identified at this stage are added to the final defect list.

3.2.4 Color Image Processing Module

The color image is essential in accurately identifying many defects such as knots, mineral streaks, splits and surface stains. Before the color image is processed, most large defects such as large knots, large holes, wane and mineral streak have often been identified in the previous analysis. Therefore, the module of color-image processing is used primarily to detect surface defects such as surface stains and small defects such as small knots and splits. However, the module is capable of detecting any defect with additional information obtained from laser and X-ray images. As with the X-ray image processing module, the processing of the color image is also composed of three levels of processing: low-level segmentation, intermediate-level region feature extraction, and high-level defect identification as illustrated in the Figure 3.2.

Before the low-level segmentation of the color image, defects found previously are removed from the color image. This is accomplished by removing pixels within the minimum inscribing rectangle of each defect area from the histogram of the color image as described in Section 8.1. This reduces processing time and increases the system's ability to detect smaller defects.

Low-level vision algorithms that are applied to the segmentation of the color image are similar to those in X-ray image processing. To reduce computational complexity, thresholding and segmentation are performed only on a black and white image that is obtained from the intensity of the color image. The intensity, I, of a pixel is calculated from a color image pixel as I = (R + G + B)/3, where R, G, and B are the color image pixel values of red, green, and blue components, respectively. Ohta [OHA80] showed that (R + G + B)/3 is the most significant feature that one can use in color-image segmentation. Full color information will be used in the extraction of component measures during the intermediate-level processing of the color image.

Pixels of color intensity, I, in clear wood are often normally distributed and constitute the majority of the pixels in an image of lumber to be inspected. Pixels in most surface defect areas are also normally distributed and they generally have either a darker color or a brighter color than clear wood. Both valley points and inflection points may exist between two adjacent peaks between each of these distributions. Hence, these valley or inflection points can be used to locate a threshold to separate pixels representing areas of clear wood from pixels that might be defect areas. Multiple thresholding techniques [CHO91] have been used to initially segment images that may have several kinds of pixels that represent areas of different intensity levels. Once again, the operation of connected-component labeling is applied on the initially segmented image to form components. These components are then forwarded to the intermediate-level processing stage.

In intermediate-level processing, small components are first eliminated and adjacent similar components are merged. Initial components are outputs of the connected-component labeling. As with the laser and X-ray processing modules, there often exist many small components caused by noise and over-fragmentation of the image. The components merging algorithm is also derived

from a rule-based scheme [HAN88]. All the rules are built from some basic measures of components such as intensity level and color components. Dempster-Shafer reasoning is applied to evaluation and combination of rules [SHA76, BOG87]. Derivation of appropriate rules is detailed in Section 7.3. The intermediate-level processing then passes a number of components together with their components measures to high-level processing.

At high-level processing, defect identification expert modules for various surface defects such as stains, knots, holes, splits/checks, and mineral streak are applied to each component founded by the intermediate-level processing module. Similar to the processing of the X-ray image, fuzzy logic approaches are used in each module to determine the similarity (approaching degree) between a defect class and a given component. After the component is analyzed with every defect identification module, the maximum value of the approaching degree is determined and the corresponding defect class label is assigned to the component. A detailed description of this module is discussed in Chapter 7.

3.3 How the Proposed System Differs From Other Systems

Although the design of this vision was influenced, to some extent, by the vision system proposed by Cho [CHO91] for industrial web inspection and the vision system described by [HAN88], the software architecture of the proposed vision system differs significantly from other existing and proposed vision systems [ANU89, CHO91, ANU91, MAT93] and commercial vision systems [BAL95] in a number of ways. In this section the proposed vision system is compared with these existing vision systems in various aspects. The purpose of the comparison is to show the differences and the similarities among these vision systems and the motivation as to why the approach developed in this research is appropriate for the automatic lumber inspection problem.

The proposed vision system is designed to detect surface, shape and internal features of hardwood lumber. A multiple sensor approach utilizes color images for surface feature detection, laser-range images for shape detection, and X-ray images for internal feature detection problems. While this vision system was influenced to some extent by Cho [CHO91], Cho's vision system

was designed to address only the surface feature detection problem based only on black/white images. The requirements and analysis methods required for internal and three-dimensional shape feature detection are, in general, significantly different from these for the surface feature detection. Moreover, the proposed system potentially needs to process as much as three times the amount of information processed by Cho's vision system within the same time interval. Hence, the proposed vision system incorporates a new methodology for sensor data fusion to provide the information necessary for feature detection and to reduce the processing time which is a critical requirement for both accuracy and processing speed.

Although some similarities exist, the strategies used for surface feature detection in the proposed vision system are also different from those used in Cho's system. For lower-level processing, both these systems utilize the same multi-thresholding method based on one global gray-level histogram to minimize the processing time needed. To extract components that contain possible defect features, Cho used the connected-component region labeling method described in [PAV77]. It is not clear how the method was applied to handle thresholded images since the method was designed for binary images [PAV77]. The proposed vision system incorporates a new connected-component region-labeling algorithm to process multiple valued images. This method will be discussed in the next chapter. Both systems extract component measures and pass them on for high-level processing in a similar manner.

At the high-level processing of color imagery data, Cho proposed to use a component's confidence vector to assign a tentative defect label for a given component. The i^{th} element of the confidence vector is the confidence that the component is a defect of type i. A component's confidence vector is computed based on the component's measures. A component is assigned a tentative label according to the largest element of the confidence vector. Then Cho used a rule-based approach, k-nearest-neighbor approach, or neural network approach to verify the label assigned earlier [CHO91]. For those components whose defect label cannot be determined, context information is used to assign labels for those components. Finally, spatial constraints are used to verify the labeling of each component, and multiple labels are resolved, but no details are

given in [CHO91]. The neural network approach was selected by Cho to verify component labels since the approach outperforms the other two approaches.

In the proposed vision system, a fuzzy logic-based approach is used that utilizes information from multiple sensors to identify each of the components found at the lower- and intermediate-level processing modules. The approach incorporates the weighted approaching-degree method to fuse the information from each sensor when necessary and determines a number of similarity measures (approaching degree) between a given component and each defect class. The component is assigned a class label according to the maximum approaching degree. The value of approaching degree is derived from a number of component measures represented by fuzzy sets and a group of weight factors. Fuzzy membership functions are derived from the distributions of component measures combined with knowledge of human experts in the domain area (hardwood lumber). The advantages of the fuzzy logic-based approach over the neural network approach are: 1) The knowledge of human experts can be directly built into the rules; hence, a large number of defect examples may not be necessary to train the vision system. 2) By adjusting corresponding parameters, the system should be able to handle changes in lumber size, lumber species, etc., with little retraining. The advantage of the fuzzy logic-based approach over a pure rule-based approach is that a relatively small number of rules are needed.

The proposed inspection system and the two systems described by Aune and Peter [ANU89, AUN91] all use multiple sensing modalities, and are designed to work on a similar problem--detecting defects in wood. But the two systems were primarily designed to detect defects in logs, not for surface features on lumber such as surface stains that are considered to be defects in some applications. The requirements for surface-defect detection are often quite different from log-defect detection. Moreover, in the systems proposed by Aune and Peter, defects such as knots and rot are determined by sudden changes of density, while in the proposed system defect, areas are analyzed by segmentation followed by recognition. A region segmentation based on the histogram dynamically partitions areas that may contain defects in an image. Then, a set of fuzzy logic rules is applied to analyze and determine if an area is a defect. Hence the proposed system is different and unique both in architecture and analysis methods.

Floyd, et al.[FLO94], described a multiple sensor method for determination of pith location relative to lumber surfaces in U.S. Patent No. 4,831,545. The method uses slope-of-grain detection sensors described in [MAT86, MAT93] and an additional optical sensor to determine the presence of wane. Both faces of the lumber are scanned to determine the presence and location of wane. Then both faces are further scanned to determine the number and size of knots on each face. Apparently, the method cannot be applied to detection of internal defects in lumber. Also, the method cannot detect surface discoloration. This limits the application of the method to hardwood lumber inspection.

Blackman [BAL95] described a vision system in a commercial ripsaw optimizer called RipScan. The vision system is designed to detect only surface features such as wane, grain, knots, and splits in pine lumber, with a single sensing modality--color cameras. The author mentioned that this vision system utilizes a "massive parallel neural net" to identify features in wood. No more details are given in [BAL95]. However, it takes the system's Pentium-powered computers close to 40 seconds to analyze the board and find the best cutting solutions before the board is ripped. Real-time operation of this vision system is in question.

3.4 Summary

A novel computer vision system for hardwood lumber inspection is described at high-level and compared with some existing systems. The system has several unique features compared with existing systems.

The system first filters and registers all three images in a special way, then processes images in a unique order: laser image, X-ray image and color image and the system subsequently detects primarily one or more defect classes at one time. After the laser image is analyzed, the system removes the defect areas found previously from consideration and extracts necessary information when it is needed to determine a defect class of a component in question. The information is extracted only from the corresponding regions in the other two images. In such a

way, the system avoids computational intensive operations in traditional data fusion, removes redundant data and improves the accuracy of defect recognition and therefore markedly reduces processing time.

The design of the system takes into account the natural variability of the lumber material that no other previous research has claimed. The system looks at the entire lumber surface or a large region of that lumber surface and makes dynamic inferences as to the features present on that region. The system handles the variability of the lumber material by a unique way of representing "fuzzy" concepts such as "redder than," "denser than," "darker than," "rounder than," etc. This representation is more dynamic than traditional systems; it depends on the variability observed within a particular lumber region in question.

Chapter 4

Image Registration and Sub-image Extraction

As was discussed in Chapter 3, the hardware and some of the software for the vision system were created by other individuals and are not part of the research reported in this dissertation. However, to understand exactly how the overall system works one must understand the processing that is done by this hardware and software.

Much of the early processing on the images that are collected by the three imaging systems is done by the MORRPH board [DRA95]. This special purpose board reduces the computational load on the image processing computer by some 80 percent. The inputs to the MORRPH board are the raw images from each of the imaging systems. These raw images are denoted as $L_{RAW}(i, j)$, the raw laser image, $X_{RAW}(i, j)$, the raw $C_{RAW}(i, j)$, the raw color image. The MORRPH board is used to do most of the preprocessing that is done on the raw images. By notational convention, the images output by the MORRPH are $L_U(i, j)$, the preprocessed but yet unregistered laser image, $X_U(i, j)$, the preprocessed but yet unregistered X-ray image, and $C_U(i, j)$, the preprocessed but yet unregistered color image. The results of the MORRPH board processing, i.e., these unregistered images, are passed along to the image collection software where the registered images, $L(i, j)$, $X(i, j)$, and $C(i, j)$, will be created. It is also worth noting that another image is created by the MORRPH board processing, a registered black and white image called $BW(i, j)$. The purpose of this work was to create the software that will receive these images from the data collection software and then to do whatever processing that is required to locate and identify defects that lie on a board face or inside a board based on the information contained in these images.

While the MORRPH board performs a variety of tasks including shade correcting the color images, and creating a black and white image from the color image, only three tasks

performed by this hardware will be described here. All of these tasks involve the image registration operation and sub-image extraction operations that are performed on the images. The rest of the operations performed will be discussed in the chapters that describe the processing done on each of the three different imaging modalities, i.e., the laser image, the X-ray image, and the color/black and white image.

The MORRPH board actually has two modes of operation, a setup mode and a run mode. This is possible since the MORRPH is based on field programmable gate arrays (FPGAs). To get the MORRPH to run in the setup mode only requires that the setup code be loaded in the FPGAs and, to then change to the run mode, only requires reprogramming the FPGAs with the run mode code. This code tells each of the FPGAs what hardware functions to implement.

As to the data collection software, this software does more than just collect data, it performs some required tasks, as well, that are related to both the image registration and sub-image selection operations. Its role in both of these operations will be explained in some detail. The discussion will begin by considering the image registration problem.

4.1 Image Registration

As with most multiple sensor systems, the problem of image registration must be addressed before any image analysis can begin. Image registration generally refers to any method, e.g., geometrical transformations, used to make the images from each sensor commensurate in both its spatial dimensions with the images from the other sensors. In other words, if all images are defined under a two-dimensional Cartesian system that have the origin (0,0) at the top/left with the I-axis pointing down and the J-axis pointing to the right, the objective of image registration is to make pixel (i, j) in color image, pixel (i, j) in laser-range image, and pixel (i, j) in X-ray image all refer to the same spatial location on the board. This registration often requires that images from each sensing device have the same resolution and reference point.

Image registration has been investigated by many researchers during the past two decades [SEV78, GIL83, MIT86, BRO92, ZHA93]. Basically, developing an image registration method requires that the following tasks be performed by the developer: Task 1) Choose the resolution and coordinate system of one image as a basis and adjust the other images to this match this basis image, e.g., in this vision system the resolution and coordinates of the color image are chosen as the basis, then the resolution of range image and X-ray image are adjusted by either interpolation or sampling to make these other images all have the same spatial resolution as the color image. Task 2) Identify the objects or landmarks that will be used in the registration, e.g., board edges, likely to be uniquely found in each of the images to be registered. Task 3) Determine the appropriate algorithms to extract these objects or landmarks reliably, e.g., algorithms to find edges in the color domain are different from those used in the range domain. Task 4) Correspond the above objects (often called reference points) and determine a spatial transformation that can properly overlay each of the other images onto the color image.

As to Task 1, in this machine vision system the imaging sensors and imaging geometry have all been selected to meet target spatial resolution of 32 points per inch (or 1.26 pixel/mm) in the horizontal direction (cross board) and 16 points per inch (or 0.63 pixel/mm) in the vertical direction (down board). These resolutions were selected based upon the size of defects secondary remanufactures must detect and identify. As was mentioned in Chapter 3, the color image data is collected at twice this resolution in the horizontal direction to increase the ability to detect small cracks. Detection and identification of small cracks were very important to manufacturers. However, after the crack-preserving filter is employed on the color image, its resolution in the horizontal direction is halved to correspond to the target image resolution. Also because of factors that will be described later, the decision was made to register the X-ray and the laser images to the color image.

As to Task 2, landmarks must be identified that can be used to do the registration. Unfortunately, due to the random nature of hardwood lumber, it is difficult to select a unique object that appears at the same spatial location in each of the three images. An object may show up in one image or two images, but typically not in all three images. For example, color and X-

ray images can be used to detect a knot, but the indications of the knot may not be present in the laser image. Most lumber edges are generally straight, so it may be reasonable to assume that lumber edges can be used as the reference points for image registration. But neither the laser image nor the X-ray image provides accurate lumber edge information. Since wane or holes can exist on the lumber edge, the laser profiling system and X-ray imaging system cannot detect the lumber edge accurately.

What all this means is that the registration operation cannot be done on a board-by- board basis since there typically are no reliably detectable features across all the imaging modalities. Hence, some other method must be used. It is this fact that lead to the concept of having two modes of operation. A setup mode, where the object scanned could be carefully selected so that there are reference points that can be reliably detected across the imaging modalities, and a run mode where boards would be registered based on the information obtained from the setup operation. The object that is imaged during the setup operation is a perfectly rectangular board, i.e., one that has the shape of a rectangular parallelepiped. For such a board the board edges can be easily found in all the different imaging modalities. Since one does not want to have to setup the system very often, there must be a materials handling system that has very good control over how boards are moved through the system so that there is a great deal of repeatability in the way the images are taken. More precisely, each board must move through the system at almost constant velocity where this velocity remains the same from one board to another. Hence, computer control of the motor or motors of the material handling system is required. Also the boards must move through each imaging station in basically the same orientation that it moves through the other imaging stations. To achieve this objective a fence is used and the materials handling system must be able to keep the board against this fence. Because these needs were known early on in the design process the materials handling system was designed to meet these criteria.

Task 3 requires one to define the operators that can be used to detect the reference points. Clearly, whatever these operators are they should be as computationally simple as possible. As one might suppose the operators used are imaging modality dependent.

For example, finding the external edges of a perfectly rectangular board in the laser or X-ray image is relatively straightforward. The laser image and the X-ray image are, in fact, 8-bit gray-level images, where the gray-level of a foreground pixel represents either a range value or a transmission value. The gray-level value of a background pixel usually is either much smaller or much larger than the value of a foreground pixel. Therefore, a simple thresholding on the range values or transmission value can easily differentiate foreground and background and this is computationally about as simple of an operation as one can define.

Finding the external edges of the board in the color image can also be straightforward if a heuristic is used in creating the color image. A color pixel in this vision system is represented by its red, green, and blue components where each component is an 8-bit value. Typically color pixels of a board's surface have red component values that are much larger than their blue component value. The blue component values are always very small. This is particularly true for red oak. The heuristic involves using this fact to select an appropriate color for the background. The background over which the board travels as it passes through the color line scan camera was selected to be blue, a very special kind of blue, one that contains almost no red or green components. To make this blue background visible in the image the blue background is illuminated by its own light source. Imaging a board, in general, and a red oak board, in particular, in this manner trivializes the operation of separating pixels of board form pixels of background in the color imagery [CON89] since pixels in foreground and pixels on background can be separated by a simple threshold on the blue component.

It should be noted that this methodology for finding board edges in color images works not only for perfectly rectangular boards but also for all types of boards. It is a robust method for finding edges. It is this fact that motivated the decision to register the X-ray and laser images to the color image since the color image provides the only reliable method for detecting board edges.

Task 4 involves defining the spatial transforms that will be needed to overlay all the images. Since this machine vision system must operate in real-time, whatever transformations are

used must be as simple as possible. Of all the spatial transformations, translations are the computationally the simplest to implement. Therefore the hardware was designed in such a way as to allow such transforms to be used in the image registration operation.

One way to assure that translations can be used is to appropriately select the imaging geometry that will be used in each of the sensing modalities. In particular, the geometry needs to be selected so that all the sensors have the same spatial resolution. This is the reason that all the sensors are setup to have a 32 pixel per inch cross board resolution and a 16 pixel per inch down board resolution. Finally, the imaging line for each sensor must have the same angle to the direction of travel of the board. For this machine vision system, this angle was selected to be 90 degrees.

However, not only must the imaging geometry be selected appropriately, the material handling system must have certain attributes as well. Just as was the case above the materials handling system must be able to move boards through the imaging systems at a constant velocity, for every board. Next, the orientation of a board must be the same as it passes through each imaging station. Given all the above, all three images can be almost perfectly registered, at least, to the degree demanded of this work by simple translations. Note simple translations of the images will not be allow the images to be perfectly registered because all three sensors have slightly different look angles along the imaging line and are not perfectly positioned. This means that the imaging system will, in reality, not have exactly the same cross board resolutions. And in fact, the cross-board resolutions are slightly different among sensors on this system. For example, the resolution of the X-ray system is 1.17 pixel/mm (29.9 pixels per inch), the resolution of the color system is 1.18 pixel/mm (30 pixels per inch, after two-to-one resolution reduction) and the resolution of the laser system is 1.19 pixel/mm (30.1 pixels per inch). As a consequence, a registered image can be a few pixels off along the board edge that is farthest from the fence. Fortunately, this error will usually not markedly affect the results obtained. The reason for this is twofold. First the lineals for which this system is designed to process are usually only a few inches wide. This limits the effects of this error. Secondly, most of the defects we are looking for are a few pixels wide. This also mitigates the effects of this error. However, it could affect, to

some extent, the identification of small defects such as splits/checks or small knots when information from multiple images (e.g., color image and X-ray image) is needed for their detection. More accurate registration can be achieved by adjusting image resolution using a technique such as image interpretation before the image registration operation is performed. Unfortunately, this is far too computationally complex to meet the real-time processing requirements of this machine vision system. Consequently, these potential problems are all ignored and it is assumed that the boards are nearly perfect in registration.

As was pointed out in Chapter 3, the image processing computer controls when each imaging sensor will begin image collection and when it will end image collection. While theoretically this control would preclude the need for any image registration translations in the down board direction, practically speaking it is difficult to define exactly what these time offsets should be. Because of this, two translations are needed to register either the laser image or the X-ray image to the color image. Therefore only four parameter values are needed to define all the spatial operations that need to be performed. Further, since the materials handling system precisely controls board speed these four parameters do not vary across boards but rather once their value has been determined these values remain constant.

As should be clear by now, the goal of the setup operation is to define the value of these parameters and the goal of the run mode is to use these values to register the laser and the X-ray images to the color image so that image processing and analysis operations can be perform on this suite of registered images.

4.2 Sub-image Extraction

Obviously, the imaging geometry for all the imaging modalities is setup in a way so that each can image the widest possible board. This width was determined based on the requirements provided by secondary re-manufacturers. Since vast majority of boards that will be processed have widths that are substantially smaller than this maximum width, it is desirable for purposes of real-time processing to get rid of the useless background data as soon as possible. This is the motivation for doing sub-image extraction.

Since boards are basically rectangular in shape, an easy way to implement this sub-image extraction is to find the minimum-inscribing rectangle and use this rectangle to define the sub-image. To do this requires a reliable way of determining board edges. The most reliable method for doing this is to use the above described blue channel threshold on the color image. Consequently, the color image is processed to find the needed upper left hand corner and the lower right hand corner of the minimum-inscribing rectangle. Given these coordinates and the image registration translation factors needed to register the X-ray and laser image to the color image, the minimum inscribing rectangles for these other two imaging modalities can be computed based on the values determined for the color image.

Actually a rectangle somewhat larger than the minimum-inscribing rectangle is used to create the sub-images. This slightly larger rectangle is used since it is important that all of the board appears in the sub-images for each of the imaging modalities and since, as was stated previously, the image registration operation based on translations is not exact. The slightly larger rectangle guarantees that the sub-image extracted for each imaging modality will contain all the board surrounded by some small background area.

4.3 Implementation Details

As was discussed in the beginning of this chapter, the MORRPH board and the data collection software interact to perform both the image registration and the sub-image extraction operation. In this section, the nature of this interaction will be described in some detail.

The first thing that has to be done, before the system can be brought up to operational status, is a setup operation must be performed. The goal of this setup operation is to define the four translation constants needed to register the X-ray and laser images to the color image. To do the setup operation, both the data collection software and the MORRPH board must be configured to do the setup. After this has been done, a special target is run through the system. This special target is a “perfect” board, i.e., one that is near to a rectangular parallelepiped as

possible. As this board is run through the system, the MORRPH board extracts the board edges from all three imaging modalities using the methods described previously. It uses the edge locations to determine the upper right hand corners of the minimum inscribing rectangles for each of the imaging modalities. The values of these corner points for each of the imaging modalities are then passed to the data collection software. This software subtracts the location of the color image corner point from the X-ray image corner point and stores this two-dimensional vector value as a program constant. This two-dimensional vector defines the two offset values needed to define the translation spatial operation that registers the X-ray image to the color image. The exact same operation is done with regards to the laser image. The two-dimensional vector value created during this operation defines the offsets needed to register the laser image to the color image. After finding this last set of translation values the setup mode is terminated.

Note that two other operations are performed during the setup operation as well. These operations include certain constant data that are needed to do shade correction on both the X-ray image and the color image of a board.

Now before describing the run mode of operation, some information about the data collection software is needed. This software maintains three blocks of memory for its use. Two of these three blocks are used to store the unregistered image data collected from the MORRPH board. These images are the unregistered color image, $C_U(i, j)$, the unregistered laser image, $L_U(i, j)$, and the unregistered X-ray image, $L_U(i, j)$. Note that all but one of these image data that are stored in these locations have undergone some type of preprocessing operations, the nature of these preprocessing operations will be discussed in upcoming chapters. Two of these unregistered image blocks are used so that continual data collection can be occurring. When the first board in run mode is scanned, these images are stored into the first block. As the second board is scanned the images are put in the second raw data block. The third board image goes back into the first block, etc. The basic idea behind this machine vision system is to be able to process information from a board while the next board is being scanned.

The final memory block is used to store the registered sub-images, i.e., the color registered sub-image, $C(i, j)$, the laser registered sub-image, $L(i, j)$, and the X-ray registered sub-image, $X(i, j)$, that are created by the data collection software for a board. Note that these sub-images are images in their own right and these images are registered so that for each (i, j), $C(i, j)$, $L(i, j)$, and $X(i, j)$ all correspond to the *same* spatial location on the board. To store these sub-images only one memory block is needed since the storage of the sub-images into this block is considered part of the processing that needs to be done on the board image.

The run mode operation begins by having the data collection software load the MORRPH board with the run time program for each of its FPGAs. After this operation is done, the system is ready to operate. A board is run through the system. The MORRPH board extracts the edges of the board from the color image, basically $C_U(i, j)$, using the threshold on the blue channel mentioned previously. It uses this information to compute the upper left hand corner and lower right hand corner points for the minimum-inscribing rectangle of the board in this color image. Please note that the MORRPH performs these operations after the crack preserving filter has been applied so that the resolution of the color image is the same as that of the other images at this stage of the processing. Note the MORRPH board is also doing many other tasks at the same time including passing image information on to the data collection software. The nature of these other tasks will be described in later chapters.

Once all the data has been collected and stored into an unregistered image memory storage block the MORRPH passes the upper left and lower right corner points found in the color image on to the data collection software. After getting this information, the data collection software subtracts the same small positive values to the upper left hand corner point and adds the same small positive values from the lower right hand corner point. This enlarges the size of the minimum inscribing rectangle and defines the sub-image, $C(i, j)$, that will be extracted from the raw color image, $C_U(i, j)$. The data collection software then uses these corner points to compute the dimensions of $C(i, j)$, i.e., *Row_Size* and *Col_Size*. Then the translation offset values for the laser image and X-ray image are added to these revised color image corner point values to define the sub-images for these imaging modalities. Then these coordinates are used to extract $C(i, j)$

from $C_U(i, j)$, $L(i, j)$ from $L_U(i, j)$ and $X(i, j)$ from $X_U(i, j)$. The data collection software then moves these sub-images to the registered sub-image memory storage area. Once this is accomplished, the analysis programs that were created in this research act upon these sub-images to detect and identify the wood defects.

Note that as the unregistered color image, $C_U(i, j)$, is being collected, an unregistered black and white image, $BW_U(i, j)$ of the board is being computed from the color image, $C_U(i, j)$. And just as in the case with the other images, a black and white sub-image, $BW(i, j)$, is extracted from $BW_U(i, j)$, and places it in the sub-image memory block. Note, the black and white image, $BW(i, j)$, will be used in the segmentation of the color image.

One last final note, it should be clear that the spatially registered images $C(i, j)$, $L(i, j)$, $X(i, j)$, and $BW(i, j)$ all have the same dimensions and that these dimensions are *Row_Size* by *Col_Size*. In each of these images, a row is defined to run across the width of a board and a column is defined to run down the length of a board.

4.4 Summary

This chapter described some of the processing that is performed by the MORRPH board and the data collection software in this machine vision system. In particular, this chapter has described how the registered images $C(i, j)$, $L(i, j)$, $X(i, j)$, and $BW(i, j)$ are created an how their dimensions Row_Size by Col_Size were determined. The analysis of these images to locate and identify possible wood defects is the thrust of the work that is reported in the next chapters.

Chapter 5

Foreground/Background Differentiation

As discussed in the last chapter, the MORRPH board and the data collection software generates nearly perfectly registered images, i.e., the images $C(i, j)$, $BW(i, j)$, $L(i, j)$, and $X(i, j)$, from the data that is collected from the color, laser, and X-ray imaging systems. Also remember that the MORRPH board provides the information needed for the data collection software to determine the size of these images, i.e., *Row_Size* by *Col_Size*. It is these images that will be analyzed by the algorithms developed during the research effort to locate and identify wood defects.

This chapter describes the first step in this analysis process, one whose output is used in the analysis of all the different images. The purpose of the algorithm described in this chapter is to perform a foreground/background differentiation where the foreground elements are picture elements (whether color pixels, black and white pixels, range elements or transmission elements) that are from the boardand where background elements are picture elements that are not from the board. Making this determination early means that the time consuming operations of low-level processing only need be applied to those elements that are of the board. This markedly reduces the computation complexity of these operations.

As to why foreground/background differentiation is needed it is very important to point out that few if any boards are perfectly rectangular parallelepipeds. Some boards vary significantly from this shape. Secondly, in the processing that was described in Chapter 4, it was noted that the images $C(i, j)$, $BW(i, j)$, $L(i, j)$, and $X(i, j)$ were intentionally created to contain some background elements along all of edges. Therefore the purpose of this processing is to locate where the board boundaries actually exist in these images.

5.1 Algorithm for Differentiation

As was argued in Chapter 4, the most robust way of locating board edges is based on the processing of the color image since this image's background is a special blue color and is illuminated by its own light source. Since the color of wood, in general, and red oak, in particular, contains very little blue in it, the blue background can easily be differentiated from the wood on the basis of a simple threshold on the blue channel. Hence the input to this algorithm is the color image $C(i, j)$ and the dimension of this image as is specified by *Row_Size* and *Col_Size*. The output of the algorithm is an integer matrix *Board_Edge* that has dimension Row_Size × 2. The other outputs are two integers, *Board_Start* and *Board_End*. *Board_Start* gives the row number of $C(i, j)$ where the first board edge was detected while *Board_End* gives the row number of $C(i, j)$ where the last board edge was found. In the rows between *Board_Start* and *Board_End*, i.e., for i such that *Board_Start* = i = *Board_End*, *Board_Edge*$(i, 1)$ gives the column number where the right edge of the board appears in row i while *Board_Edge*$(i, 2)$ gives the column number where the left edge of the board is located in row i.

Recall from Chapter 4, the rows of $C(i, j)$ run across the board while the columns of this image run down the board. Given this situation the search for the board edges are conducted on a row-by-row basis starting with row 1. The algorithm begins by setting *Board_Found* = *False*. The algorithm starts with the second pixel of each row, say row i, and works its way across the board until it finds the first color pixel whose blue component is greater than a preset threshold, T_{Blue}. The algorithm continues the search for this element until it reaches column *Col_Size* - 1. If the algorithm finds a pixel that is greater than or equal to the threshold T_{Blue} the algorithm records the column number where this pixel is located, say column j_{Right}. If this is the case then *Board_Edge*$(i, 1)$ is set equal to j_{Right} and begins search for the left board point of the board by starting the search at $C(i, Col\_Size - 1)$ working backward across the row until it reaches j_{Right}. Since, in this particular circumstance the algorithm must find a color pixel is equal to or greater than T_{Blue}, the algorithm records this column, say j_{Left}, by setting *Board_Edge*$(i, 2) = j_{Left}$. Next the algorithm checks *Board_Found*. If *Board_Found* = *False* then the algorithm sets

Board_Start = *i* and sets *Board_Found* = *True*. Then it continues the search all over again on the next row. If no pixel is found during the search from columns 2 to *Col_Size* – 1 and *Board_Found* = *False* then the algorithm begins the search of the next row starting from the right hand side of the image. Otherwise, *Board_Found* must be *True,* which means that the trailing edge of the board has been found. Suppose this occurs at i_{TE}, then *Board_End* = i_{TE}.

It is worth noting that this algorithm finds only the right most and left most edges of a board. Should a board contain voids such as large cracks or large holes, the edges formed by these features would not be detected.

5.2 Summary and Conclusions

In this chapter the algorithm for doing foreground/background differentiation was given. The outputs of this algorithm are *Board_Start*, *Board_End*, and *Board_Edge*. *Board_Start* is an integer that specifies the first row of $C(i, j)$ that contains a board edge. This means that the board does not appear in $C(i, j)$ until row *Board_Start*. Similarly, *Board_End* is an integer that specifies the last row of $C(i, j)$ where a board edge is detected. This means that the board does not appear in $C(i, j)$ after row *Board_End*. *Board_Edge* is a *Row_Size* × 2 integer array. For all *i* such that *Board_Start* = *i* = *Board_End*, *Board_Edge*(*i*, 1) gives the column in $C(i, j)$ where the right most edge of the board is located while *Board_Edge*(*i*, 2) gives the column in $C(i,$ j) where the left most edge of the board appears.

Further, since $C(i, j)$ is nearly perfectly registered with $L(i, j)$, $X(i, j)$ and perfectly registered with $BW(i, j)$, the values *Board_Start*, *Board_End*, and *Board_Edge* give the locations of the board in these images as well.

Chapter 6

Analysis of Laser Images

As discussed in Chapter 3, laser-range profiling is intended to locate and identify defects associated with critical thickness deviations such as wane and thin components in the board. Both defects are thinner than normal wood and, therefore, are indicated by dark areas in the laser image. The laser-range image is processed first because the processing and analysis of the laser image is the most straightforward. The reason for this is that one can precisely define what is important in the laser image; that is, areas thinner than a certain acceptable thickness threshold. The areas of interest cannot be so easily defined in the color image or the X-ray image. Moreover, after wane and thin components are removed, the subsequent analysis of the X-ray and color images is more accurate and less computationally complex. The wane and the thin board areas often have lower densities than clear wood. If these areas are removed from the histogram of the X-ray image before the selection of thresholds, the visibility of other lower-density defects in a histogram will be increased, and a better threshold for lower-density areas can generally be obtained. The removal of wane from the histogram of the color image can increase the effects that knots and holes have on the histogram and, therefore, allow more accurate thresholds to be obtained. As a consequence, other defect components can be detected accurately and efficiently in the subsequent processing.

The actual analysis of laser image involves four levels of processing: preprocessing, low-level processing, intermediate-level processing and high-level processing. The only preprocessing done on the laser image is of the type that is described in Chapter 4. However, by notational convention the laser image $L_{RAW}(i, j)$ is the notation used to represent the raw image that comes directly from the laser imaging system into the MORRPH board for processing. The output of the MORRPH board of is denoted by $L_U(i, j)$. $L_U(i, j)$ is called the unregistered laser image that is processed by the data collection software. Please note that since no preprocessing is done on the

laser image by the MORRPH board $L_{RAW}(i, j)$ actually is the same as $L_U(i, j)$. However, for sake of notational convention $L_U(i, j)$ will be used to denote the unregistered laser image anyway. It is $L_U(i, j)$ that is the input for the processing described in Chapter 5. Finally, the registered laser image is denoted by $L(i, j)$. It is the laser image that has had most of the background removed and is nearly perfectly registered to the color image, $C(i, j)$, and the X-ray image, $X(i, j)$, i.e., it is the output of the processing of Chapter 5. It is the registered laser image $L(i, j)$ that is the input to the low-level processing modules described here. The same notational conventions will be used in the chapters that follow.

The low-level processing extracts image primitives such as edges or components that contain possible objects of interest found in the input laser image, $L(i, j)$. The low-level processing of a laser image in this vision system consists of image segmentation, a segmentation based on an image thresholding operation, and connected-component labeling. The intermediate-level processing attempts to characterize the image primitives found in the segmentation operation by extracting measures from these primitives and combining these measures to form a measure vector. The goal of the high-level processing of the laser image is to identify wane and thin-board defect regions.

The objective of this chapter is to provide a detailed description of the low-level, intermediate-level, and high-level algorithms used in the analysis of the laser image and to demonstrate processing results that were obtained from these algorithms.

While it is not explicitly stated in the description of the low-level vision algorithms, these algorithms only are only applied to range elements that are from the surface of the board. This application is possible because of the processing that was described in Chapter 5. This processing computes the values for *Board_Start*, *Board_End*, and *Board_Edge*. The algorithm for using *Board_Start*, *Board_End*, and *Board_Edge* to find only those range values that are only from the board face is straightforward. It is straightforward since the color image is almost perfectly registered to the laser image. Hence, this algorithm will not be discussed in this chapter.

Finally, it should be noted that there is a basic problem associated with laser images. If part of the board's surface is very dark then the amount of light reflected from the incident laser beam is small, frequently too small to be detected by the imaging cameras. In this event the laser imaging system assigns a range value of 255. Since this value obviously may not truly reflect the thickness of the board, the analysis only the laser image has no means by which to judge what is really happening when this range value is reported. Consequently, components having this range value are not labeled by the high-level algorithms. Rather these components are passed on and are not analyzed again until the analysis of the color image is performed. When this analysis is done the darkness of the surface can be determined and an appropriate label for such components can be determined.

6.1 Low-Level Processing of Laser Images

The low-level processing of a laser image extracts image regions (or components) that contain possible objects (or defects) of interest found in the laser image. As was discussed in Chapter 2, some of the region-based image segmentation approaches are more appropriate for low-level processing in real-time computer vision systems since these approaches are generally less computationally complex. Of all these region-based approaches, the computationally simplest method is one that consists of the following steps: 1) an image thresholding operation, which separates normal pixels from pixels that are possibly in a defect area; and 2) a connected-component labeling operation, in which similar pixels are assigned to the same labels to form initial components; this approach was selected primarily due to its computational simplicity and because of the very good results it gave during the testing.

The above approach to image segmentation creates two new images. In the first step, the image thresholding operation takes a laser image, $L(i, j)$, as input and outputs another image--a thresholded image, $L_T(i, j)$, that is exactly registered to the laser image. The value of each pixel in the thresholded image is determined by examining the corresponding pixel in the laser image and seeing if this pixel has a value above or below a threshold. The thresholded image is subsequently used as the input to the connected-component labeling operation. The connected-component

labeling operation then generates a new image, a connected-component (or segmented) image, $L_C(i, j)$, that is exactly registered to the thresholded image. In this image, all pixels in the same connected component have the same value, and this value is different from the value of a pixel in any other connected component.

Ideally, a laser image of a board provides an accurate thickness profile of the board. The thicker the board, the larger the pixel values in the corresponding laser image. But like many other imaging systems, the laser-profiling imaging system has some limitations. First, the system is not able to detect wane or board regions that are too thin, due to the dynamic range of the laser-profiling system. Second, the system cannot sense board thickness when the laser line passes over very dark regions of a board surface, i.e., when black surface stains, dark knots, and deep holes [HOU93, LAC97] are present on the board. The reason this is the case stems from the fact that little of the laser light is reflected off these surfaces. In fact, the amount of light reflected is diminished to the point that the laser-profiling system cannot detect the presence of any light coming off such areas. When this occurs, the laser-profiling system outputs a value of five (while the background is set to 0) as the detected range value. Hence, the laser-profiling system cannot differentiate among defects such as dark surface stains, dark knots, deep holes, and *very* thin areas. The regions containing these defects are classified as *unknown* areas by the analysis software since what is inside these areas cannot be unambiguously determined using the information in laser image alone.

6.1.1 Thresholding

The first operation that is performed by the low-level processing is thresholding. This operation takes as input the registered laser image $L(i, j)$ and then computes a histogram, H_l, of the board area of this image. The thresholds are selected by examining the shape characteristics of H_l.

The purpose of this thresholding operation is to differentiate pixels of normal surface area, i.e., areas that are thick enough, from pixels that might be in a defect area, i.e., either too thin or

unknown. This operation, in fact, classifies all pixels into one of several classes, and each class represents one of the above areas. This operation is accomplished by 1) assigning each pixel in a laser image, $L(i, j)$, a new value that represents one of the above areas according to so-called thresholds; and 2) storing this new value into the thresholded image, $L_T(i, j)$ that is registered to the original image. Therefore, before performing the operation, one must first determine the threshold or thresholds that will separate pixels that possibly contain a defect from those pixels that represent normal wood.

From the discussion in 6.1.1, it should be clear that foreground pixels in the laser images could be classified into any one of three categories. These are: 1) pixels that are from areas of the board that are thick enough; 2) pixels that are from areas of the board that are too thin to appear in a rough part, i.e., wane and thin-board regions; and 3) pixels that are from areas of the board whose identification cannot be determined from the laser image alone. Theoretically, two thresholds can be used to separate these three categories of pixels. The first threshold T_{TE} is used to separate thick-enough pixels from the other pixels, since the value (i.e., range value) of a thick-enough pixel is greater than the values of other pixels. The second threshold T_{TU} is used to separate the *unknown* (thickness unknown) pixels from pixels representing areas that are too thin because the range values of these *unknown* pixels are much smaller than the range values of pixels in thin areas. Figure 6.1 shows a histogram of a typical laser image that contains these three categories of pixels. Clearly, threshold T_{TE} separates thick-enough pixels from pixels representing thin areas and *unknown* pixels. While threshold T_{TU} isolates *unknown* pixels from pixels representing too-thin areas, background pixels are pixels whose value is zero.

The first threshold T_{TE} can be obtained by measuring pixel values (range values) in the laser image of a flat board with a desired target thickness. For example, if the required thickness for a rough part is 25.4 mm (1 inch), one could measure a calibration target of clear wood that is 25.4 mm thick and find the average range value of its laser image. Generally, for a certain type of rough part, the minimum board thickness required by the manufacturer is known before the inspection begins. Another method for determining this threshold is to find a valley point on the left side of the major peak of the histogram of the laser image, and use the location of the valley

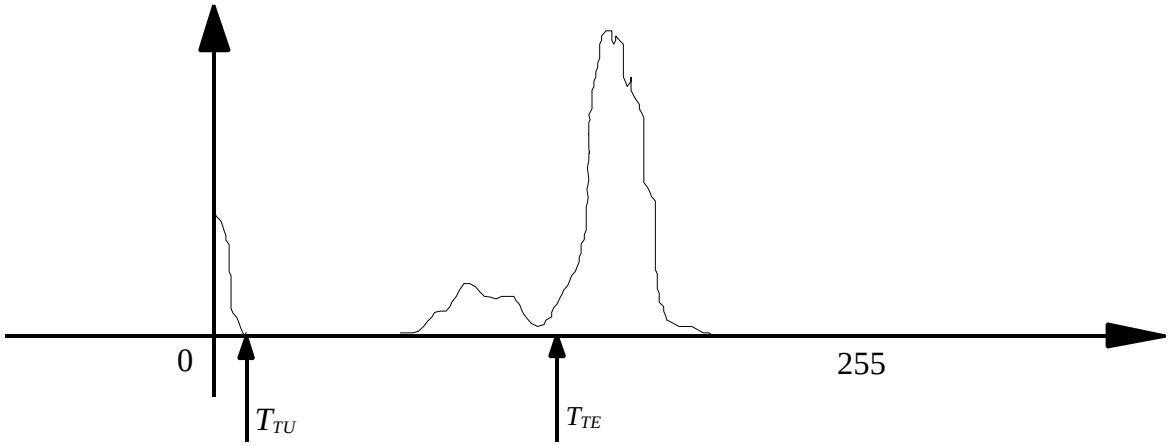


Figure 6.1 A histogram of a typical laser image, with three pixel classes: thick-enough ($\geq T_{TE}$), thin, and thickness-unknown ($\leq T_{TU}$)

as the threshold. Details regarding how to find such a valley point will be discussed in Section 7.2. In this research, although both methods have been tested, the former is selected due to both its simplicity and its ability to assure that wooden pieces meet a certain thickness requirement. In practice, T_{TE} is pre-determined during calibration and stored in a configuration file. The threshold T_{TE} can be changed by modifying this configuration file.

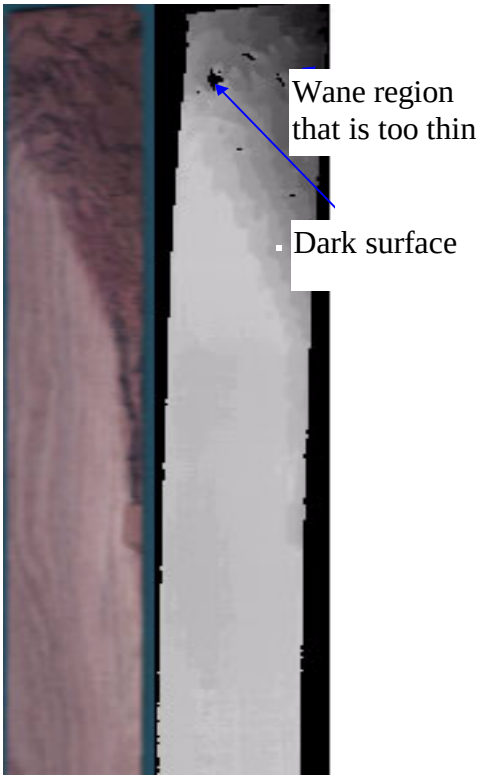
The second threshold T_{TU} used to segment unknown pixels is often assigned a very low value and stored in the configuration file. In this research, the laser-profiling system assigns a 5 to each *unknown* pixel. Therefore, a second threshold T_{TU} = 10, for example, would adequately separate *unknown* pixels from pixels representing thin-board pixels. Threshold T_{TU} is considered an empirical parameter that depends on the observed distribution of *unknown* pixels and the minimum value of pixels in thin areas. This threshold could be any value that separates *unknown* pixels and pixels that represent too-thin areas. For most of the laser images collected during the course of this research, the range values of *unknown* pixels are 5, and a range value of a typical pixel in a thin-board area is generally greater than 60. Hence a T_{TU} value of 10 will clearly separate these two classes of pixels for the system.

Thresholds T_{TE} and T_{TU} are then used to perform a global thresholding on the laser image. The resultant image is a thresholded laser image whose values are determined based on Equation (6.1) given below:

$$L_T(i,\ j) = \begin{cases} 50 & if \quad 0 < L\ (i,\ j) < T_{TU} \\ 100 & if \quad T_{TU} \le L\ (i,\ j) < T_{TE} \\ 150 & if \quad L\ (i,\ j) \ge T_{TE} \\ 0 & if \quad L\ (i,\ j) = 0 \end{cases} \tag{6.1}$$

The thresholded image $L_T(i, j)$ is an image that is perfectly registered to the laser image and has possible pixel values of 0 (represents background), 50, 100, and 150. While any values within the range of 0 to 255 could have been selected for use in $L_T(i, j)$, the pixel values in Equation 6.1 were selected because they allowed one to easily distinguish the different regions when the thresholded image was viewed on a display.

An example of a thresholded laser image is shown in Figure 6.2. Figure 6.2 (a) shows both the color image and the laser image of a board segment that contains wane and dark surface areas. The color image shows that some areas of this board segment are very dark, so dark that



(a) color image (left) and laser image of a wane region

(b) thresholded laser image (left) and labeled laser image

Figure 6.2

the laser-profiling system cannot find the laser line over these areas. In the laser image, these dark areas manifest themselves as very dark areas since the laser-profiling system cannot detect the laser line and it assigns very low non-zero pixel values (e.g. close to 5) to pixels from these areas. Clearly, these areas are much darker than the wane regions in which pixels have relatively large range values. Thick-enough areas are bright since their range values (pixel values) are larger than those of pixels in the wane regions. Figure 6.2 (b) shows the corresponding thresholded laser image (left) after applying thresholds T_{TU} and T_{TE}, and the labeled laser image (to be discussed in Section 6.2.3). In the thresholded image the *unknown* areas, wane regions (too-thin regions) and thick-enough regions are clearly separated from each other.

6.1.2 Connected-Component Labeling

The thresholding operation described above classifies pixels in the laser image into one of the four classes: background, normal surface areas (areas that are thick enough), "too-thin" areas, and *unknown* areas (areas whose classification cannot be determined from the laser image alone). To find discrete components (or objects) that represent possible defects, one needs to join these classified pixels by using neighborhood connectivity rules. This operation is called connected-component labeling. The goal of connected-component labeling is to determine exactly how many possible defects are in a thresholded image and to assign each of them a unique label. This operation takes the thresholded image, $L_T(i, j)$, as input and produces another image, a connected-component (or segmented) image, $L_C(i, j)$, that is exactly registered to the thresholded image .

Although many methods for connected-component labeling have been proposed and studied by researchers [SHI87, JAI89, GON92, HAR92], there appear to be no efficient methods which can handle thresholded images that have more than two values. Most of the methods published and tested can only be applied to the connected-component labeling operation of a binary thresholded image (i.e., a pixel is either foreground or background). Since thresholded images produced in this research are images with multiple pixel levels, these methods cannot be used here.

An efficient method for performing 4-connectedness connected-component labeling has been developed in this research to process such thresholded images. This new algorithm is given below. It takes two passes to label the connected components in the thresholded image. In the first pass, each pixel $L_T(i, j)$ is examined and assigned a tentative component label. The output is stored as a segmented image $L_C(i, j)$. In the second pass, each component label is examined and equivalent labels are merged.

The algorithm for the first pass is described as follows:

1) First initialize a $K \times 1$ array **r** as follows:

   for n = 1 to K

   r(n) = n;

   where K is the maximum number of possible components. The use of this array will be discussed in greater detail in Step 5.

2) Start at the top left-hand corner and proceed from left to right, top to bottom to examine each pixel $L_T(i, j)$;

3) If pixel L_T (i, j) has a background value (zero), then assign L_C (i, j) (in the labeled image) a zero value.

4) If pixel L_T (i, j) has a foreground value (non-zero), and its up (L_T (i-1, j)) and left (L_T (i, j-1)) neighbors are background, then assign L_T (i, j) a new component label and store it in L_C (i, j). A component label is a unique number that ranges from 1 to the maximum number of possible components. A component counter (initialized with 1) is used to assign and track how many component labels have been used. If either neighbor L_T (i-1, j) or L_T (i, j-1) has the same value as L_T (i, j), assign the corresponding neighbor's component label to L_C (i, j). Suppose it is L_T (i, j-1) that has the same value as $L_T(i, j)$, then L_C (i, j) would be assigned the label that L_C (i, j-1) has. If on the other hand both neighbors L_T (i-1, j) and L_T (i, j-1) have identical values to each other and both neighbors have the same component label, then assign that label to L_C (i, j). If the neighbors have different component labels, assign the smaller component label to L_C (i, j). In the later case where both neighbors L_T (i-1, j) and L_T (i, j-1) have the same pixel level value but different labels, the two labels are equivalent

labels. Equivalent labels are then stored in a one-dimensional array **r** as discussed in Step 5. Finally, if neither of the neighbors has the same pixel level value that L_T (*i-1*, *j*) has, and both are not background, assign L_C (*i*, *j*) a new component label.

5) If label L_C (*i*, *j*)= n_1 and label L_C (*i-1*, *j*) or L_C (*i*, *j-1*) = n_2 are equivalent labels as found in Step 4, then set

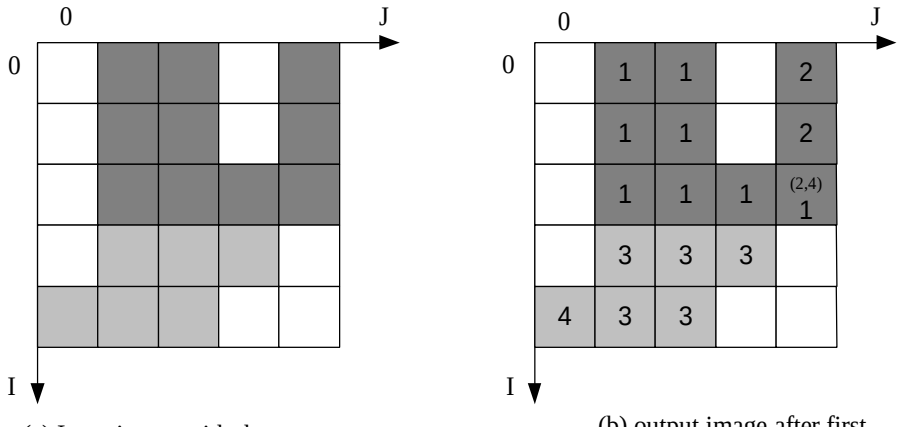
$r(n_1) = n_2$

and set

$r(n_2) = n_1$

The above operations exchanged content of $r(n_1)$ and $r(n_2)$. This makes $r(n_1)$ point to the component label n_2 (via $r(n_1) = n_2$) and $r(n_2)$ point to the component label n_1. The initialization in Step 1 makes an entry r(*n*) point to itself (i.e., $r(n_i) = n_i$). If $r(n_i) = n_i$, then label n_i has no equivalent label.

The above algorithm is applied to each pixel in the thresholded image and results in a labeled image and an array that contains pointer information about equivalent labels. The first pass of the connected-component labeling is illustrated with a sample image in Figure 6.3. The input image is displayed in Figure 6.3 (a). It contains foreground pixels for two different components (shadowed squares of two different gray levels) and background pixels (white squares). When the above algorithm is applied to the first two rows, the pixel located at (0,1) is first assigned label 1, while pixels at (0,4) and (1,4) are assigned label 2. Obviously all the above



(a) Input image with three pixel values

(b) output image after first pass

Figure 6.3 Example of connected-component labeling

pixels are connected via Row 3 and have the same gray level. Consequently, they are found to be equivalent labels when the pixel at (2,4) is examined. Next, foreground pixels at Row 3 are assigned a new label (label 3) because their pixel value is different from the value of pixels in Row 2. Subsequently, the pixel at (4,0) is assigned a new label since it is the first pixel in Row 4. But it is equivalent to label 3. The output image is shown in Figure 6.3 (b).

The changes within the entries of array **r** are shown in Figure 6.4. The array **r** has the initial values shown in Figure 6.4 (a). When labels 1 and 2 are found to be equivalent, **r**(1) and **r**(2) are exchanged [see Figure 6.4(b)], since all related pixels are of the first gray level. This makes label 1 and label 2 point to each other. Later, labels 3 and 4 are found equivalent and the contents of **r**(3) and **r**(4) are then exchanged as shown in Figure 6.4 (c). At this time, all related pixels are at the second gray level. Upon finishing the first pass, all equivalent labels have been so recorded in array **r**. Therefore, the equivalent component labels can be obtained by later tracing the pointers.

$$\begin{bmatrix}1\\2\\3\\4\end{bmatrix} \qquad \begin{bmatrix}2\\1\\3\\4\end{bmatrix} \qquad \begin{bmatrix}2\\1\\4\\3\end{bmatrix}$$

(a) (b) (c)

Figure 6.4 The Content of array **r** during the first pass

After the first pass is over, the equivalent component labels can be obtained by tracing the pointers. At the same time, final component labels are stored in the array **r** for the purpose of assigning correct component labels in the second pass. This tracing process is illustrated in Figure 6.5. The value of array **r** after the first pass is finished is shown in Figure 6.5 (a). The first element of **r** indicates that components 1 and 2 should be merged. This merging operation assigns components label 1 as the final label, i.e., the final label is the minimum of the label assignments for the two components. The third value of **r** indicates that components 3 and 4 are equivalent. The merged component is assigned the final label of 3. The result is shown in Figure 6.5 (b). Then the component labels are re-numbered as Figure 6.5 (c) to reflect the final labeling of the components.

$$\begin{bmatrix}2\\1\\4\\3\end{bmatrix} \qquad \begin{bmatrix}1\\1\\3\\3\end{bmatrix} \qquad \begin{bmatrix}1\\1\\2\\2\end{bmatrix}$$

(a) (b) (c)

Figure 6.5 Changes in array **r** when resolving equivalent labels

This process of renumbering is accomplished using the following algorithm. Here k is the maximum number of tentative component labels, and the braces {} are used to delineate comments.

$j \leftarrow 0$; { counter for components }

For $i \leftarrow 1$ to k

If r(i) < i do nothing; { since r(i) has been updated. }

Else

$j \leftarrow j + 1$; $h \leftarrow i$ { increase counter j }

while r(h) $\neq$ i

do

$n \leftarrow$ r(h); r(h) $\leftarrow j$; $h \leftarrow n$; {exchange content of **r**}

The above algorithm resolves the equivalent component labels and stores the resulting final labels back into the **r** array. The result is a simple table lookup that will be used to do the final labeling.

With the equivalent labels resolved, the output image can be labeled in a straightforward way during the second pass. In the second pass, image L_C (i, j) is examined from left to right, top to bottom. That is if $L_C(i, j) > 0$, then $L_C(i, j)$ is assigned its equivalent label stored in r($L_C(i, j)$). Since if $L_C(i, j) > 0$, then $L_T(i, j)$ must be foreground, and $L_T(i, j)$'s smallest equivalent label is stored in r($L_C(i, j)$). After the second pass, each component is assigned a unique label. Connected components that possibly could be a defect or a part of defect are formed and stored in the segmented image $L_C(i, j)$ after the second pass.

Since the first pass stores the relations of equivalent component labels in a one-dimension array, the equivalent labels can be resolved quickly by simply processing the array as discussed above, a very computationally inexpensive procedure to follow. During the second pass, equivalent labels are quickly determined by simple table lookup and the final component labels are efficiently assigned. Therefore, the algorithm can label images of multiple gray values in a very computationally efficient manner.

6.2 Intermediate-level Processing of Laser Images

To identify what the components in the segmented image $L_C(i, j)$ really are, one has to determine which component measures can potentially be used to identify each defect class. Component measures are simply measures defined on the component that provide some type of information about the component. In this research, a set of basic component measures is extracted first to eliminate or merge components. Then additional component measures that are derived from the basic measures are used to perform the high-level processing (Section 6.3).

6.2.1 Initial Measurement Extraction

The input into this processing operation is the connected component labeled laser image, $L_C(i, j)$.

The primary concern in the analysis of laser image is to identify two defect classes: wane and too-thin board. Since average range value can indicate how thick a component is, any component thinner than a threshold would be a defect. The location information can suggest if a component is on a board edge (wane often appears on a board edge). Information regarding the shape of a component (e.g., height/width) could be used to differentiate wane vs. thin board because wane usually appears as a long stripe in the vertical direction (I direction). Therefore, average range value, as well as the location and shape of a component provide important clues as to what a component might be. The area of a component is another important component

measure because it can be used to determine if a component is large enough to be a defect or if it is small enough to be considered noise.

The following basic component properties are the ones initially extracted from the components found by the connected-component labeling procedure:

1. Average range value
2. Upper left and lower right corners of the minimum inscribing rectangle
3. Center of mass as defined in Section 2.1
4. Height and width of minimum inscribing rectangle of the component
5. Perimeter defined as total numbers of pixels on the component's boundary
6. Touch_edge defined by the ratio of the number of boundary pixels that touch a board edge to the total perimeter of the component
7. Area of the component

Two of these component measures are needed immediately to do small component elimination and component merging. These are a component's average range value and the component's area. The rest are computed here because it is computationally simple to do so.

Among the components found above, there are one or more components whose range values are greater than threshold T_{TE}, and whose area is 20 percent or more of the total area of the board. These components represent clear wood (i.e., the board can be used for rough parts) since it has been assumed (see Section 1.3) that defect areas occupy only a small percentage of total lumber surface area. Usually, there is only one component of clear wood. But there might exist one or more defects that whose area goes completely across the board widthwise. When this occurs there are two or more components that represent clear wood. Such components can be easily determined from their area sizes and range values, and therefore are identified after initial component measures are extracted. The component label of clear wood is a label needed in subsequent processing.

The output of this processing operation is a list of components and a list of measurement values associated with these components. These values are the inputs to the next processing step which also has as input $L_C(i, j)$.

6.2.2 Small Component Elimination

The small components (i.e., ones containing only one or two pixels) formed by the connected-component labeling are most probably caused by digitization noise and the nature of random structures in lumber. Experiments have shown that these small meaningless components make as much as 70 percent of the components formed by the connected-component labeling process. Therefore, these small components should be eliminated before any further processing is performed since they often are not part of any actual defect. Also, defects (such as wane) are almost always over-fragmented due to the nature of rough surfaces of defects. The components comprising this over-fragmented version of the defect need to be merged so that the more reliable measures can be extracted, ones that will more accurately reflect the attributes of the complete defect. These two operations are performed by a small component elimination algorithm and a component merging algorithm.

Small component elimination is a simple operation that eliminates components whose areas are too small to be meaningful. The algorithm performing the small component elimination is as follows: 1) For each component whose area is smaller than a pre-selected threshold (i.e., two pixels for this research), its average range value is compared to the average range value of any component with whom the small component shares a common border. 2) Find the neighboring component whose average range value is closest to that of the average range value of the small component. 3) Merge this small component to the component found in Step 2, if such a component exists. This merging is accomplished by assigning the component label of the component found in Step 2 to all pixels in the small component. If no such neighbor is found, simply delete this component. The deletion is accomplished by assigning the label of clear wood to all pixels within the component. Both of the above operations make changes to the segmented image $L_C(i, j)$. Consequently, one additional step is needed. 4) The measures of the small

component must either be deleted or a new set of measures must be computed from the newly created merged component.

The small component elimination is performed first, due to its computational simplicity and due to the fact that eliminating small components from consideration markedly reduces the work that the component merging operation must do. Currently, only components containing one- and two-pixels are eliminated by this operation. Examining all neighboring components of the small components that contain three or more pixels is more complicated and time-consuming. Because every pixel in the small components is compared with its neighboring pixels during this operation, these small components with area size less than a pre-determined threshold are handled by component merging operation.

The output of this processing operation is an updated $L_C(i, j)$, an updated list of components together with the associated lists of measures that were computed from these components.

6.2.3 Component Merging

The input into this processing step is the image $L_C(i, j)$, the list of components currently found in $L(i, j)$, and the associated of measures that were computed from these components.

As will be seen, the component merging algorithm is much more computationally complex than is the small component elimination algorithm. Since there is no need to differentiate too-thin areas and wane, this algorithm will merge any two components that represent too-thin areas whenever they are either adjacent or close enough (within a distance of one pixel). After two components are merged, the measures of the newly created component have to be computed.

The component merging operation involves the following steps:

1) Mark all components as “not-examined.”

2) Select component R_1, whose area is the smallest among the all "not-examined" components.
3) Find component R_2 that is most similar (i.e., too thin or wane) to R_1 in average range level, among all the components adjacent to R_1. The adjacency of two components is defined as one component with one or more pixels that are next to or one pixel away from the pixels of the other component.
4) If component R_2 is found in Step 3, then merge the two components R_1 and R_2, and subsequently mark component R_1 as "deleted." The merging is actually accomplished by assigning the component label of R_2 to all pixels within R_1 in the segmented image $L_C(i, j)$. Otherwise, mark component R_1 as "examined" since there are no components with which this component can be merged.
5) Repeat Step 2 through Step 4 until all the components have been labeled as being examined.

6.2.4 Additional Measurement Extraction

The input into this processing step is the list of components and the associated list of measures that were computed from these components.

Immediately following the component merging operation some additional measures are computed for the all the components defined in $L_C(i, j)$. Fortunately, these new measures can be derived from the list of component measures described above. This means that one does not have to make any new calculations on a pixel by pixel basic. Therefore the computational complexity of the problem is markedly reduced. The new measures computed are as follows:

- Ratio of height vs. width of minimum inscribing rectangle of the component
- Compactness is, as defined in Chapter 2, the ratio of its perimeter squared to four times its area

These two new measures provide additional information about the geometrical shape of a component.

The output of this processing operation is the list of components together with a newly updated list of measures that were effectively computer from these components.

6.3 High-Level Processing of Laser Images

The goal of high-level laser image processing is to identify wane and thin board defect components. Both wane and thin-board components are thinner than clear wood areas, therefore they appear as darker areas in the laser image. With high quality laser images, these two kinds of defects can be unambiguously identified by a set of measures or features that uniquely describe each defect type. Two rules have been developed for use in the high-level processing module for laser image processing. The first rule (*Is_Wane)* is implemented in the defect detection module and is designed to identify the defect wane. The second rule (*Is_thin_board*) is programmed in the defect detection module and determines if a component is a thin-board area. Each module takes as input a component and its measures that were computed and forwarded to it by the intermediate-level processing. The module then outputs a confidence value as to whether the component is of the defect type that the module was formulated to identify. These rules will be described in Sections 6.3.1 and 6.3.2.

The final decision is made based on the confidence values returned by the two modules. Once every component has been identified, these results are passed on to a post-processing module. The post-processing module merges defects where possible as described in Section 6.3.3, and finally stores identified defects on a defect list. Components that cannot be identified are stored on another list. Components on this list will be analyzed later (see Section 8.5). Figure 6.6 shows the architecture of the laser high-level processing module.

6.3.1 Wane Identification *(The Is_Wane Rule)*

Wane is defined as an area where the outer surface of the log or bark is included in the board. A unique characteristic of wane is that it will almost always appear on or "touch" the outer edges of a board. A second characteristic is that a wane component is usually long and narrow.

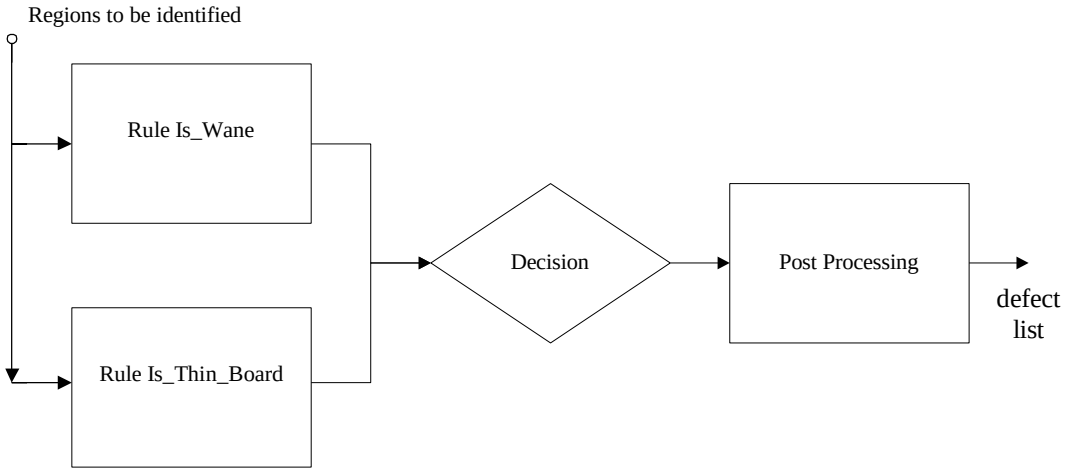


Figure 6.6 Architecture of high-level processing

Besides these two general characteristics wane may have several different manifestations. One manifestation is when no bark is attached to the board. In this instance the thickness of the waney edge of the board gradually becomes thinner as one moves from the center of the board to the waney edge. The laser-range system can unambiguously detect this wane if the thickness is below a desired thickness threshold. A second manifestation of wane is when bark remains attached to the wane area of a board. In this manifestation the wane area oftentimes has the same thickness as that of the board. As such, it is possible that the thickness of the wane component is not below the desired thickness threshold. In such cases it is much more difficult if not impossible to detect these wane areas using the laser image information alone. This difficulty is compounded by the fact that bark is typically very dark. Oftentimes when the bark is dark, the laser line used to create the laser image cannot be detected by the laser-profiling system. A third manifestation of wane is when wane is included across the entire width of the board. In this manifestation, the shape of wane is typically much more compact and does not have its usual long and narrow shape.

Given the above manifestations of wane, a wane defect area in a laser-range image will have the following measures:

- Low range value. Since wane is often thinner than the board on which it appears, pixels in such areas have smaller range values. In fact, these pixels can easily be separated during lower-level processing. However, there exist two exceptions: 1) If bark is present, the laser-range image often cannot detect any thickness variation in the

area; hence the wane is very hard to detect. 2) If bark is attached to the lumber and the bark is very dark, the laser line cannot be detected by the laser-ranging system; hence, the range values in the component cannot be determined. These exception areas will be handled later in the processing when either the X-ray images are analyzed (Chapter 7) or when the color images are analyzed (Chapter 8). In this later processing, information from all images is combined to accurately identify such components. Hence the focus here is on the components that can be unambiguously identified using only the laser image.

- Long and narrow area. Wane usually is a long and narrow area oriented along the length of the board. Figure 6.7 shows a typical waney area found on a hardwood board. In general, the length of wane is observed to be at least several times greater than its width (except the third manifestation of wane).
- Local range-value variation is large since wane is thinner near the board edge and thicker near the interior edge of the waney area. This variation can be measured by calculating the variance of range pixel values (i.e. variance of the pixel gray level values) inside the component. Since variance of range pixel values is relatively computationally complex to determine, it is not used in the current system, but it can be easily added if additional classification accuracy is needed.
- On the lumber edge. Wane generally appears along the edge of a board. This is measured by the componental measure *touch_edge*. Moreover, the component measure *center of mass* can also be used to indicate the location of the wane.

Given the above measures of wane, the rule that was formulated to determine if a component is wane is as follows:

If a component

has range value that is smaller than a thickness threshold, $T_{thickness1}$, and,
has height/width ratio that is greater than a threshold, T_{hw} (e.g., 2.5), and
the *touch_edge* (i.e. on the board edge) is greater than a threshold, T_{touch_edge} (e.g., 0.15);

Figure 6.7 Examples of wane shown in both the laser image and color image

Then

the component is considered to be wane with confidence value C_w of 0.95.

else

the component is considered to be wane with confidence value C_w of 0.05.

The thickness threshold $T_{thickness1}$ is usually the threshold T_{TE} (thick enough) described earlier in section 6.1.2 that is obtained by measuring a calibration board with a known target thickness. This approach can be used since anything thinner that T_{TE} is considered too thin to be used for rough part. The threshold T_{hw} as well as T_{touch_edge} were determined based on empirical measurements taken from a number of samples of waney regions [FOR96]. From Table 6.1, one can see from this sample, the ratio of height vs. width varies from 3.06 to 10.86. The average value of height/width is 7.20, with a standard deviation 3.15. Assuming a normal distribution, the 5th percentile of height/width is found to be 2.02 (i.e. the probability that a height/width is larger than 2.02 is 95 percent). The touch_edge parameter varies from 0.21 to 0.39. It has an average value of 0.30, with a standard deviation of 0.08. The threshold T_{touch_edge} is set to 0.17 in this research according to the 5th percentile [MIL90] calculation of touch_edge. The selection of confidence values C_w is based on the fact that more than 95 percent of the wane samples should have a value of height/width greater than T_{hw} and a value of touch_edge larger than 0.17. Note that these 5th percentile values are smaller than the observed minimum values due to a small number of wane component samples (N=10) available in this research. Choosing the 5th percentile levels dictates what confidence value is assigned to the THEN portion of the rule (e.g. C_w). Setting T_{hw} = 2.02 and T_{touch_edge} = 0.17 based on the 5th percentile level appeared reasonable for the performance of this system. However, if more or less confidence is desired, the selection of percentile values can be adjusted as needed.

Table 6.1 Component measure statistics extracted from a sample of wane components

Component Measures	Average value	Standard deviation	Maximum	Minimum	5^{th} percentile
Height/width	7.20	3.15	10.86	3.06	2.02
Touch_edge	0.30	0.08	0.39	0.21	0.17

This crisply defined IF-THEN rule was found to be adequate for the scope of this research. Similar rule-based approaches have been used successfully by many other researchers for applications such as interpreting airport scenes [MCK85], identifying defects in rough lumber [CHO91], and analyzing biomedical images [PUL93].

Problems in using such a rule have led to erroneously classifying wane as a "thin-board" component if the third manifestation occurs. In this research, however, this is not a critical error since both of these defects are unacceptable to the manufacturer [FOR95].

However, if an application is required distinguishing wane from thin components, a fuzzy logic rule could be formulated to achieve better classification results. The development of more sophisticated fuzzy logic rules will be discussed in detail in Chapter 7. This new fuzzy rule would be able to weigh certain parameters more heavily (e.g. a higher height/width ratio and a higher portion of *touch_edge* gives a stronger indication that a defect is wane) when making a certain classification. Furthermore, the inclusion of more component measures such as the pixel-value variance described earlier should enhance the accuracy of classification.

6.3.2 Thin-Board Identification *(The Is_Thin_board Rule)*

An area of thin board may be defined as any area that has a thickness thinner than a specified threshold. For example, this threshold can be user specified to ensure that a minimum part thickness is maintained. Thin-board components can be present for a variety of reasons. For example, imprecise cutting of logs using inappropriate sawing technology or improperly maintained sawing adjustments are leading causes of thin or "scant" boards. Thin-board components may appear in many shapes [KLI97]. However, thin-board components are generally indicated in a laser-range image as follows:

- Small range pixel values since thin board is thinner than normal lumber,
- Ratio of height/width of a thin-board area is typically smaller than that of wane,
- The local range-level variation is small since the thin board area is typically thin and flat,

- Large area--typical thin board components in a laser image are on the order of several hundred range pixels in an area.

Consequently, the rule formulated for identifying thin board is as follows:

If a component

has gray level that is smaller than a thickness threshold, $T_{thickness1}$,

has height/width ratio that is smaller than a threshold, T_{hw} (e.g., 2.5), and

has a big area that is greater than a threshold, T_{area} (e.g., 100 pixels);

Then

the component is considered to be *thin board* with a confidence value C_t of 0.95,

Else

the component is considered to be *thin board* with a confidence value C_t of 0.05.

The thickness threshold $T_{thickness1}$ is usually the same threshold T_{TE} described in section 6.1.2. The thresholds T_{hw} and T_{area} are selected in a manner similar to the one discussed in 6.2.2. The confidence value C_t is based on the fact that more than 95% of the sample thin boards should meet the above conditions. As was mentioned earlier, it is often not at all crucial to distinguish "thin board" from wane since both defects are considered to be removable defects by manufacturers. If it is necessary to classify wane and thin board more accurately, local range-level variation and additional component measures from X-ray and color images can be used to more accurately classify these components, and/or fuzzy rules could be added.

6.3.3 Remaining Components of Unknown Identity

If a component cannot be identified by either of the two rules, the component is classified as an unknown component. Components with unknown identify are added to an unknown identity list. Components on this list are ones whose defect type cannot be identified from the laser image data only. Components on this "unknown" identity list are usually ones that are either very dark or ones whose thickness values lie outside the dynamic range of the laser-profiling system. Such components generally have a very low value for their average range pixel. Experiments have shown that such components usually contain holes, dark knots, and dark

surface stains that can be identified later using either or both of the X-ray and/or color images. Hence, any component put on the "unknown" identity list is one whose classification is deferred for further analysis later.

6.3.4 Post-Identification Processing

After each component has been analyzed, further defect component merging is sometimes necessary. The reason for this is that wane components are often over-fragmented even after the component merging was done in the lower levels of processing. This over-fragmentation may be caused by either the limitations of the laser-profiling system or by the inherent structure of the board surface. In any event, it makes no sense to report two wane components that are very close to each other since the clear wood that lies between these two defects cannot be used to make any additional rough parts. Therefore, the goal of post-identification processing is to merge components which have been assigned to the same defect type and which are sufficiently close to each other.

Computing the distance between two components can be a computationally complex task since it generally involves searching for the shortest distance between two boundary points, one in each component. For the purpose of this work a simple distance calculation is required, one that still gauges the distance between two components in a reasonable manner. An alternate approach using minimum inscribing rectangles might be an attractive alternative approach. With this approach, the distance between two components is approximated by the distance between two minimum inscribing rectangles. This distance usually is the shortest distance between two corners of the two rectangles, with one corner on each rectangle (see Figure 6.8).

The minimum inscribing rectangle of a component is defined by its upper left (i_{UL}, j_{UL}) and lower right-(i_{LR}, j_{LR}) hand corners. Since two edges of this rectangle must always be parallel to the image axis I, the distance between the two rectangles in the down-board direction (coordinate I) can be calculated as:

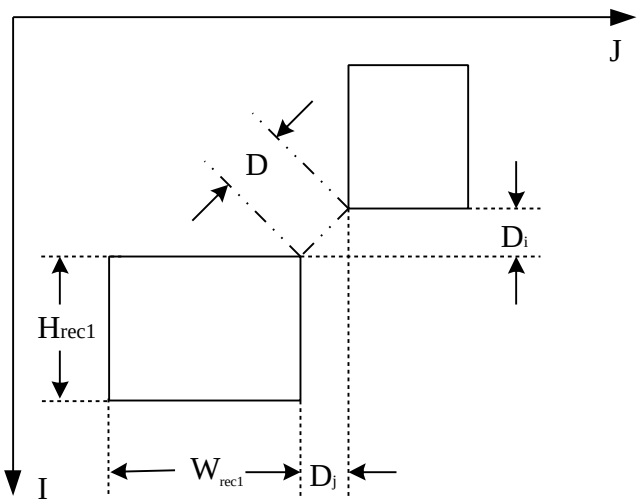


Figure 6.8 Distance (D) between two rectangle with no overlaps in any direction

$$D_i = \begin{cases} 0 & \text{if } \left|i_{ctr1} - i_{ctr2}\right| \le \frac{1}{2}(H_{rec1} + H_{rec2}) \\ \left|i_{ctr1} - i_{ctr2}\right| - \frac{1}{2}(H_{rec1} + H_{rec2}) & \text{otherwise} \end{cases} \quad (6.2)$$

where $i_{UL1}$and i_{LR1} are the I-coordinates of the upper left corner and lower right-hand corners, respectively, of the first rectangle; i_{UL2} and i_{LR2} are these I-coordinates of the upper left and lower right-hand corners, respectively, of the second rectangle; i_{ctr1}= ½(i_{UL1}+ i_{LR1}), i_{ctr2}= ½(i_{UL2}+ i_{LR2}); H_{rec1} is the height of the first rectangle (see Figure 6.8) and H_{rec2} is the height of the second rectangle. If D_i given by Equation 6.2 is 0, the two rectangles are overlapped in the down-board direction. If (i_{ctr1} - i_{ctr2}) = ½ (H_{rec1} + H_{rec2}), the two edges, each close to the other in the different rectangles, are overlapped edges (see Figure 6.9). If (i_{ctr1} - i_{ctr2}) < ½ (H_{rec1} + H_{rec2}), the two rectangles would overlap even more, and the distance between them should be set to 0.

Similarly, the distance in the cross-board direction is

$$D_j = \begin{cases} 0 & \text{if } \left|j_{ctr1} - j_{ctr2}\right| \le \frac{1}{2}(W_{rec1} + W_{rec2}) \\ \left|j_{ctr1} - j_{ctr2}\right| - \frac{1}{2}(W_{rec1} + W_{rec2}) & \text{otherwise} \end{cases} \quad (6.3)$$

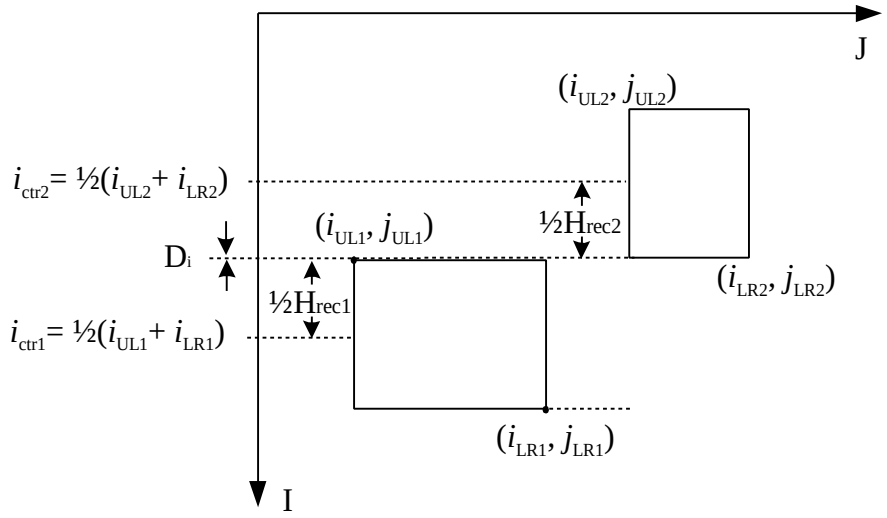


Figure 6.9 Distance (D_i) becomes 0 if the two edges are overlapped in I direction

where $j_{UL1}$and j_{LR2} are the I-coordinates of the upper left corner and lower right-hand corners, respectively, of the first rectangle; j_{UL2} and j_{LR2} are these I-coordinates of the upper left and lower right-hand corners, respectively, of the second rectangle; and $j_{ctr1}= ½(j_{UL1}+ j_{LR1})$, $j_{ctr2}= ½(j_{UL2}+ j_{LR2})$, W_{rec1} and W_{rec2} are the heights of two minimum inscribing rectangles, respectively. If D_j is less than or equal to 0, then the two rectangles are overlapped in the cross-board direction. The two rectangles are considered overlapped only if they are overlapped in both directions.

The distance of the two rectangles is then given by

$$D = \sqrt{D_i^2 + D_j^2} \tag{6.4}$$

If the two rectangles are overlapped in the I-direction only ($D_i = 0$), the distance between them is D_j. Similarly, if they are overlapped in the J-direction only ($D_j = 0$), the distance between them is D_i. If there is no overlap in any direction, the distance between the two rectangles becomes the shortest distance between two corners of rectangles (see Figure 6.9).

If the distance between two minimum inscribing rectangles is smaller than a threshold (for example, three pixels), the two components are merged to form a larger component provided the components have been assigned to the same defect class. This threshold is a variable that is stored a configuration file. This variable can be modified based on the requirements of the manufacturer.

The merging of two components forms a new component whose upper left corner (i_{UL}, j_{UL}) and lower right corner (i_{LR}, j_{LR}) of a minimum inscribing rectangle are given by:

i_{UL} = minimum { i_{UL1}, i_{UL2} }; j_{UL} = minimum { j_{UL1}, j_{UL2} }

i_{LR} = maximum { i_{LR1}, i_{LR2} }; j_{LR} = maximum { j_{LR1}, j_{LR2} }

Where (i_{UL1}, j_{UL1}), (i_{LR1}, j_{LR1}), (i_{UL2}, j_{UL2}) and (i_{LR2}, j_{LR2}) are the coordinates of the upper left corner and lower right corner of the two minimum inscribing rectangles of the two components, respectively. This new minimum inscribing rectangle will enclose the two defect components since it encloses every minimum inscribing rectangle of the two components. This merging procedure assumes that weights on both the I and J direction distance are equal. However, this procedure can be easily adapted to accommodate different weights for the two directions.

The algorithm to merge defect components that are close enough is as follows:

1) Select a component, say R_1 from the list of components to be processed (all identified defect components are put on the list initially).
2) For each component (other than R_1) on the list, find the distance between R_1 and that component, say R_2, using Equation 6.4. If the distance between the two is less than the given threshold, merge the two components and mark the component R_2 removed from the list. Update the component measures of the new component after the merge is done.
3) If R_1 cannot be merged with any component, remove it from the list.
4) Repeat step 1 until no more than one component is left on the list

Once every defect component is examined and processed, the resultant defect components are put on a final defect list. This final defect list and the list of defects that cannot be determined with laser image only are passed to the next module--analysis of X-ray images (Chapter 7).

6.4 Processing Results

The laser images are thresholded with a pre-determined global threshold, $T_{TE,}$ as defined in Section 6.1. The threshold T_{TE} is determined by averaging the range levels of the laser image of a

flat board with a desired target thickness. Anything that is thinner than this board is considered too thin. Threshold T_{TU} for *unknown* pixels was determined from experimental images and was chosen to be 10 since the values of unknown pixels are typically set to a low value of approximately 5 by the laser-profiling system. Two laser images and their thresholded images are shown in Figure 6.10. The board in the image on the left has several waney areas along its right edge approximately halfway down the board. The board in the image on the right has a wane area at one of its ends, the one at the top of the image. All the wane areas appear darker as compared to thick-enough areas in the thresholded images.

Connected-component labeling is then applied to these thresholded images. Initial components are formed by this operation. Figure 6.11 shows the results obtained by applying the connected-component labeling operation to the two board images of Figure 6.10. As one can see, there are many small and meaningless components generated by the connected-component labeling operation. The different connected components are shown as various shades of pink. The shade level is determined by the value of the component label on a connected component. If the label of clear wood (determined during the above operation) is very small, the clear wood would look darker than other components or the clear wood of another board [Figure 6.11 (d)] in labeled images.

These small noise-induced components need to be eliminated before component measures are extracted from each of the components. This is accomplished by the operation of a small component deletion as discussed in Section 6.1 (currently, only one- or two-pixel components are eliminated because anything larger might be part of an over-fragmented defect. Eliminating these small components reduces the processing time required to do component merging and defect recognition. For image (a) of Figure 6.11, the connected-component labeling operation detected total 64 connected components. The distribution of a connected-component size (area) is illustrated in Figure 6.12. Note that of these 64 components, 35 components contain just one or two pixels. Eliminating these small components leaves only 29 components to consider.

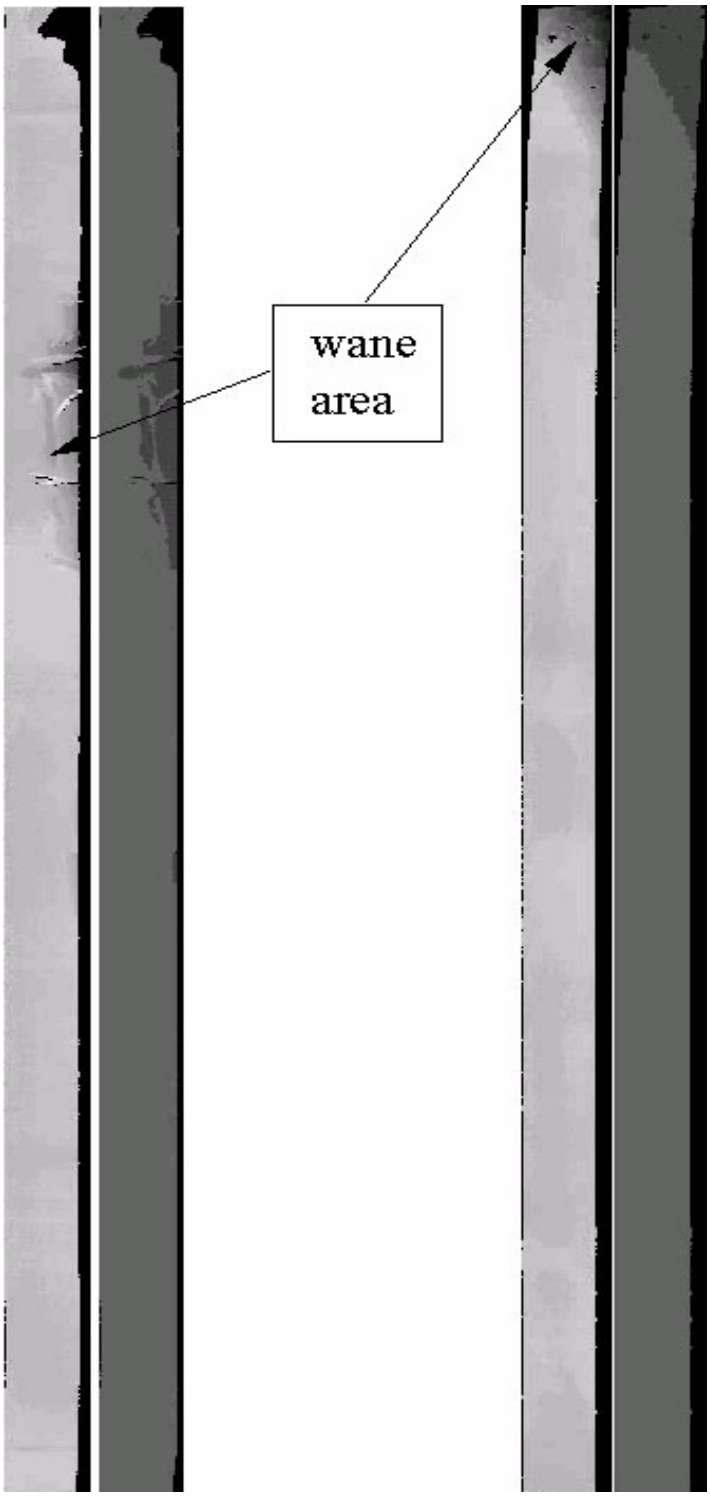


Figure 6.10 Laser images (left image in each pair) and their thresholded images.

Figure 6.11 Original laser images (left in each pair) and their labeled images

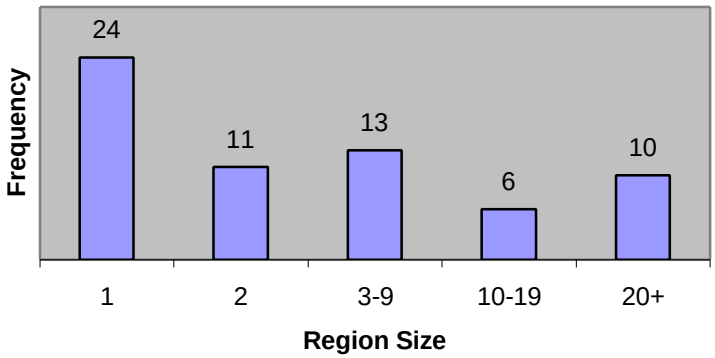


Figure 6.12 Size distribution of initial components

After the small connected components are eliminated, the component merging operation is performed to combine similar adjacent components (e.g., components that are within one pixel of each other). This operation further reduces the total number of connected components from 29 down to 26, as is shown in Figure 6.13. One of the remaining 26 components represents the whole board.

Figure 6.14 shows the processing results after the small component elimination and the component merging operations have been performed. The component merging process ensures that over-fragmented components are now recombined. As one can see from Figure 6.13 and Figure 6.14, even after component merging there are still some small connected components (e.g., components only containing three or four pixels). It is believed that these small components are caused either by noise or by image quality. Since the major purpose of laser-image analysis is to

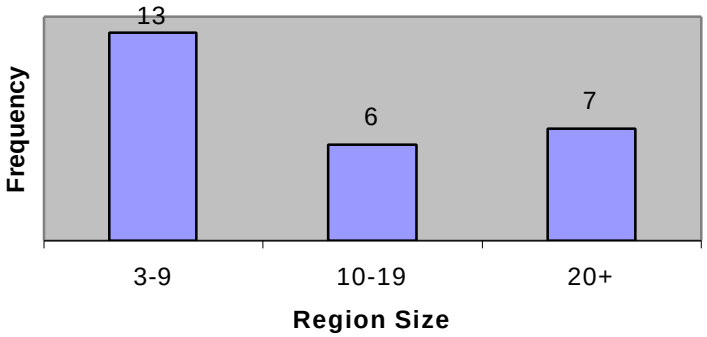


Figure 6.13 Connected-component size distribution after the small component elimination and the component merging operations have been performed

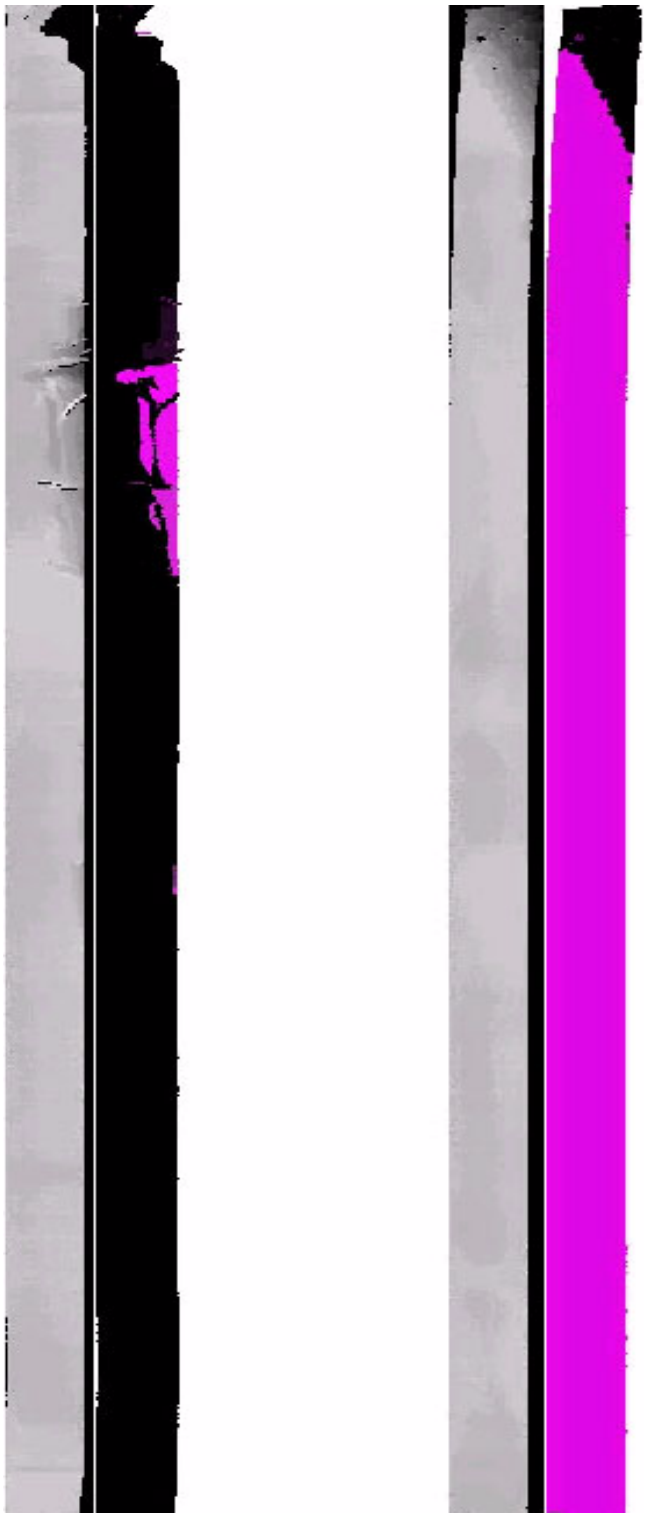


Figure 6.14 Small components still exist after small component elimination and component merging

identify defects such as wane and thin board, defects that are generally fairly large, components with areas less than a preset threshold T_A are eliminated. The threshold T_A was chosen to be 9 based on requirements provided by the manufacturer [FOR96]. However, T_A was implemented in a way that makes it a variable that can be easily adjustable. The idea is to allow the value of this variable to depend on the smallest defect size a manufacturer is willing to tolerate. Because defects in the wane class generally have large areas, several hundreds of pixels, a relatively large value for T_A is not unreasonable.

After this last small connect component elimination process, component measures are extracted for each of the remaining connected components. These component measures (see Table 6.2) include area, minimum inscribing rectangle (MIR, described by the coordinates of upper left corner and lower right corner of the rectangle), average range level (ARL), length of the perimeter of the connected component (PERIM), center of the mass (CMI CMJ, coordinates of the center of mass), touch_edge (TE), and compactness (COMP). After the component measures have been extracted, the remaining connected components and their associated component measures are forwarded to the high-level processing module that classifies these components.

Table 6.2 Components and their measures forwarded to high-level processing

Component Index	Area size	ARL	MIR	CMI	CMJ	Comp	Perim	TE
1	90209	185.05	(0, 0) (1746, 57)	0	0	0	0	0
2	11	163.82	(20, 0) (23, 2)	0.9	21.4	1.42	14	0.29
3	16	62.06	(24, 31) (32, 36)	31.3	30.6	4.48	30	0.07
4	1020	137.87	(348, 30) (417, 59)	386	46.2	6.65	292	0.23
5	29	0	(416, 35) (420, 49)	418.4	41.1	4.39	40	0
6	953	134.16	(416, 13) (461, 55)	434.6	39.3	5.91	266	0.13
7	1890	147.68	(454, 28) (558, 55)	505.7	42.8	9.46	474	0.21
8	18	164.17	(462, 39) (470, 43)	464.9	40.6	3.47	28	0
9	19	0	(552, 20) (554, 31)	552.7	25.9	3.77	30	0
10	12	0	(560, 39) (561, 49)	560.9	43.8	4.48	26	0
11	997	145.21	(561, 35) (661, 55)	599.6	48.4	10.58	364	0.28
12	76	114.86	(996, 51) (1019, 54)	1007.3	52.9	3.77	60	0.4
13	21	113.33	(1023, 52) (1029, 54)	1026	53	1.52	20	0.35

During high-level processing, each of the components found by the intermediate-level processing and its associated measures are sent to the two defect identifying modules (Section 6.5). Each module returns a confidence value for the component, with one module returning the confidence value of the component as wane while the other returns the confidence that the component is thin board. The results of defect identification are shown in the Table 6.3.

The decision-making module determines what final label will be assigned to the component. Before each component and its label are added to the final defect list, it is checked to see if it is close to any other component that has the same defect label. If it is, the two components are merged. Table 6.4 shows the final results obtained from the laser vision module for the board whose laser image is shown in Figure 6.11(b).

Table 6.3 Confidence values on wane/thin-board for each component

Index	Area	Avg. Range	Min Inscribing Rect.	CV on wane	CV on thin board	Comments
1	90209	185.05	(0, 0) (1746, 57)	0	0	This is the whole board
2	11	163.82	(20, 0) (23, 2)	0.05	0.05	Clear wood
3	16	62.06	(24, 31) (32, 36)	0.95	0.05	Wane
4	1020	137.87	(348, 30) (417, 59)	0.95	0.05	Wane
5	29	0	(416, 35) (420, 49)	0	0	Cannot be determined
6	953	134.16	(416, 13) (461, 55)	0.95	0.05	Wane
7	1890	147.68	(454, 28) (558, 55)	0.95	0.05	Wane
8	18	164.17	(462, 39) (470, 43)	0.05	0.05	Clear wood
9	19	0	(552, 20) (554, 31)	0	0	Cannot be determined
10	12	0	(560, 39) (561, 49)	0	0	Cannot be determined
11	997	145.21	(561, 35) (661, 55)	0.05	0.95	Thin board
12	76	114.86	(996, 51) (1019, 54)	0.95	0.05	Wane
13	21	113.33	(1023, 52) (1029, 54)	0.95	0.05	Wane

Table 6.4 Final results of laser image module

Index	Min Inscribing Rect.	wane	Thin board	Comments
1	(31, 24) (32, 36)	y		
2	(348, 30) (417, 59)	y		
3	(416, 35) (420, 49)			Cannot be determined
4	(416, 13) (558, 55)	y		Formed by 6 and 7 in Table 6.2
5	(552, 20) (554, 31)			Cannot be determined
6	(560, 39) (561, 49)			Cannot be determined
7	(561, 35) (661, 55)		y	
8	(996, 51) (1029, 54)	y		Formed by 12 and 13 in Table 6.2

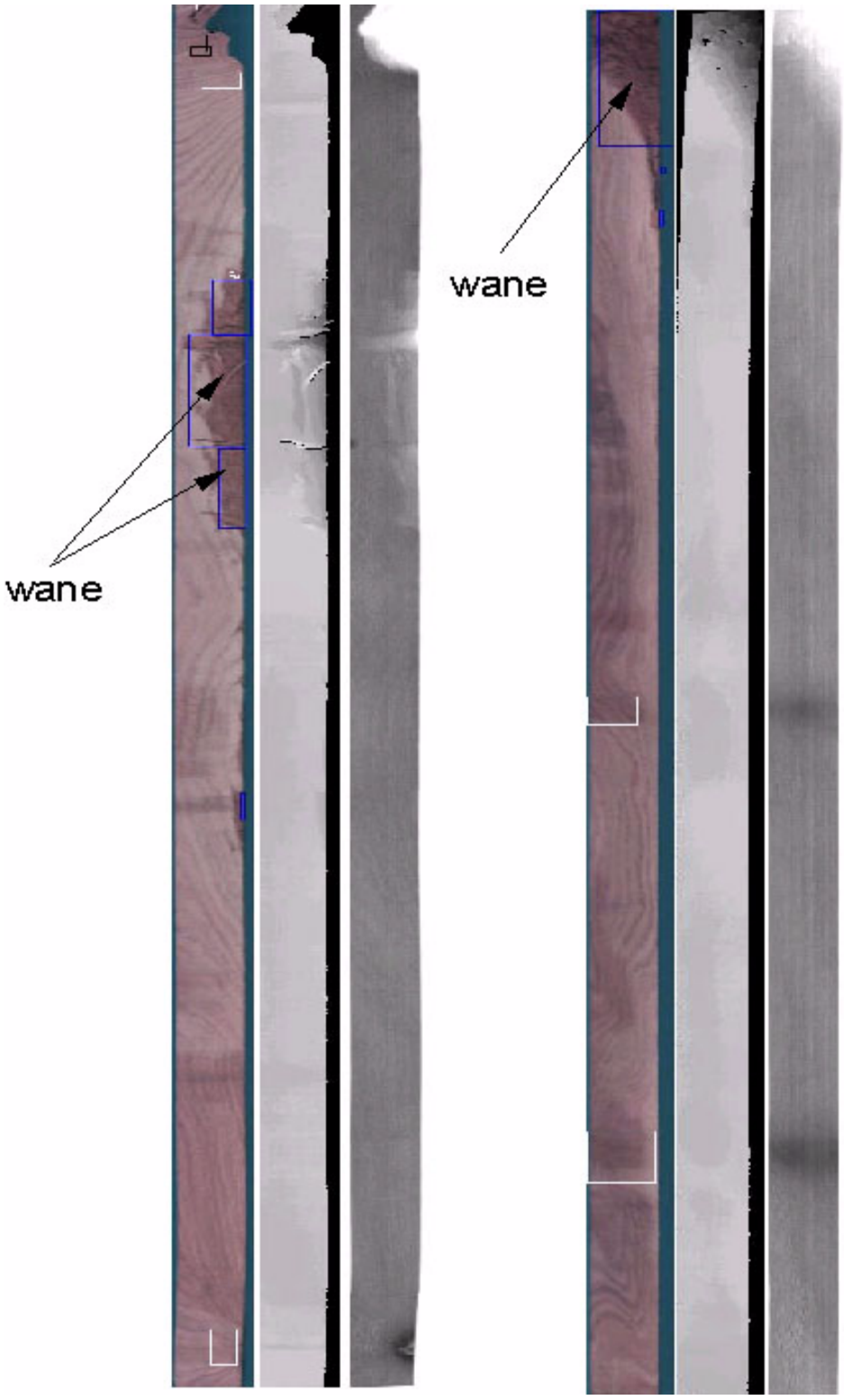


Figure 6.15. The final identification results--left image in each group is the color image, the middle one is the laser image, and the right image is the X-ray image. The areas within blue rectangles are wane. The areas within white rectangles are high-density components (identified in next two chapters).

The final processing results for the two boards whose laser images are shown in Figure 6.11 are given in Figure 6.15. In this figure, there are two groups of three images. The group on the left represents the results discussed above. In this example, major wane areas (components 4, 6, 7, 12 and 13 in Table 6.3) and thin board (component 11 in Table 6.3) have been identified correctly. Although three of the components could not be identified by the laser vision software, this was due to limitations in the laser imaging systems. As will be seen in later discussions these two reasons are finally correctly identified by the system when the analysis of the color image is performed (see Section 8.5).

6.5 Summary

The analysis of laser image is aimed at detecting wane and thin-board defects. The analysis consists of low-level, intermediate-level, high-level, and a post-identification processing. The goal of the low-level processing or segmentation is to find image components that contain possible defects. The image segmentation is accomplished by a global thresholding operation followed by a connected-component labeling procedure. An algorithm for connected-component labeling was developed in this research to handle thresholded images that have multiple gray level values. The outputs of a low-level processing module are then fed into an intermediate-level processing module, where components that contain possible defects are found and measures that characterize each of these components are computed. In this intermediate-level processing, small components are eliminated and similar components are merged. The final components and their associated measures are then sent to a high-level processing module that gives each component a defect label and a confidence measure about the label or labels that have been assigned. The post-identification processing merges those components that are adjacent to each other and have the same defect label. Finally, all components which have been assigned a defect label are put on a final defect list, and components that cannot be identified based on laser information alone are put on a list of components to be analyzed later using either X-ray information and/or color information. Both lists are then passed to the module performing analysis of X-ray images.

Chapter 7

Analysis of X-ray Images

The X-ray image is mainly utilized to identify defects such as knots, holes, mineral streak, internal voids, splits, and foreign metallic objects based on the density variation of these defects from that of clear wood. The X-ray images used in this research are 8-bit (256 gray-level) images that sense the transmission values of X-rays passing through a board. Common defects such as knots and mineral streak manifest themselves as higher density areas (compared to clear wood), within which the X-ray image is darker because less X-ray energy can penetrate these areas. Other common lumber defects such as holes, splits, internal voids and decay have a lower density than clear wood. This allows more X-ray energy to transmit through these areas, which results in these areas appearing lighter than clear wood. The background of the X-ray image is often very bright (close to the maximum image value, e.g., 255) due to air absorbing very little X-ray energy. X-ray imaging has been widely used to detect high-density and low-density defects in logs [FUN85, ZHU93]. Applications of X-ray imaging to lumber inspection have been suggested and studied by several researchers [CON93, KLI96, AGU96].

Broadly speaking, the processing of the X-ray image involves image preprocessing, i.e., shade correction and the processing that was described in Chapters 4, and the actual analysis of the image. All the preprocessing operations are performed by the either the MORRPH board or the MORRPH board working in conjunction with the data collection software. While the algorithms employed by the components are not part of this research they will be described here for completeness. The preprocessing transforms the X-ray image, $X_{RAW}(i, j)$, that comes directly from the X-ray imaging system to the X-ray image, $X(i, j)$, that has had most of the background removed and is nearly perfectly registered with the color image, $C(i, j)$, and the laser image, $L(i, j)$.

The scene analysis consists of two stages. Stage 1 aims to identify large defect areas such as big knots, mineral streak, big holes and splits. The objective of Stage 2 is to identify small defects after the effect of the large defect areas is removed from the histogram of the image. This second stage can be skipped if smaller defects (e.g., whose area sizes are less than 16 pixels) may be neglected. Similar to the processing of laser-range images (see Chapter 6), each stage of the X-ray image processing consists of three processing levels: low-level processing, intermediate-level processing, and high-level processing. Both of these processing stages utilize the exact same algorithms. Obviously this two-stage processing of X-ray images takes more time. But this two-stage processing allows smaller critical defects such as small knots and fine splits to be identified.

The low-level processing employs a dynamic histogram-based image thresholding segmentation procedure [CHO91, CHO93] followed by a connected-component labeling operation. Due to the complex variations that can occur in hardwood density, a more sophisticated thresholding procedure is needed to process the X-ray images than that discussed in Chapter 6 for laser-image processing. The intermediate-level processing attempts to characterize the image components found in the low-level processing by extracting measures that can be used to describe each of these components. The high-level processing of the X-ray image identifies these components and determines which defect label to assign to each or whether they should be assigned the clear wood label.

The high-level processing stage can extract information from both the color image, $C(i, j)$, and the laser image, $L(i, j)$, when it is needed to determine the correct label to assign to a component. By cross-referencing multiple images, more component measures can be extracted dynamically when they are needed and fused with measures obtained from the X-ray image.

What the above cross-referencing of multiple images means is that input to this analysis module includes not only $X(i, j)$ but $C(i, j)$ and $L(i, j)$ as well. It also means that this module is able to assign a label to any component that it detects. That is to say, unlike the analysis of the laser image where certain components may need to be passed on to the color image analysis module, all the processing for components found in the X-ray image are labeled in the analysis

described here. While it is true that one could have designed an X-ray image processing module that would pass components that could not be accurately labeled based on information in only the laser image and X-ray images onto the color image processing module, it seemed more expedient to compile all necessary information, i.e., include information from the color image, and assign the labels in this module. The strategy was motivated, in part, because $L(i, j)$, $X(i, j)$ and $C(i, j)$ are all nearly perfectly registered and so gathering all the information together in this module was a straightforward thing to do.

The multiple-sensor fusion method developed to combine all the information from these various sensors differs from traditional sensor fusion methods in that it dynamically fuses information from other sensors only when such fusion is required. The advantage of this approach is that it reduces processing time while providing all the benefits of using multiple sensors.

The objective of this chapter is to provide a detailed description of the algorithms used in analysis of X-rays images and to demonstrate the processing results obtained from using these algorithms.

Finally, just as was the case in Chapter 6, the low-level vision algorithms described in this chapter are only applied to transmission elements that pass through the board. This application is possible because of the processing that was described in Chapter 5. This processing computes the values for *Board_Start*, *Board_End*, and *Board_Edge*. The algorithm for using *Board_Start*, *Board_End*, and *Board_Edge* to find only transmission values that pass through the board is straightforward. It is straightforward since the color image is nearly perfectly registered to the X-ray image. Hence, this algorithm will not be discussed in this chapter.

7.1 The Preprocessing of X-ray Images

There are only two preprocessing operations performed on the X-ray image $X(i, j)$. One of these operations is shade correction. The input to the shade correction algorithm after certain constants have been defined is the image $X_{RAW}(i, j)$. The output of the this preprocessing is the

unregistered X-ray image $X_U(i, j)$. The unregistered X-ray image is then input to the preprocessing operation that is described in Chapter 4. The goal of this processing is to create a registered X-ray image $X(i, j)$.

7.1.1 Shade Correction

The first preprocessing operation that is performed on the X-ray image, $X_{RAW}(i, j)$, is image shading correction [SAW77]. Shading correction is needed in order to remove any non-uniformities in the X-ray energy or differences in sensitivity among the individual X-ray detectors.

Since from a theoretical point of view the X-ray imaging system behaves just like a black and white camera, and since the development of the shading correction algorithm was developed for a black and white camera, this type of development will be given below. Using the black and white camera as the method for development will prove useful then shade correction for color images is described in the next chapter.

In the development of the shading correction algorithm the camera output (image data) is modeled as a nonlinear function of camera input:

$$g(x, y) = N(f(x, y)) \tag{7.1}$$

where $g(x, y)$ represents the camera output to a given input $f(x, y)$, (x, y) denotes the pixel position, and N denotes the nonlinear function. The input $f(x, y)$ may be viewed as the product of a reflectance component $r(x, y)$ and an illumination component $i(x, y)$.

Since line scan cameras are used in the proposed vision system, equation (7.1) reduces to

$$g(x) = N(f(x)) \tag{7.2}$$

where x denotes pixel position within an image frame. Thus $g(x)$ can be represented as a Taylor's series expansion at a fixed input value $f_0(x)$:

$$g(x) = N(f_0(x)) + (f(x) - f_0(x))\frac{\partial N}{\partial f}\bigg|_{f_0(x)} + \frac{(f(x) - f_0(x))^2}{2!}\frac{\partial^2 N}{\partial f^2}\bigg|_{f_0(x)} + \ldots$$

Collecting all the powers of $f(x)$ yields

$$g(x) = a_0(x) + a_1(x) f(x) + a_2(x) f^2(x) + \ldots \quad (7.3)$$

where $a_0(x)$, $a_1(x)$, $a_2(x)$ …, are constant coefficients.

For most high quality imaging sensors, the higher order terms of f ($f^2(x)$ and above) in equation (7.3) can be neglected [SAW77]. Hence the camera input $f(x)$ can be approximated by

$$\hat{f}(x) = \frac{g(x) - a_0(x)}{a_1(x)} \quad (7.4)$$

which gives the value of the shade corrected image at location x.

However, in equation (7.4), both $a_0(x)$ and $a_1(x)$ are constant coefficients that need to be determined experimentally.

The coefficient $a_0(x)$ and $a_1(x)$ can be determined by two known values of camera input $f(x)$ in the equation (7.3). In practice, $f(x) = 0$ and $f(x) = 255$ (for all values of x) are used since $f(x) = 0$ represents a uniform black surface, and $f(x) = 255$ represents a uniform white surface. These two values also represent the minimum and maximum pixel values that can appear in any of the shade corrected images. The brightness of the uniform white surface determines the brightest possible object that can be imaged by the camera when the shading correction is used. When $f(x) = 0$, equation (7.4) reduces to

$$g(x) = a_0(x) = g_{black}(x) \quad (7.5)$$

where $g_{black}(x)$ is the camera output for a black uniform surface. This completely black uniform surface is obtained by placing a cover on the camera lens and collecting image data while the lens is covered.

For an X-ray imaging system the completely white target corresponds to reading the values of the X-ray detectors when the X-ray source is on and when there is nothing between the source and the detectors. Finding the completely uniform black is a bit more complicated. One must find a uniform target, i.e., one that is made of solid uniform material, i.e., metal or plastic, and which is of uniform thickness, whose absorption is slightly greater than the highest absorption of any object that will be imaged by the system. For this work the target is placed in the same

location as the boards are when they are imaged. The target that is used for this system is a 45 mm thick piece of polyethylene that has a uniform density of 0.94 g/cm^3.

For $f(x)$= 255, and neglecting the higher order terms of f(x), one can get

$$a_1(x) = \frac{g(x) - a_0(x)}{255} = \frac{g_{white}(x) - a_0(x)}{255} \tag{7.6}$$

where $g_{white}(x)$ is the camera output when uniform white surface is used as input. The shading corrected image $\hat{f}(x)$ is then given by

$$\hat{f}(x) = \frac{g(x) - g_{black}(x)}{g_{white}(x) - g_{black}(x)} \cdot 255, \text{ for all x} \tag{7.7}$$

To apply the shading correction algorithm to an input image, two images, $g_{black}(x)$ and $g_{white}(x)$, need to have been first collected and stored. The $g_{white}(x)$ must be collected under the same illumination conditions as is used when the camera scans new images.

The shading correction operation must be performed on the entire image on a pixel-by-pixel basis, which is computationally demanding. For this vision system this computationally complex operation is implemented in the MORRPH [DRY95] real-time collection hardware.

The output of this preprocessing step is given in the unregistered X-ray image, $X_U(i, j)$. This image is then input into the processing algorithms that were described in Chapter 4. The result is the registered X-ray image, $X(i, j)$. This image along with the registered laser image, $L(i, j)$, and the registered color image, $C(i, j)$ will be used as inputs to the analysis software that is described below.

7.2 The Low-Level Processing of X-ray Images

The purpose of the low-level processing is to extract image components that might contain wood defects. As was discussed in Chapter 6, a component-based segmentation approach (e.g., image thresholding followed by a connected-component labeling operation as discussed in

Sections 6.2.2 and 6.2.3) was selected for the low-level processing of laser images because of their simplicity and the fact that these algorithms produce good results. Basically these same algorithms are used to process X-ray images. The only difference is that a more complicated segmentation procedure is used on these images than was employed on the laser images. One other difference is that before the low-level processing is performed on an X-ray image, defect areas previously identified during laser-image processing are removed from consideration in the processing of the X-ray images.

7.2.1 Removal of Defects Identified in the Laser Image Processing

Before processing the X-ray image of a board, defect regions that were found during the analysis of the laser image are flagged so as to remove these areas from further consideration. Removing these previously identified defect regions not only improves the processing results of the X-ray image but also reduces processing time. The two wood defects that can be identified based on the laser image data alone are wane and thin-board areas. These defects both have a lower density than clear wood. If these low-density defects are removed from the histogram of the X-ray image before dynamic threshold selection, a better threshold for lower density areas can generally be obtained. Studies have shown that the transmission levels of wane and thin-board areas often overlap with the transmission levels of other lower density areas such as holes and splits/check. Hence, the inclusion of large areas of wane and thin board areas can completely obscure the effects of the smaller defects on the transmission level histogram. Removal of already known defects such as wane and thin-board areas increases the ability to separate other critical low-density defects in the histogram.

A simple algorithm was developed to remove the defect areas found in the laser image processing from further processing consideration. This algorithm first removes the effects of these defects on the transmission histogram, H_{xt}, of the X-ray image, $X(i, j)$. To do so, a transmission level histogram, H_{xd}, of the defect regions and a transmission level histogram, $H_{xt,}$ of the X-ray image (board only) are extracted. The transmission level histogram H_{xd} is computed from all the pixels in every previously identified defect region, where each defect region includes a

defect and the part of its surrounding area that is contained inside the minimum inscribing rectangle of that defect. While the coordinates of the minimum inscribing rectangle were found during laser-image processing, the same coordinates apply to the X-ray image because all images have been registered to a common coordinate system (see Section 4.1). The new transmission level histogram, H_x, with these defect areas removed, is computed using $H_x = H_{xt} - H_{xd}$ where both H_{xt} and H_{xd} are one-dimensional arrays that have the same dimension (256).

Next, for each pixel *(i, j)* within each defect region in the X-ray image, $X(i, j)$, the corresponding pixel in the thresholded X-ray image, $X_T(i, j)$, is marked with a special value (–1). This is done before the connected-component labeling operation is performed. In the thresholded X-ray image $X_T(i, j)$, the value 0 is used to represent background, any values greater than 0 are used to represent foreground objects. The value -1 is used to identify pixels in the removed regions.

7.2.2 Threshold Selection

The X-ray thresholding operation classifies all pixels into one of three classes: clear wood areas, high-density areas and low-density areas. It is important to determine near-optimum thresholds, that is ones that will do the best job separating pixels in defect regions from pixels in clear wood regions.

For the quality of lumber that is typically used by secondary manufacturers, the following assumptions are generally valid:

1. The total area of defects is a relatively small percentage, typically less than 10%, of the total board surface area.
2. The “transmission levels” of pixels from both clear wood and defect regions are normally distributed.

Under the above assumptions, a typical transmission level histogram of an X-ray image of a board often should have one big peak that represents the clear wood area, and possibly two small peaks, one on each side of the big peak. The small peak whose average transmission level is

less than that of the primary peak usually represents areas that have a higher density than clear wood. While the other peak whose average transmission level is greater than that of the primary peak represents areas that have a lower density than clear wood (see Figure 7.1).

One or more valley points or inflection points should exist between two adjacent peaks. However, in practice, the transmission levels of the clear wood area, the transmission levels of the high-density defects, and the transmission values of the low-density defects are not always that clearly separated. These valley or inflection points can be detected. Once detected they can be used as thresholds to separate pixels that represent areas of clear wood with normal density from pixels that might be from high- or low-density defects. Obviously this use of valley or inflection points means that these thresholds are selected dynamically. Dynamic threshold selection is needed because wood density varies significantly between boards. Hence, the critical step in the threshold selection is to accurately find valley and inflection points adjacent to the main peak of the histogram that separate clear wood from defects.

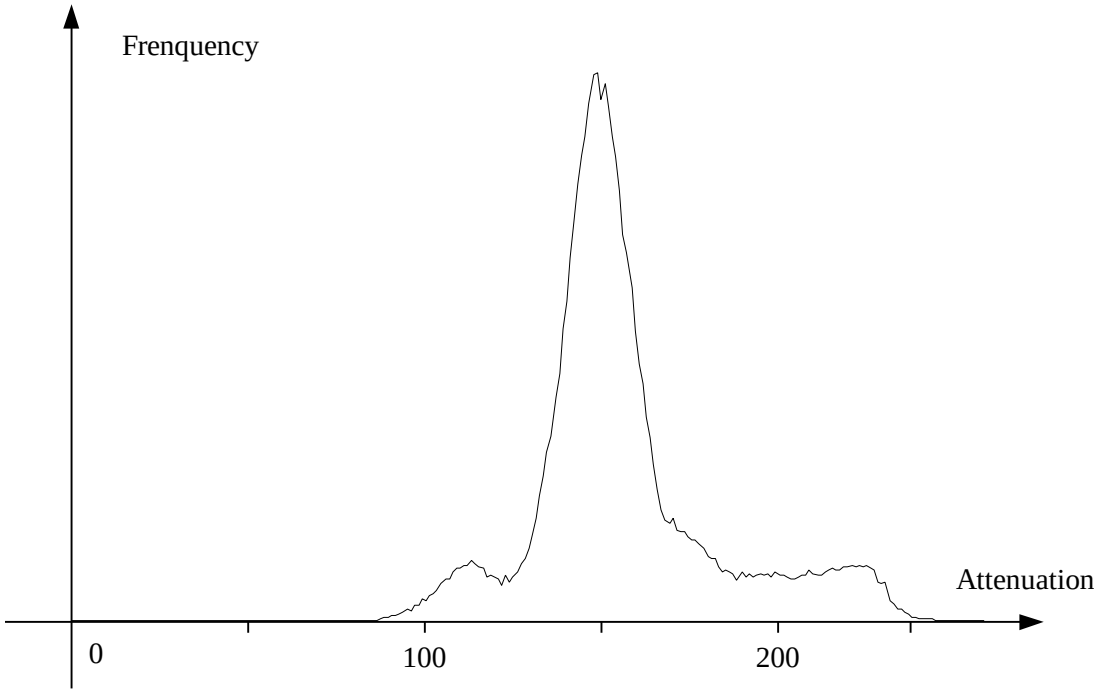


Figure 7.1 Histogram of a typical X-ray image where the small peak on the left represents areas of higher density and the small peak on the right represents areas of lower density

To find the valley and inflection points, the basic algorithm developed by Cho [CHO91] is used. However, some minor modifications had to be made to the algorithm to achieve satisfactory results for lumber defect detection. The revised algorithm consists of the following steps:

1) Find conspicuous or deep valley points in the histogram $H_x(k)$ of the X-ray image and choose these valley points as thresholds, $t_1, t_2, \ldots t_n$. A conspicuous or deep valley point is defined as a clearly identifiable local minimum in the histogram. Actually, the algorithm that finds these valley points also finds the peaks. These peak positions are also used at a later stage of the processing (See Section 7.3.2.). Set *peak_position* = 0, *valley_position* = 0, *peak_counter* = 0, *valley_counter* = 0 and $k = 1$. Also set *direction* = *rise* to indicate that the direction of the slope of the histogram at $H_x(0)$ should be rising. Then perform the following steps:
   a.) If *direction* = *falling* go to Step b. Compare $H_x(k)$ to H_x(*peak_position*). If $H_x(k) > H_x$(*peak_position*), then let *peak_position* = *k*, $p_{peak\_counter}$ = *peak_position*, *peak_counter* = *peak_counter* + 1, and go to Step c. Else if $H_x(k) < H_x$(*peak_position*), and (*k* – *peak_position*) is less than or equal to a threshold *MIND*, e.g., 5, then go to Step c. The objective here is to eliminate false peaks caused by noise. Else if $H_x(k) < H_x$(*peak-position*), and (*k* – *peak_position*) is larger than *MIND*, then let *direction* = *falling*, *valley_position* = *k* and go to Step c.
   b.) Compare $H_x(k)$ to H_x(*valley_position*). If $H_x(k) \leq H_x$(*valley_position*), then set *valley_position* = *k*, $t_{valley\_counter}$ = *valley_position*, *valley_counter* = *valley_counter* + 1 and go to Step c. Else if $H_x(k) > H_x$(*valley_position*) and *k* – *valley_position* is less than or equal to *MIND*, then go to Step c. Else if $H_x(k) > H_x$(*valley_position*) and *k* – *valley_position* is greater than *MIND*, then set *direction* = *rising*, *peak_position* = *k*, and go to Step c.
   c.) If $k = 255$, then set $t_0 = 0$ and *n* = *valley_counter*; if *valley_position* ≠ 255 set *n* = *valley_counter and* $t_{valley\_counter + 1} = 255$ and terminate execution of this algorithm. Else set $k = k + 1$ and go to Step a.
2) For each *i*, $i = 0, 1, 2, \ldots, n$, compute

$$A_i = \sum_{k=t_i}^{t_{i+1}} H_x(k)$$

where $t_0 = 0$, the minimum possible transmission value in H_x, and $t_{n+1} = 255$, the maximum possible transmission value in the histogram H_x.

3) Find the interval $[t_i, t_{i+1}]$ that has the largest value of A_i for $i = 0, \ldots n$. Note this interval should contain the largest peak in $H_x(k)$. A large peak is defined to be one where its A_i value is at least M percent of the total image histogram area, i.e., M percent of

$$\sum_{j=0}^{n} A_j \ .$$

The value of M is derived by experimentation and is species-dependent. Theoretically for the type of lumber that is considered in this research, a peak that contains more than 10 percent of the total board area should be considered as clear wood. This follows from the assumption that the total defect area on a board should occupy no more than 10 percent of the total board area. However, experiments [CHO91] have shown that an M value of 25 yields better results. Hence, M is set to 25.

4) Within each interval $[t_i, t_{i+1}]$ of the histogram, perform a Gaussian smoothing [BAL82, SHI87] on the histogram. The Gaussian function is defined by

$$G(x) = \exp\left(-\frac{k^2}{2s^2}\right)$$

where k is a variable representing transmission level values and s is the standard deviation of the Gaussian smoothing filter. Obviously if too small a value for s is used in the function then the histogram might not be smoothed enough. If too large a value for s is used then the histogram's shape might be distorted. Experiments suggest that $s = 3$ is a good choice.

5) Find the subintervals $[t_j, t_{j+1}]$, in $\{[t_i, t_{i+1}] \mid i = 0,1, \ldots, n\}$ such that $H_x(k) \geq CMIN$ for all $k \in [t_j, t_{j+1}]$. The reason for choosing these subintervals is that Gaussian smoothing typically generates many spurious inflection points when it is applied to intervals where the values of $H_x(k)$ are small. As before, the value of *CMIN* is experimentally selected and varies depending on the typical size of the defects present on the material surface [CHO91]. Here *CMIN* is selected to be 10 percent of the largest value of $H_x(k)$. Again,

the idea of using this selection process is to make everything as dynamically selected as possible.

6) If $t_i > t_{i-1} + MIND$, select t_j as a threshold. *MIND* is a parameter that controls the minimum allowable distance between two consecutive thresholds. *MIND* is set to 3, the same value used in [CHO91].
7) For each $t \in [t_j, t_{j+1}]$ determine if (a) the first derivative of the smoothed histogram at t is positive and the second derivative of the smoothed histogram changes from negative to positive; or (b) the first derivative of the smoothed histogram at t is negative and the second derivative of the smoothed histogram changes from positive to negative. If either of the above is true, then t is considered as a valid inflection point and is selected as a threshold.
8) If there exist two consecutive intervals, say $[t_j, t_{j+1}]$ and $[t_{j+1}, t_{j+2}]$, of the histogram $H_x(k)$ such that both A_j and A_{j+1} are both greater than or equal to M percent of the total image area, the two intervals must be merged into one interval, i.e., $[t_j, t_{j+2}]$. Since these two big peaks are caused by early wood/late wood variation, wherefore, both peaks represent clear wood. To merge the two intervals, the threshold that separates the two intervals is deleted. The result of this pair-wise merging of intervals is an interval $[t_H, t_L]$, which represents clear wood, with an area bigger than M percent (as define in step 3). Therefore, t_H is selected as the threshold for high-density regions, and t_L is selected as the threshold for low-density regions.

After the thresholds have been found, intervals of the histogram between each two consecutive thresholds are assigned a unique interval label. All pixels that fall into an interval of the histogram are then assigned the value of the interval label. This operation classifies pixels into different categories associated with high-, normal-, or low-density regions. The result of thresholding is another image; a thresholded image $X_T(i, j)$ that is registered to the original image $X(i, j)$. If a pixel falls into these defect regions found earlier as mentioned in 7.1.1, no change is made to the corresponding pixel in the image $X_T(i, j)$.

7.2.3 Connected Component Labeling

The thresholding operation segmented pixels into one of three categories: pixels with higher density, pixels with normal density, or pixels with lower density. The same algorithm and procedures discussed in section 6.2.3 are now used to perform connected-component labeling on the thresholded X-ray image. The operation of connected-component labeling takes the thresholded X-ray image $X_T(i, j)$ as its input and assigns all the pixels that have the same class label, i.e., segmentation category, and that form a connected component, a unique connected-component label. It then stores these results into a connected-component image $X_C(i, j)$.

7.3 The Intermediate-level Processing of X-ray Images

To determine if connected components in $X_C(i, j)$ are truly wood defects, component measures are extracted to determine if the connected components belong to a certain defect class. The primary objective in the analysis of the X-ray image is to identify the following defect classes: knots, holes, splits, high-density (internal, not visible from wood surface) defects and low-density (internal) defects. For example, average X-ray transmission can indicate if a connected component has a high density or a low density and can, therefore, provide an important clue as to whether a connected component should be labeled as a knot (e.g., a high-density defect) or a hole (e.g., a low-density defect). The size of a connected component, i.e., its area, is an important connected-component measure because it can indicate whether a connected component is large enough to be considered as an objectionable defect. Moreover, measures of location and shape are used to precisely identify a connected component (see Section 7.3).

Similar to the processing of laser images described in Section 6.3, the initial connected components that are created by this operation often include many meaningless small ones (e.g., connected components that only contain one or two pixels that are usually caused by noise) or over-fragmented defect areas, possibly caused by digitization noise or the nature of the density variations in wood. These small connected components should be either eliminated if they are

caused by noise or merged with similar adjacent components before any further analysis is done on the X-ray image.

7.3.1 Initial Measurement Extraction

For each connected component in the connected-component image, $X_C(i, j)$, the following simple measures are extracted:

- Area of the component
- Height of the minimum inscribing rectangle of the component
- Width of the minimum inscribing rectangle of the component
- Average transmission level (inversely proportional to density)
- Upper left corner of the minimum inscribing rectangle of the component
- Lower right corner of the minimum inscribing rectangle of the component
- Perimeter (total number of pixels on the boundary of the component).
- Touch_edge (total number of pixels on the perimeter of the component that touch the board's edge)

These simple component measures are easy to compute and are extracted first. The areas of connected component and average transmission level are essential ones because they are required to perform the operation of small component elimination and component merging that is performed on the X-ray image. The other measures described above are extracted at this point of the processing because it is computationally efficient to do so.

During the extraction of these component measures, the largest connected component, i.e., the area that represents clear wood, is also identified. Since it has been assumed that total areas of all the defects constitute only a small percentage of the total board area, the largest area should always be an area of clear wood. The average transmission value of this component is the average transmittance of clear wood to X-rays and this value will be used to normalize transmission values of other defect regions later. Note that on rare occasions a large defect region might go from one edge of the board to another. If this occurs, the clear wood area would be divided into the two largest regions.

7.3.2 Small Component Elimination

Small components that result from the over-fragmentation of a defect area will normally be adjacent (e.g., one or two pixels) to other components comprising the defect. For each small component whose area is smaller than a threshold (e.g., two pixels) and is adjacent to one or more larger components, this small component is merged with the larger component to which it is nearest (in terms of transmission value). If no adjacent component is found, the small component is then considered to be noise and deleted. If smaller component is merged, the basic measures of the two components are updated immediately after the merge occurs. For a small component with more than two pixels, one needs to check more neighbors of the component and consequently more complex algorithms are required. Therefore, it is better to leave such components alone until after the component merging operation is performed. This small connected-component elimination process is computationally simple and has been found to work well with X-ray images of wood from the sensing device described in Chapter 3.

However, in other applications where much more noise is present, a more computationally complex connected-component merging method might have to be used. Such methods are described in [PAV77].

7.3.3 Component Merging

Even after all the small two-pixel components have either been merged or deleted, further component merging operations are still necessary because of the nature of the rough surface of some defects. A number of Dempster-Shafer theory-based component merging schemes have been proposed by researchers to merge two adjacent components [HAN88, CHO91]. These methods were found to be appropriate in this application. This additional merging addresses the problem of merging components that have different average transmission values. Since two different average transmission values usually indicate that two components do not belong to the same defect class, a more complex component merging approach is required to determine whether

components with variations in transmission value are statistically equal and thus should be merged.

In the Dempster-Shafer theory of evidence, a frame of discernment Q is defined as a set of mutually exclusive and exhaustive propositions about this frame, e.g., a set of hypotheses concerning the component merging problem. In the component merging problem, Q = {*merge*, *don't_merge*}, where *merge* (*don't_merge*) represents a hypothesis that two given components should (should not) be merged. Associated with each distinct piece of evidence is a basic probability assignment (*bpa*) that represents the impact of the evidence on a subset of Q. A distinct piece of evidence is considered as a statistical test discussed below that generates its own *bpa*. Let m_i be the *bpa* for *i*-th test. Since, by definition, $\sum_{A \subseteq \Theta} m_i(A) = 1$ and $m(f) = 0$. Therefore $m_i(\Theta)$ that is defined by

$$m_i(\Theta) = 1 - (m_i(merge) + m_i(don't\_merge))$$

can be greater than 0. Under this formulation $m_i(\Theta)$ defines the measure of ignorance representing the extent to which the *i*-th test cannot discriminate between *merge* and *don't_merge*. Since Dempster's rule of combination is commutative and associative, the m_i*'s* from any number of tests can be combined regardless of order. Let m_1 and m_2 be *bpa*'s corresponding to two distinct pieces of evidence (or tests *TX* and *TY*). Then a new *bpa m* can be calculated using Dempster's rule of combination given by

$$m(f) = 0$$
$$m(A) = K \sum_{TX \cap TY = A} m_1(TX) m_2(TY), \quad A \neq f$$

where

$$K^{-1} = 1 - \sum_{TX \cap TY = f} m_1(TX) m_2(TY)$$

Currently, two statistical T-tests are performed on the two components C_1 and C_2. The results of these tests are used to determine if the two components should be merged. One of these T-tests examines the difference in average transmission value between the two components. The other one tests the difference in average transmission value between adjacent local areas (areas L_1 and L_2 in Figure 7.2) of the two components. To understand this second test one must

understand how these local areas are defined. Let MIR_1 be the component defined by the minimum inscribing rectangle of component C_1. Similarly let MIR_2 be the component defined by the minimum inscribing rectangle of component C_2. Now let MIR be the intersection of MIR_1 with MIR_2. Then local area, L_i, is defined to be the intersection of component C_i with MIR. See Figure 7.2 for a graphical presentation of this mathematical development. Each of these two tests generates a basic probability assignment. The two probability assignments are combined using the Dempster-Shafer theory of evidence [SHA76, CHO91] to generate a final confidence or belief value concerning whether the two components should be merged.

The component merging operation is implemented using the following steps [CHO91]:

1. Select component C_1, whose area is the smallest among all the components.
2. Find the component C_2 that is the most similar to C_1 in terms of average transmission value among the all components that are adjacent to C_1.
3. Apply a T-test to the components C_1 and C_2. Let m_1 be the probability assignments generated by this statistical T-test. Also apply a T-test to the two local areas, L_1 and L_2 defined in Figure 7.2 and let m_2 denote the probability assignments generated by the this statistical T-test.
4. Calculate m by combining m_1 and m_2 using Dempster's rule of combination. If m is greater than 0.5, then merge the two components.

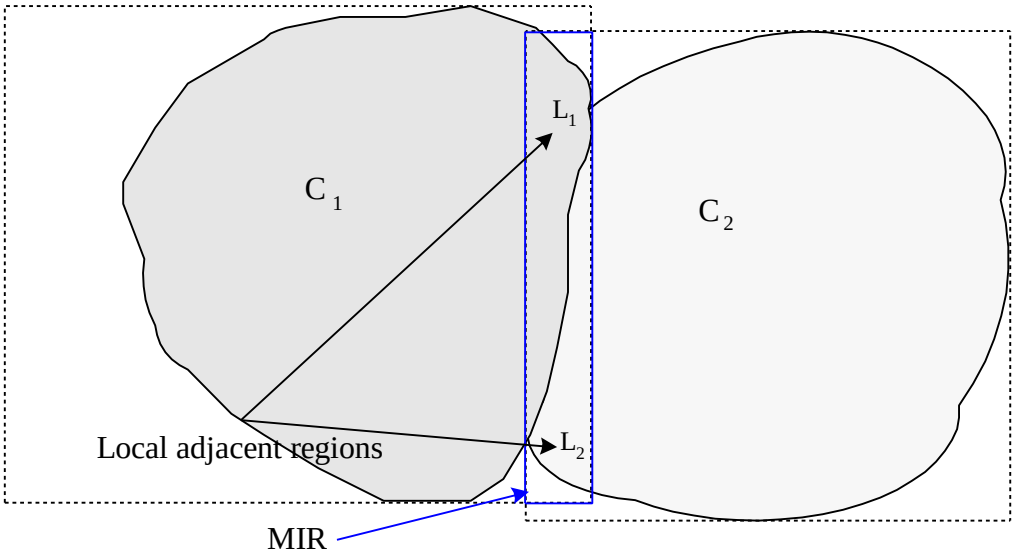


Figure 7.2 Local adjacent region L_1 (L_2) defined as intersect of C_1 (C_2) and MIR

5. Repeat steps 1 through 4 until each component whose area is smaller than a predetermined threshold has been examined.

The statistical T-tests are based on two assumptions: 1) the transmission values of pixels in both components C_1 and C_2 are stochastically independent; 2) the transmission values are from identical normal distributions, and that both the mean and the variance of this distribution are unknown but equal. Similar assumptions and tests have been used by other researchers [HAR85, CHO91, GOZ92].

Step 3 above involves:

1) Calculating the sample means $\overline{Y}_1$, $\overline{Y}_2$, and the sample variances S_1 of Y_1, S_2 of Y_2 where the random variables Y_1 and Y_2 denote transmission values of pixels in component C_1 and C_2, respectively.
2) Let n_1 and n_2 be the sample sizes (number of pixels in a component) of Y_1 and Y_2 respectively, and compute the test statistic, t_{obs}:

$$t_{obs} = \frac{\overline{Y}_1 - \overline{Y}_2}{S\sqrt{\frac{1}{n_1} + \frac{1}{n_2}}}$$

where

$$S = \sqrt{\frac{(n_1 - 1)S_1^2 + (n_2 - 1)S_2^2}{n_1 + n_2 - 2}}$$

3) Determine probability assignment m_1 (or m_2) using the following rule:

If $|t_{obs} < T_1|$ then

$m_1(merge) = 0.5$; $m_1(don't\_merge) = 0.1$;

Else if $|t_{obs} > T_2|$ then

$m_1(merge) = 0.1$; $m_1(don't\_merge) = 0.5$;

Else

$m_1(merge) = 0.2$; $m_1(don't\_merge) = 0.2$;

where T_1 and T_2 are thresholds that depend on lumber species and are often determined experimentally [CHO91]. T_1 is selected so that if $|t_{obs} < T_1|$ holds, there should not be a

significant difference between the average transmission value of C_1 and the average transmission value of C_2. Similarly, T_2 is determined based on if $|t_{obs} > T_2|$ holds; then there is a difference between the two average transmission values. For red oak lumber, T_1 and T_2 are selected as 5 and 20, respectively.

7.3.4 Additional Measurement Extraction

Once the component merging operation is finished, the following additional component measures are computed for each component:

- Center of mass
- Compactness
- Elongatedness

The final output of the intermediate-level processing includes: 1) the final version of image $X_C(i, j)$, called the connect-component image where each component is assigned a unique label; 2) a list of components and their respective measures; and 3) a number specifying the total number of final components that have been found. The output is then forwarded to the high-level X-ray image-processing module.

7.4 High-Level Processing of X-ray Images

The high-level processing of X-ray images is responsible for identifying the defect class to which a defect component belongs. There are six defect classes in the X-ray images, i.e., *knot*, *hole*, *split/check*, *mineral streak*, *high-density*, and *low-density* area. Each defect class is defined and described by a set of measures that uniquely characterize that defect class. The measures used to characterize each defect class were determined by the analysis described below. The goal of this analysis is to aid in the design of expert defect recognition modules, one module for each of the six possible defect classes. A number of expert modules have been developed in this research. Each of these modules is designed to identify a particular defect class using a fuzzy logic approach. Each component is sent to every expert module for evaluation. Each module

then returns a similarity measure called "approaching degree" designating the similarity between the component and a specific defect class. The final decision is made based on maximizing the value of this measure. More precisely, a component is identified as being a member of the defect class for which its value of approaching degree is the largest if the value of approaching degree is greater than a threshold value (e.g., 0.5). If all values of approaching degree are smaller than this threshold, then the component is considered to be clear wood. All the classification results are subsequently passed to a post-processing module. This module merges together those defects that are close to one another as discussed in section 6.3.3. The structure of the X-ray vision module is shown in Figure 7.3.

To identify some defects such as knots or high-density areas, information from other sensing modalities is often needed. For example, a knot appears as a round area with high density in the X-ray image. However, whether the knot is visible on a board's surface cannot be determined from the X-ray image alone since the X-ray image only measures transmission value. The color image of the board provides crucial information regarding whether the knot is on a board's surface. But information provided by the other two imaging sensors is extracted only when it is necessary and is discussed in Section 7.3.1. This approach of cross referencing other images only when needed substantially reduces the time required to fuse the information from all three imaging modalities for defect identification when compared to the traditional approach where all the information is extracted first, and subsequently fused.

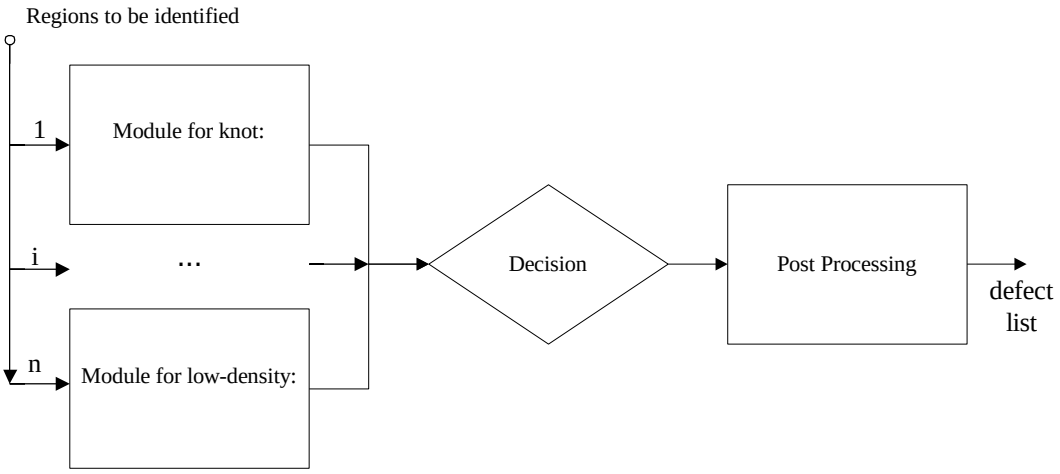


Figure 7.3 Structure of the X-ray vision module

7.4.1 Fuzzy Logic Approach to Defect Classification

As was argued previously in Section 3.2.3, a fuzzy logic approach seems the most appropriate for identifying knots and other wood defects. The nature of fuzzy sets and the relationship between fuzzy sets with different membership functions provides a capability for identifying defects. In particular, the theoretical basis for the fuzzy logic approach used to label defects is based on the concept of "weighted approaching degree" [WAN83, ROS95]. In the next several paragraphs, the concept of approaching degree will be briefly introduced. Then, how this concept is adapted to identify lumber defects will be described.

Let B and A be two fuzzy sets in the same universe U with membership function $m_B(x)$ and $m_A(x)$, respectively. To assess the similarity of the two fuzzy sets B and A, Ross [ROS95] defined the following equations:

$$(B, A)_1 = (B \bullet A) \wedge \left(\overline{B \oplus A}\right) \tag{7.8}$$

$$(B, A)_2 = \tfrac{1}{2}\left[(B \bullet A) + \left(\overline{B \oplus A}\right)\right] \tag{7.9}$$

where $(B \bullet A)$ and $(B \oplus A)$ are the inner product and the outer product of A and B, i.e.,

$$(B \bullet A) = max\{min[m_B(x), m_A(x)]\}, \quad x \in U \tag{7.10}$$

$$(B \oplus A) = min\{max[m_B(x), m_A(x)]\}, \quad x \in U \tag{7.11}$$

and where $(\overline{B \oplus A})$ is the complement set of $(B \oplus A)$.

Equations 7.8 and 7.9 both describe metrics for assessing the degree of similarity between B and A. Thus we have that the similarity of B to A can be gauged using either

$$(B, A) = (B, A)_1 \text{ or } (B, A) = (B, A)_2 \tag{7.12}$$

When either of the values of (B, A) from Equation (7.12) approaches 1, the two fuzzy sets A and B are "more closely similar;" when either of the values of (B, A) approaches 0, the two fuzzy sets are "more dissimilar." The metric in Equation (7.8) uses a minimum property to describe similarity, and the metric in Equation (7.9) uses an *arithmetic* expression to describe similarity [ROS95]. Both of these metrics express a concept called the *approaching degree* [WAN83, ROS95].

In general pattern recognition problems, one is often interested in comparing a new data sample to a number of known classes (or patterns of known objects) and finding which class the sample most closely resembles. Suppose there are c classes of defects, say $label_i$, i = 1, … , c, and each class, say $label_i$, is characterized by m_i measures. Hence, each class, $label_i$, in m_i-dimensional measure spaces can be represented as a fuzzy class given by

$$A_i = \{A_{i1}, A_{i2}, \cdots, A_{im_i}\}$$

where $i = 1, 2, \cdots, c$ corresponds to each of the c classes (patterns). Each A_{ij} is a fuzzy set that has membership function $m_{A_{ij}}(x)$ and describes jth measure of data pattern $label_i$.

With regard to class $label_i$ a new data sample is also described by m_i measures as a collection of non-interactive fuzzy sets [ROS95],

$$B = \{B_1, B_2, \cdots, B_{m_i}\}.$$

The approaching degree concept can be used to compare the new data sample with the known data patterns. Often, some of the measures are more important than others in an application. Therefore, a set of normalized weighting factors w_j is introduced, where

$$\sum_{j=1}^{m} w_j = 1$$

the value of the approaching degree of fuzzy set B, i.e., the data sample, to each of the known c patterns ($i = 1, 2, \cdots, c$) is given by

$$(B, A_i) = \sum_{j=1}^{m} w_j \left(B_j, A_{ij}\right) \qquad (7.13)$$

where (B, A_i) is the approaching degree defined as in Equation 7.8.

The sample B most closely matches pattern A_j if

$$(B, A_j) = \max_{1 \le i \le c} \{(B, A_i)\}, \qquad (7.14)$$

i.e., where the maximum approaching degree value is achieved. When the collection of fuzzy sets $B = \{B_1, B_2, \cdots, B_{m_i}\}$ reduces to a collection of crisp singletons, i.e., B = {x_1, x_2, …, x_m}, then Equation 7.13 reduces to

$$m_{A_i}(x) = \sum_{j=1}^{m} w_j \cdot m_{A_{ij}}(x_j) . \tag{7.15}$$

The sample singleton, x, is closest to pattern A_i when Equation 7.15 reduces to

$$m_A(x) = \max_{1 \le i \le c} \{ m_{A_i}(x) \} \tag{7.16}$$

The first steps that must be undertaken to use this fuzzy theoretic approach is: 1) to identify measures that can be used to separate the various classes; 2) to define fuzzy set membership functions on the domain of definition of each of these measures, i.e., a fuzzy membership function for each class for each measure; and 3) one must determine the weights that will be used to define the approaching degree measures. In the following sections each of these problems will be addressed for each of the six defect classes that are considered in the processing of X-ray images.

7.4.2 Knots

A knot is defined as a portion of a branch that is inside the trunk of a tree. The grain in a branch generally runs perpendicular to the grain of the tree trunk. However, this defect class includes not only knots but also other similar defects such as burl, bud trace, and bird peck [FOR96]. Whenever one of these defects occurs, it will be classified as a knot defect. It is difficult to describe knots mathematically because they appear in many different shapes and sizes. Large knots may have a diameter of up to two to three inches or sometimes even larger, while small knots have a diameter of 1/4 inch or less. Figure 7.4 illustrates the variability in knot shape and density as they appear in X-ray images of the boards used in this research work.

A knot typically appears as a dark area in an X-ray image since it is usually of higher density than clear wood. Experiments have shown that the average transmission value of a knot is 10 percent to 50 percent lower than the transmission value of clear wood. This attribute can be gauged by the normalized transmission value, which is the transmission value of the knot divided by the average transmission value of surrounding clear wood. The average transmission value of clear wood is the location of a histogram peak found in Section 7.2. If a histogram has two peaks

Small knots

Large knots

Figure 7.4 Knots in X-ray images

due to early/late wood, the average transmission value is the middle point of the two peak locations. In some rare cases, the density of a component can be very high, so the average transmission value of the component is 60 percent lower than the average transmission value of clear wood. This component is probably not a member of the knot defect class but rather is more likely to be a foreign metallic object that is sometimes embedded in wood. This work does not address identifying such objects though doing so would be relatively straightforward.

While all of the above suggests that measuring a component's average transmission level would be a very useful measure for identifying knots, this measure is not sufficient by itself. Components containing knots do not always have lower transmission values than clear wood. In fact, knots may have a lower density than clear wood due to the fact that part of the knot might be missing. Hence the average transmission level of the knot is reduced to a value that is below that of clear wood. A knot also can have one or more splits/checks in it. These splits/checks reduce the average transmission values to a point where the X-ray image of a knot might not appear significantly different from that of clear wood.

Therefore, other measures for identifying knots are needed. Another seemingly useful measure would be one that can characterize a knot's shape. A knot often appears as a cross-section of a branch. Hence, most knots have an approximate circular or elliptical shape.

However, due to natural variations, few knots have an ideal circular or elliptical shape. Unfortunately, just extracting the boundary of a knot has proven to be a complicated and computationally time-consuming task [CHO91]. Trying to curve-fit this boundary data to a conic section equation just adds more complexity and computational time to this shape-measuring process. Because of this complexity, a simpler method for measuring shape was chosen for this research. The computationally simple shape measure chosen was the ratio of component height to component width. This measure can be used to determine whether a component has an approximate circular or elliptical shape. The ratio of component height to component width is small for approximate circular or elliptical shapes. While the ratio of component height to component width is effective at differentiating components that are knots from components that are mineral streak, this measure cannot be used to separate knots from some of the other defect classes.

Consequently, another quickly computable shape measure was added to the measurement set used to identify knots. This measure is the compactness measure. The compactness of a component is defined to be the ratio of the square of its perimeter to four times its area. The compactness of a circle is

$$C_c = \frac{(2p \cdot r)^2}{4p \cdot r^2} = p$$

and the compactness of a square is

$$C_s = \frac{(2p \cdot a)^2}{4a^2} = p^2$$

A rectangle has a higher compactness measure than that of a square. The higher the compactness measure, the less compact the component. Because of their approximate circular or elliptical shape, components that are knots have low values for their compactness measure. Statistics extracted from the sample of knots used in the research show that the compactness of knots has a mean value of 6.5 and a standard deviation of 2.8.

Unfortunately, just as in many pattern recognition applications, adding the compactness measure to the measurement set still does not guarantee that knots can be accurately identified.

For example, mineral streak can also have compactness measurement values that are similar to those of knots. Hence, if knots are to be accurately identified by the system, additional sensory information from the color image and/or the laser image must be used.

In a color image, a component containing a knot typically has a low intensity, i.e., is often dark and is reddish in color. In contrast, a component that is mineral streak typically has a low intensity, i.e., is dark; is a strip, i.e., has a high compactness measure value; and is less reddish in color than knots are. Therefore, the average intensity and the average red level of a component are two additional measures that are useful for distinguishing knots from mineral streak. As a consequence, the knot and mineral streak defects can be more easily separated when multiple dissimilar sensors are used to make the identification.

Given the above, it appears that the defect label *knot* should be assigned to components that have the following characteristics:

1) The component's average density is high, or the average transmission value (inversely proportional to density) t is low;
2) The component's height/width ratio h is low;
3) The component's compactness measure cp is small;
4) The component's color intensity, g, is low (it appears as a dark region in the color image);
5) The component's average red gray level r is high.

Hence the knot defect class can be expressed as a fuzzy class given by $A_{knot} = \{A_{knot,\ low\_trans}, A_{knot,height/width}, A_{knot,\ compactness}, A_{knot,\ darker}, A_{knot,\ redder}\}$. In this case, the number of measures, m, is five for the knot class. Here A_{11} is a fuzzy set that describes low-average transmission values with membership function $\mu_{knot,\ low\_trans}(t)$. A_{12} through A_{15} are fuzzy sets that characterize the other four measures and have membership functions $\mu_{knot,\ height/width}(h)$, $\mu_{knot,\ compactness}(cp)$, $\mu_{knot,\ darker}(g)$, and $\mu_{knot,\ redder}(r)$, respectively.

Once the key measures are selected, the next step is to construct the fuzzy sets and their associated membership functions. It is these membership functions that are used to actually describe the *knot* class. As with many fuzzy systems, this system's performance depends heavily on the fuzzy rules that will be used. The fuzzy rules, in turn, depend heavily on the fuzzy membership functions that are defined to describe them. Therefore the selection of an appropriate fuzzy membership function will determine the performance of this fuzzy logic approach.

In many engineering applications [COX94, RAH97] such as system control, a fuzzy set is often constructed from the design engineers' intuitive understanding of a specific problem. A membership value is generally not an absolute value but rather is relative and sometimes even quite subjective in nature [KAN82]. In fact, selecting the right membership function is the basic challenge in knowledge engineering. However, the best way to derive fuzzy membership functions has become a popular research topic. Probability frequency distributions (probability density functions or probability mass functions), subjective approximations of the control surface [COX94], and mathematical surface sampling have all been used to generate fuzzy membership functions. Many researchers have used trapezoidal functions as membership functions or piece-wise linear functions without giving justification as to why this membership function might be the most suitable for an application. Consequently, the best method for constructing fuzzy set membership functions remains an open question.

In this research, conditional probability density functions are used to construct the fuzzy membership functions. These conditional density functions are assumed to be Gaussian. For a given measure, the conditional density function employed is $f(x|knot)$, that is we are interested in the distribution of the measure's values from the *knot* class. Standard procedures are used to estimate the mean and variance needed to define each of the needed conditional density functions. Other researchers have also used Gaussian functions as membership functions [ROS95].

Consequently, in this research a typical Gaussian-based membership function $\mu_A(x)$ for the fuzzy set *A* has the general form:

$$\mu_A(x) = \begin{cases} e^{-\frac{(x-a)^2}{s^2}} & if\ X_1 < d < X_2 \\ 0.1 & Otherwise \end{cases} \quad (7.17)$$

where a and s are, respectively, the mean value and the standard deviation of the measure. The parameters a and s define the shape of the membership function. The values $X_1$and X_2 depend on the mean, a, and the standard deviation, s. Usually, X_1 and X_2 are selected to be ($a - 1.65\,s$) and ($a + 1.65\,s$), respectively. These choices correspond to the 5th and 95th percentiles of Gaussian distribution defined by a and s. For this type of membership function, when $x = a$, the membership function value is 1.0. As the difference between x and a increases, its membership function's value decreases. When x is far away from a, its membership is set to 0.1. This reflects the likelihood that x is not a member of fuzzy set A.

The membership function for the fuzzy set $A_{11} = A_{knot,average\_transmission}$ can therefore be described by the membership function:

$$\mu_{\text{knot, low\_trans}}(t) = \begin{cases} e^{-\frac{(t-77)^2}{49}} & if\ 66 < t < 89 \\ 0.1 & Otherwise \end{cases} \quad (7.18)$$

where t is the normalized transmission value and is inversely proportional to density. The normalized transmission value is the average transmission value of knots divided by the average transmission value of clear wood. This function is an approximation to the histogram of the normalized average transmission values that were computed from about 100 sample knots available at the time this research was conducted. Since the average transmission value of a knot depends on the density of the knot as well as the thickness of the board, the normalized average transmission value provides a better comparison of transmission values between a knot and the clear wood of the board.

Similarly, the membership function for the fuzzy set $A_{12} = A_{knot,height/width\_ratio}$ is small and is defined as:

$$\mu_{\text{knot, height/width}}(h) = \begin{cases} e^{-\frac{(h-2.4)^2}{1.44}} & if\ 0.42 < h < 4.4 \\ 0.1 & otherwise \end{cases}. \quad (7.19)$$

The membership function for the fuzzy set $A_{13} = A_{knot,compactness}$ is defined by:

$$\mu_{\text{knot, compactness}}(cp) = \begin{cases} e^{-\frac{(cp-6.25)^2}{2.25^2}} & if\ \ 2.54 < cp < 9.96 \\ 0.1 & otherwise \end{cases} \tag{7.20}$$

The membership function for the fuzzy set $A_{14} = A_{knot,intensity}$ is defined by:

$$\mu_{\text{knot, darker}}(g) = \begin{cases} e^{-\frac{(g-0.53)^2}{0.0225}} & if\ \ 0.28 < g < .78 \\ 0.1 & otherwise \end{cases} \tag{7.21}$$

Finally, the membership function for the fuzzy set $A_{15} = A_{knot,red\_component}$ is defined by:

$$\mu_{\text{knot, redder}}(r) = \begin{cases} e^{-\frac{(r-1.42)^2}{0.04}} & if\ \ 1.09 < r < 1.75 \\ 0.1 & otherwise \end{cases} \tag{7.22}$$

Given the above described fuzzy sets and their associated membership functions for any new data sample, i.e., a component together with its measures, $B = \{t_1, h_1, cp_1, g_1, r_1\}$, the approaching degree between this data sample and knot defect class can be calculated using Equation (7.15), i.e., using

$$\begin{aligned} AD_{knot} &= \sum_i w_i m_{A_{1i}} \\ &= w_1 m_{A_{11}} + w_2 m_{A_{12}} + w_3 m_{A_{13}} + w_4 m_{A_{14}} + w m_{A_{15}} \\ &= w_1 \mu_{\text{kont, low\_trans}}(d_1) + w_2 \mu_{\text{knot, height/width}}(h_1) + w_3 \mu_{\text{knot, compactness}}(cp_1) \\ &\quad + w_4 \mu_{\text{knot, darker}}(g_1) + w_5 \mu_{\text{knot, redder}}(r_1) \end{aligned} \tag{7.23}$$

where w_i's are weight factors and are determined by experimentation.

In this research, all the weight factors were often set at 0.2. That is, each measure is given an equal weight. This means that the resulting sum is an arithmetic average of all of the membership function values. In general, the use of equal weighting factors has resulted in satisfactory identification results. Note that further research is needed to create methods for

determining optimum weight value factors. Since creating optimum weight values is beyond the scope of this research, equal weighting factors were selected for use here.

The approaching degree between a given X-ray image component and the *knot* defect class is the output of the knot expert defect recognition module. Each component found in the X-ray image is analyzed by all the expert defect recognition modules. The output of each expert defect recognition module is the value of the approaching degree between this component and the specific defect class the expert defect recognition module was created to identify. All approaching degrees for this component are then fed to the post-processing module. Within this module, the maximum approaching degree for this component is determined. Based on this information as well as some other factors it is the post-processing module that assigns the defect label to this component.

7.4.3 Mineral streak

Mineral streak is an olive to greenish-black or brown discoloration of undetermined cause. Mineral streak usually appears in long narrow strips on a board's surface. A component of mineral streak in an X-ray image generally has the following three characteristics:

- It appears as a dark area with smaller transmission values since mineral streak has a higher density than clear wood. Although knots also have a higher density than clear wood and the density difference between a knot and mineral streak is not always significant, the smaller transmission value is an important way to help identify the mineral streak.
- Its ratio of component height/component width (as defined before) is usually large since mineral streak is often a long narrow strip. There is no clear boundary between defect class *knot* and defect class *mineral streak* for this measure since some knots have height/width ratios that are close to the ratios for some shorter mineral streaks. However, as the height/width ratio increases, the likelihood that a component might be mineral streak also increases.

- It has a higher compactness value (as defined before) since a component of mineral streak is usually almost rectangular in shape. Again, knots oftentimes have the same compactness values.

Clearly, it is difficult to unambiguously label a component as being mineral streak with measures extracted from an X-ray image alone. Hence, additional information from the corresponding color image is needed to increase classification accuracy. In the color image, a component of mineral streak manifests itself as follows:

- The component is very dark and of low intensity when compared to clear wood. The measure used to gauge this property is called the normalized average color intensity, *ng*. Note that this is a new measure that has not been mentioned previously. It is defined to be the ratio of local average intensity to the average intensity of clear wood. The local average intensity is the average intensity of the component. The average intensity of clear wood is estimated by using the location of the largest peak in the histogram of the black/white image as the value for this measure. If two such peaks exist, the average value or the middle point of the two locations is used. The histogram of the black/white image is extracted at the image pre-processing stage, and the peak(s) are detected with the algorithm discussed in Section 7.2.
- The component usually is less reddish than either clear wood or knots. The measure used to gauge this quality is called the intensity normalized red value, *inr*. It is defined to be the ratio of the average red channel value of the component to the average color intensity of the component. The local red channel value is the average red channel response of the component in the color image. The average local intensity value is the average of all three color channels over the component in the color image.

Obviously, mineral streak possesses different characteristics, so the very similar measures used previously could not be utilized to describe defect class mineral streak and new measures have to be used. It would seem that a component that contains *mineral streak* could be described as follows:

1) The average transmission value, *d*, of the component is low (i.e., average density is high);
2) The height/width ratio, *h*, of the component is high;
3) The compactness measure, *cp*, of the component is large;
4) The normalized average color intensity (*ng*) is low;
5) The intensity normalized red value (*inr*) is low.

The above observations are then used to construct fuzzy sets that describe defect class *mineral streak.* All these fuzzy sets are created in a manner that is very similar to what was done in the previous section. A fuzzy class describing *mineral streak* is given by $A_{\text{mineral}} = \{A_{mineral,transmissions}, A_{mineral,height/width}, A_{mineral,compactness}, A_{mineral,color\_intensity}, A_{\text{mineral,intensity\_red}}\}$ where the membership function for $A_{mineral,transmission}$ is denoted by $m_{mineral,transmission}(d)$, the membership function for $A_{mineral,height/width}$ is denoted by $m_{mineral,\text{height/width}}(h)$, the membership function for $A_{\text{mineral,compactness}}$ is denoted by $m_{mineral,\text{compactness}}(cp)$, etc. All membership functions are of the general form given in Equation (7.17), with parameters listed in Table 7.1. The parameters in Table 7.1 are based on the statistics of component measure listed in Table 7.2. The statistics are extracted from about 50 samples available for this research.

Table 7.1 Parameters of Fuzzy Membership Functions for Mineral Streak

Member Function	Mean *a*	Variance *s*	X1 and X2
$\mu_{\text{mineral,transmission}}(d)$	84.5	5.0	76.25 / 92.75
$\mu_{\text{mineral,height/width}}(h)$	5.30	1.85	2.25/8.35
$\mu_{\text{mineral,compactness}}(cp)$	16.8	2.4	12.9/20.7
$\mu_{\text{mineral,color\_intensity}}(ng)$	65.0	4.6	57.6 / 72.6
$\mu_{\text{mineral,intensity\_red}}(inr)$	0.85	0.10	0.685 / 1.015

For any given data sample, i.e., a component along with its computed measures, the approaching degree of that data sample to the defect class *mineral streak* is computed using:

Table 7.2 Statistics of component measures for a sample of mineral streak

Measures	Average value	Standard deviation	Maximum value	Minimum value	5 & 95 percentile
Avg. normalized transmission (*d*)	84.5%	5.0%	92.0%	80.0%	76.25 / 92.75
Height / Width (*h*)	5.30	1.85	9.6	2.9	2.25/8.35
Compactness (*cp*)	16.8	2.4	21.4	13.1	12.9/20.7
Local intensity / Global intensity	65.0%	4.6%	72.0%	58.7%	57.6 / 72.6
Local red intensity / Local intensity	0.85	0.10	1.015	0.685	0.685 / 1.015

$$AD_{mineral} = \sum_i w_i m_{A_{2i}} \qquad (7.24)$$

$$= w_1 m_{A_{21}} + w_2 m_{A_{22}} + w_3 m_{A_{23}} + w_4 m_{A_{24}} + w m_{A_{25}}$$

$$= w_1 \mu_{\text{mineral, transmission}}(d) + w_2 \mu_{\text{mineral,height/width}}(h) + w_3 \mu_{\text{mineral, compactness}}(cp)$$

$$+ w_4 \mu_{\text{mineral,color\_intensity}}(ng) + w_5 \mu_{\text{mineral,intensity\_red}}(inr)$$

where w_i's are weight factors and where $\sum_i w_i = 1$. As in the case of the *knot* class, all the *mineral steak* class weight factors are 0.2. That is to say each measure is given equal weight.

7.4.4 High density areas

A high-density component is an area that has higher density than clear wood but is not visible in the color image. A high-density component could be an internal defect, or a knot or mineral streak that is not visible on the board face that is being analyzed but is visible on the other board face. A high-density component that the system cannot positively identify as either a *knot* or *mineral streak* will be assigned the label *high density*. Since each face of a board is examined independently, the decision-making computer has to put together the two independent interpretations to arrive at an overall evaluation of what defects need to be removed. Labeling an

area as *high-density* helps the decision-making computer arrive at its analysis by indicating that the defect found is not visible on the board face that is being examined. In some cases, if a defect is only visible on one board face, it may not need to be removed. For example, a mineral streak that is visible on only one board face may not affect the final quality of a rough part, hence there is no need to remove it.

There are two ways to classify a component as a *high-density* area. First, it can be detected by using an approach that is similar to one that was described above for detecting knots. The only difference is that now one must make sure that there is no appreciable color variation between this component and clear wood. Note that when the system is trying to label a component as being a *knot*, it tries to assure that there is a color difference between the component and clear wood, while here the system needs to make sure that no such difference exists.

Clearly in the X-ray image, a component that is a *high density* will have very similar measure values to a component that is either a *knot* or *mineral streak*. This follows from the fact that the vast majority of *high-density* defects are knots or mineral streak that are not visible on the board face being processed. Consequently, if the expert defect recognition experts that label *knots* and *mineral streak* are applied first to a component that is a *high-density* area, then at least one of these expert defect recognition modules should return a *relatively* high value of approaching degree (e.g., 0.5) for this component. This follows from the fact that while this component will have dissimilar color intensity and reddish values to either a knot or mineral streak, it should have a very similar average transmission value, a very similar height/width ratio, and a very similar compactness value to one of these defect classes. And it is the similarity in the last three values that should cause the overall relatively high approaching degree to either the *knot* class or *mineral streak* class. Hence, a component that has a high approaching degree from either of these two expert recognition modules should be checked to see if its corresponding area in the color image does have a different color than clear wood. If no appreciable difference in color is found for this component, then it should be assigned the label *high density*.

Note that this approach to labeling the *high density* class has one very important advantage. Since both the *knot* class and the *mineral streak* class expert recognition modules have to be applied to each component anyway, this approach saves processing time over creating a completely new *high-density* class expert recognition module which would also have to be applied to each component.

Just as was done in the analysis of *knot* and *mineral streak*, this comparison involves checking two component measures--normalized average color intensity and red normalized red. The red normalized red value is a new measure. Its unusual name indicates that the average red value of clear wood is used to normalize the average red value of the component. This distinguishes the measure from the intensity normalized red value where intensity is used to normalize the red value. The intensity of a component will be regarded as being the same as the intensity of a board's clear wood if the normalized average color intensity falls in the interval [0.9, 1.1]. The normalized average color intensity is equal to I_C/I_G, where I_C is the average color intensity of the component and I_G is the average color intensity of the board's clear wood. Please refer to the previous section to see how the average color intensity of a board's clear wood is computed. Similarly, the red color of a component is considered to be the same as the red color of the board's clear wood if the red normalized red value is in the interval [0.9, 1.1]. The normalized red value is equal to I_{CR} / I_{GR} where I_{CR} is the average red value within the component and I_{GR} is the average red value of clear wood. If the values that are computed from the components yield results that fall into these two intervals, then the component is considered to have the same "color" as clear wood. Consequently, the approaching degree between this component and the *high-density* class is assigned a larger value, i.e., 0.9.

7.4.5 Holes

Holes are voids in the wood caused by pests, missing knots, etc. In the grades of hardwood lumber used by the secondary manufacturers, big holes can have a diameter as large as one inch, while worm holes can be as small as 1/16 inch in diameter or smaller. Holes may appear in any part of a board. Figure 7.5 shows some typical holes found in hardwood boards.

Figure 7.5 Holes in color and X-ray images

A component that is part of the *hole* class in X-ray images will generally have the following characteristics:

- The component will be lighter in intensity than clear wood since a hole always has a higher transmission value than clear wood. In more complicated cases such as with small holes or when holes are not perpendicular to the board surface, the transmission value is still higher than clear wood but there may not be that much difference between them. Experiments show that the average transmission value for a hole can vary from 25 percent to 150 percent above the average transmission value of clear wood. This depends on the size, the depth, and the orientation of the hole. However, even given these variations it should be clear that the transmission values of a component are a good indicator as to whether the component is part of the *hole* class. The measure that is used to gauge the difference between a component's transmission values and those of clear wood is called the normalized average transmission value. The normalized average transmission value is defined as the ratio of the average transmission value of the component to the average transmission value of clear wood. The average transmission value of clear wood is estimated to be the average transmission value of the largest component in the X-ray image. Note that the validity of this estimate is based on the previously described assumptions that were made concerning the nature of the boards that are processed by secondary manufacturers.
- The component should have a small height/width ratio, *h*. This follows from the fact that these components typically have a general circular to elliptical shape. This means that the height/width ratio of holes is generally fairly low. This height/width ratio

measure is relatively effective in separating members of the *hole* class from members of the *split/check* defect class. The reason for this is that a split is typically a long narrow separation of wood. This means that components that are of the *split/check* class generally have high height/width ratios. However, in some rare cases, holes such as worm holes may have a large height/width ratio while some splits may have a relatively small height/width ratio. Although it is hard to find a threshold of height/width ratio that can accurately separate all holes from all splits, the ratio is still a good indicator as to whether a component is a member of the *hole* class since a component with a large height/width ratio is much more likely to be a split than a hole.

- The component should have a relatively low compactness measure, *cp*, value. This is due to the fact that these components are typically of a round or elliptical shape. This characteristic is something the *hole* class shares with the *knot* class.
- The component will typically be very dark in color, i.e., has lower color intensity, and will be much less reddish in color than clear wood. The previously described normalized average color intensity is used to gauge the darkness in color as related to clear wood. The previously defined measure red normalized red value is used to gauge the redness of the color in relation to clear wood.

While the X-ray image does contain most of the information that is needed to label a component as being a member of the *hole* class, it cannot be used to determine whether a hole is visible on a board face, and subsequently whether the hole should be removed. Using this system's multi-sensor capability, the color image can be used to make this inference. The required information can be quickly extracted from the color image by simply examining the area of the color image that corresponds to the component in the X-ray image. In a color image, a hole will generally manifest itself as a round dark area that is less reddish in color than either clear wood or knots. In the special case where the hole goes all the way through the board, the component appears light blue in color, i.e., the color of the illuminated background that is used on the color scanning system. Please see Section 3.1 for more information on the illuminated background that is used in the color imaging system.

Based on the discussion presented above, a component that is a member of the *hole* class has the following characteristics:

1) The component's normalized average transmission value (*nd*) should be high;
2) The component's height/width ratio *h* should be low;
3) The component's compactness measure *cp* should be small;
4) The component's normalized average color intensity value (*ng*) should be low;
5) The component's red normalized red value (*rnr*) should be low.

The fuzzy class, describing the *hole* class, $A_{hole} = \{A_{hole,normalized\_trans}, A_{hole,height/width}, A_{hole,compactness}, A_{hole,intensity}, A_{hole,red\_red}\}$ is constructed in a way that is similar to that discussed in Section 7.3.2. The fuzzy set $A_{hole,normalized\_trans}$ has a membership function called $m_{hole,normalized\_trans}(d)$, and the fuzzy set $A_{hole,height/width}$ has a membership function called $m_{hole,height/width}(h)$, etc. All the membership functions are of the general form given in Equation (7.17). The parameters for each membership function are listed in Table 7.3.

Table 7.3 Parameters of Fuzzy Membership Functions for holes

Member Function	Mean Value *a*	Variance *s*	X1 / X2
$m_{hole,normalized\_trans}(nd)$	194	30	144 /244
$m_{hole,height/width}(h)$	1.0	0.45	0.26/1.74
$m_{hole,compactness}(cp)$,	1.5	0.55	0.59/2.41
$m_{hole,intesity}(ng)$,	40	12	20 / 60
$m_{hole,red\_red}(rnr)$,	51	20	18 / 84

For any sample $B = \{nd_1, h_1, cp_1, ng_1, rnr_1\}$ that has been found by the lower-level processing, the approaching degree of this sample to the *hole* class is computed using:

$$\text{AD} = \sum_i w_i m_{A_{3i}}$$

$$= w_1 m_{A_{31}} + w_2 m_{A_{32}} + w_3 m_{A_{33}} + w_4 m_{A_{34}} + w m_{A_{35}}$$

$$= w_1\mu_{\text{hole, normalized\_trans}}(nd_1) + w_2\mu_{\text{hole,height/width}}(h_1) + w_3\mu_{\text{hole,small\_compact}}(cp_1) + w_4\mu_{\text{hole,intesity}}(ng_1) + w_5\mu_{\text{hole,red\_red}}(rnr_1); \quad (7.25)$$

where, just as for the previous cases, the w_i's are weight factors each of which equals 0.2.

7.4.6 Splits

A split is a lengthwise separation of wood. The major difference between a split and a check is that for a split, one of the ends of a split always touches the perimeter of the board, typically the end of the board, see Figure 7.6. A check, on the other hand, always lies completely inside the perimeter of a board. Both splits and checks are critical defects that must always be removed from a rough part. In the hardwood lumber used by secondary manufacturers, large splits can be 0.3m (12 inches) long and as wide as 6.4mm (0.25 inches). Small splits (or checks) might be shorter than 25.4 mm (or one inch) with a width of 0.8mm (1/32 inch). Even though splits and

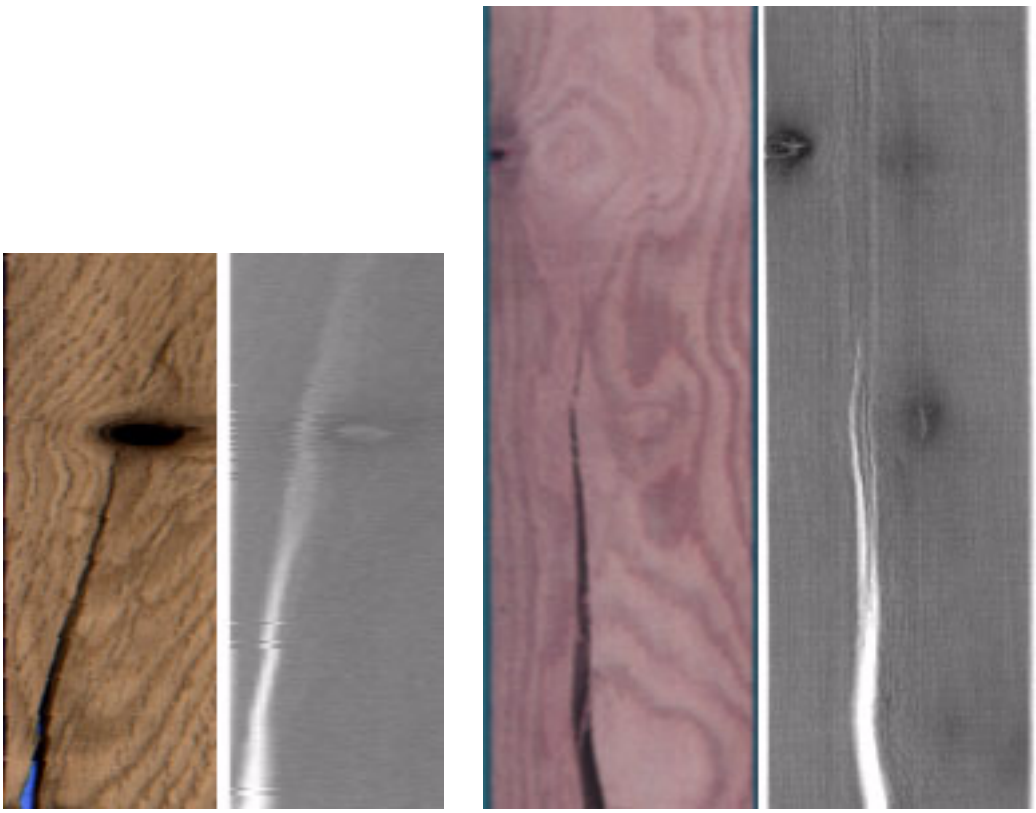


Figure 7.6 Splits occur at one end of boards

checks are quite different in size, they are considered as part of the same defect class, i.e., the *split/check* class.

A component that is a member of the *split* class generally has the following characteristics:

- The component typically has a higher transmission value than clear wood. The reason for this is that in an X-ray image a split is created by imaging the air between the two parts of a board that bound the split. Experiments indicate that the average transmission value of a split can vary from 150 percent to 250 percent higher than the average transmission value of clear wood. This variation depends on the size, the depth, and the orientation of the split. This suggests that the average transmission value should be an important measure for identifying splits. Although holes also possess this same attribute, they generally differ significantly in shape from splits. Figure 7.7 shows the distribution of average transmission values from 50 samples of splits. This histogram is constructed with normalized transmission values, *nd*, i.e., the normalized transmission value is equal to the average transmission value of the component divided by the average transmission value of the clear wood of the board from which the split sample was taken.

Experiments suggest that a component from the *split* class should have the following characteristics:

- The component's height/width ratio, *h*, should be low. This follows from the fact that a split's height is usually large in relation to its width. This is an important measure for identifying splits. It is one of the best attributes that separate the *split* class from the *hole* class.
- The component's compactness measure value, *cp*, should be large. This follows from the fact that a split typically is much longer than it is wide.
- The component's width, *w*, should be small since a split or check is a longitudinal narrow strip.

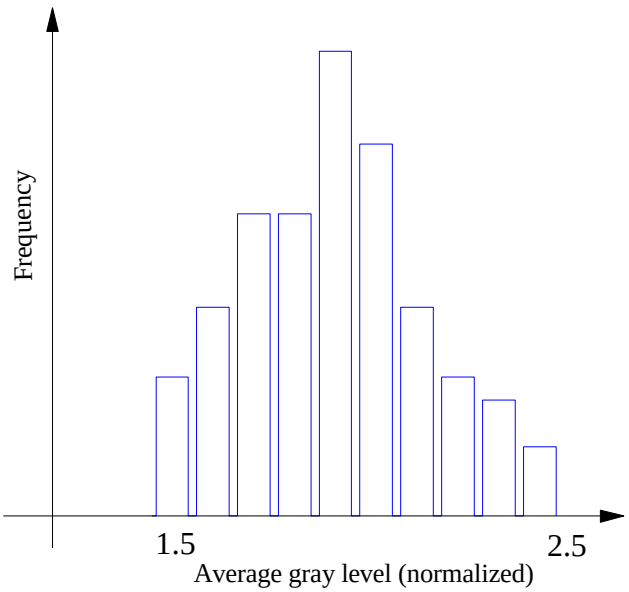


Figure 7.7 Distributions of transmission values taken from 50 splits samples.

Experiments have shown that enlongatedness (as defined in Section 2.1) is another measure that can gauge the compactness of a component. However, this measure was not used in the final implementation of this vision system. The reason for this is that the measure enlongatedness is a much more complicated measure to compute than the standard compactness measure used here. And experiments show that the performance gains obtained using enlongatedness does not offset its computational complexity.

The X-ray image cannot be used to determine whether a split is visible on a board surface. This is an important piece of information since if a split is visible it must always be removed. Rough parts should never contact a visible split. Consequently, just as in the case of holes, color image provides this important piece of information. This information can be quickly extracted by simply processing the area of the color image that corresponds to the component in the X-ray image. In the color image a split generally appears as a narrow, dark strip (see the color images in Figure 7.6). Hence, using surface color, one can easily determine if a split is visible on a board's surface or is completely internal within the board. The measure used to gauge this image feature is the normalized intensity value.

Based on the above discussion, a component that is a member of the *split* class should possess the following characteristics:

1) The component's normalized average transmission value, *nd*, should be high;
2) The component's height/width ratio, *h*, should be high;
3) The component's compactness measure, *cp*, should be large;
4) The component's the normalized color intensity value, *ng*, should be low;
5) The component's width, *w*, should be small.

Consequently, the *split* class can be represented by a fuzzy class given by $A_{\text{split}} = \{A_{split,normalized\_trans}, A_{split,height/width}, A_{split,compactness}, A_{split,intensity}, A_{split,width}\}$. As before, the fuzzy set $A_{\text{split,normalized\_trans}}$ has a membership function called $\mu_{\text{split,normalized\_trans}}(nd)$, the fuzzy set $A_{\text{split,height/width}}$ has a membership function called $m_{split,height/width}(h)$, etc. As before, all these membership functions have the general form given in Equation (7.17). The parameters used to define these functions are given in Table 7.4.

Table 7.4 Parameters of Fuzzy Membership Functions for splits

Member Function	Mean *a*	Sample variance *s*	X1 / X2
$\mu_{\text{split, normalized\_trans}}(d)$	194	30	144 /244
$\mu_{\text{split, height/width}}(h)$	7.5	2.5	3.38/11.63
$\mu_{\text{split, compactness}}(cp)$	4.8	2.0	1.5/8.1
$\mu_{\text{split, intensity}}(ng)$	25.0	10.0	8.5 / 41.5
$\mu_{\text{split, width}}(w)$	12	5	4 / 20

Also just as before, the approaching degree between a component and its set of measures and the *split* class is computed using:

$$\text{AD} = \sum_i w_i m_{A_{4i}}$$

$$= w_1 m_{A_{41}} + w_2 m_{A_{42}} + w_3 m_{A_{43}} + w_4 m_{A_{44}} + w m_{A_{45}}$$

$$= w_1\mu_{\text{spit, normalized\_trans}}(d_1) + w_2\mu_{\text{split, height/width}}(h_1) + w_3\mu_{\text{split, compactness}}(cp_1) + w_4\mu_{\text{split,intensity}}(ng_1) + w_5\mu_{\text{split, width}}(w_1); \quad (7.26)$$

where w_i's are weight factors used and where $\sum_i w_i = 1$. Again all weight factors are set to 0.2.

7.4.7 Low density

Similar in concept to a member of the *high-density* class, a low-density area is defined as an area that has a lower density than clear wood (hence higher transmission values in the X-ray image) and is indistinguishable from clear wood in the color image of board face that is being processed. Since each face of a board is examined independently, the decision-making computer needs information about what is visible on each of the board's surfaces before reaching an overall conclusion about what defects need to be removed. The *low-density* class label is assigned to low-density defects that do not show up in the color image of the board face that is being examined. Such defects may be visible on the other board face or they may be internal defects that are not visible on either board face. Examples of the *low-density* class include holes that are either completely contained in the interior of a board or holes that are visible only on the other board face. Other examples of the *low-density* class include either internal decay or decay that is visible only on the other board face. Still other examples include internal checks or splits, or splits/checks that are visible only on the other board surface. Consequently, a low-density area (an area of high transmission value) that cannot be identified as a hole, split/check, or decay will be labeled as a member of the *low-density* class.

As with the *high-density* class, there are two possible ways of determining whether a component is a member of the *low-density* class. The most straightforward way would be to design an expert defect-recognition module for identifying this defect class. Such a module would necessarily be very similar to the ones used to identify membership in the *hole* class or *split/check* class. That is to say this module would have to analyze connected components that have higher transmission values. However, unlike the modules for identifying holes or split/check, this module would have to examine the corresponding component in the color image in order to assure that a component's corresponding area in the color image is similar to rather than different from the

appearance of clear wood in the color image. If this approach were used, this module would have to be applied to every component that has a lower density (higher transmission value) than clear wood. A second, less obvious, way to assign membership into the *low-density* class would be to apply the modules that identify holes and splits to a component first. If one of these two modules returns a relatively high value of approaching degree (e.g., 0.5) for this component then it is possible that this component might be a member of the *low-density* class. To make the final determination about the component's appropriate label, the surface color of the component's corresponding area in the color image would have to be checked to assure that this area is not visually dissimilar to clear wood. This second alternative is clearly less computationally complex than the first and it is the one selected for use here. It works because a component that is a member of the *low-density* class almost always will have measure values that are very similar to those of the *hole* class or the *splits/check* class in X-ray imagery. Consequently, the approaching degree should be relatively high, reflecting the similarities in measure values that are extracted from the X-ray image. However, the approaching degree should not be very high since there should be little contribution from the measures that are defined on the color image.

To check the color similarity between a component and that of clear wood, normalized average color intensity is used. Note that in the case of the *high-density* class two measures were used to check for color similarity. These were the normalized average color intensity and the red normalized red value. To check the similarity of color in this case only the normalized average color intensity is used. While this measure was defined previously, it will be defined again for the convenience of the reader. The normalized average color intensity is defined to be the local average color intensity (I_C) divided by the average color intensity (I_G) of clear wood. The local average color intensity is the average color intensity of the area in the color image that corresponds to the component in the X-ray image. The average color intensity is estimated to be the location of the tallest peak in the black and white image of the board. If the normalized average color intensity falls in the region [0.9, 1.1] then it is assumed that there is no color difference with clear wood. Therefore, a larger value (e.g., 0.9) is assigned to the value of approaching degree for the class of *low-density*.

7.4.8 Defect identification and Post-processing

After all of the connected components have been processed and each of the defect expert defect recognition modules has been applied to each component, the decision-making module determines the class label that will finally be assigned to each component (see Figure 7.3). The module uses the values of approaching degree that were assigned to the component by each expert defect recognition modules as its input. The decision-making module assigns a label to a component by selecting the class label that has the maximum value of approaching degree to the component's measurement values. The decision-making module uses a three-step process for making the assignment. Step 1) The class label that has the maximum approaching degree to the component's measures is found. Call this label $label_k$. Step 2) If class $label_k$ has an approaching degree to the component that is greater than a threshold value (e.g., 0.5), then label $label_k$ is assigned to the component. Step 3) Else assign the class label clear wood to the component (i.e., the component is considered to be clear wood and not one of the defect classes).

Once all the components have been assigned labels by the decision-making module, these components and their assigned labels are then passed on to a post-processing module (see Figure 7.3). The purpose of the post-processing module is to merge two defects if the distance between them is less than a threshold (e.g., three pixels) and both defects have the same defect class label. The distance between two defects is the distance between their minimum inscribing rectangles (MIR) as discussed in Section 4.4.3. After two defects are merged into one, the defect's MIR is also updated. If a smaller defect is inside another larger removable defect, the smaller defect will be deleted regardless of which defect class it belongs.

After all the components and their assigned labels have been examined by the post-processing module, these label assignments are put on the final defect list. This is exactly the same process that was discussed in Section 4.4.3. If the detection of smaller more subtle defects is required, one can process the X-ray image one more time as discussed in Section 7.4. Otherwise, the final defect list and the list of defects that cannot be determined with a laser image only are passed to the next stage--analysis of color images (Chapter 8).

7.5 Detecting Small Defects in X-ray Images

To detect smaller more subtle wood defects in the X-ray image such as small internal voids or small holes, a new segmentation approach was implemented. In this approach, the bigger, more prominent defect components are all segmented first, including connected-component labeling. After this is done, the effects of these components are removed from the histogram of the X-ray image as discussed Section 7.1.1. Then the images are processed in the same way for a second time (i.e., repeat processing steps in Section 7.1 through Section 7.3).

The motivation for this new method comes from the fact that within a histogram of an X-ray image, the transmission values of pixels representing small defects, such as small interior splits and holes, often lie in the same interval as the transmission values of pixels of big defects. Since the segmentation approach used here is a dynamic thresholding operation, the selection of thresholds directly affects the performance of the segmentation. The thresholds are selected by finding inflection points in the transmission value histogram that are caused by either high-density (low transmission value) defects (e.g., knots) or low-density (high transmission value) defects (e.g., holes). Theoretically it can be argued that if the transmission values of larger defects such as large splits, large holes, large areas of mineral streak or big knots are removed from the histogram of the X-ray image, then the inflection points caused by smaller defects can be more easily found by the algorithm that selects the thresholds. Therefore, removing the effects of these larger defects from the transmission value histogram should allow better segmentation results to be obtained, and consequently, better detection of the smaller defects can be obtained.

Note that the detection of smaller defects in X-ray images is at the expense of extra computations. Unfortunately, at the time that this research was conducted only a few samples were available for experimentation. Based on the limited number of samples and contrary to the theory described above, only a very few additional small defects were found by this new procedure. Based on this experimental data, as limited as it was, the decision was made to make this histogram modification operation an optional step (controlled by a configure file) in the

overall segmentation operation. Further research is required to study the merit of processing the X-ray image a second time to detect smaller defects.

7.6 Processing Examples

In this section a couple of processing examples will be given. The objectives for discussing these results are two-fold. First, the discussion should help reinforce the description of the algorithms that were given above. The reader will be able to see exactly when a particular algorithm is applied and will be able to see examples of processing results that were obtained from its application. The second purpose is to illustrate how less-than-perfect early vision processing can still lead to a meaningful analysis of a board. Please note that Chapter 9 will give other processing examples which is the reason that only two are considered here.

7.6.1 Removal of Defects Found in the Laser Image Processing

As was discussed in Section 7.1, defects that were found during the analysis of the laser image are removed from all future consideration in general, and from the processing of the X-ray image in particular. Removing defects that have been already identified not only improves the quality of the segmentation of the X-ray image but also reduces the computation time. Please note that each of these defects has an associated minimum inscribing rectangle that defines their location in the laser image. However, since the laser image and the X-ray image are "almost" perfectly registered (just as the X-ray and color image are also "almost" perfectly registered) these minimum inscribing rectangles also define these defect locations and their extent in the X-ray image as well. Four steps are used to remove these defects from consideration: 1) A transmission value histogram H_d is computed only for regions contained within the defect's (those defects found from laser image analysis) minimum inscribing rectangle. 2) A transmission value histogram H_t of the entire board is computed. Note, this histogram does not include the values of any background pixels. 3) A transmission value histogram of the board without these known defects, i.e., their minimum inscribing rectangles, is computed using H_{xt} - H_{xd}. 4) As the histogram H_{xd} is being computed, each pixel within each of the defect's minimum inscribing

rectangles is a special value that appears in the X-ray thresholded image, $X_T(i, j)$. This special value is -1. Since in the thresholded image a value of 0 is used to represent background and values ≥ 1 are used to represent foreground objects, pixels that have a value of -1 effectively removes them from any future processing since all future processing is aimed at pixels with values greater than or equal to 1.

The sample set that was available at the time this research was conducted was very limited. Examples of boards with large defect areas that were either wane or thin board were unavailable even though it is known that such cases occur rather frequently. Consequently, no examples of this type of processing were available to show the effectiveness of removing known defects before processing the X-ray images. However, Chapter 8 will show the effectiveness of this approach by removing large X-ray defects before processing color images.

7.6.2 Thresholding and Segmenting X-ray images

The purpose of the thresholding operation is to separate pixels that might be a defect from pixels that are clear wood. The thresholds are found using the algorithms described in Section 7.1. Figure 7.8 shows the transmission value histograms of the two boards shown in Figure 7.9. In the histogram in Figure 7.8 (a), the algorithm finds two thresholds that define four intervals. The precise range of each interval, and the area of the board covered by pixels whose transmission values lie in each interval, is given in Table 7.5.

The second interval found, i.e., the one ranging from transmission value 88 to transmission value 193, represents clear wood. The labeling of this interval as representing clear wood is based on the fact that pixels whose transmission values lie in this interval occupy the majority of the area of the board. As a consequence, the threshold T_H that is used to define high-density (low transmission value) components was selected by the program to be 88. That is, any pixel that has a transmission value of 88 or lower is considered to be a pixel that comes from a high-density component. The threshold T_L that is used to define low-density (high transmission value)

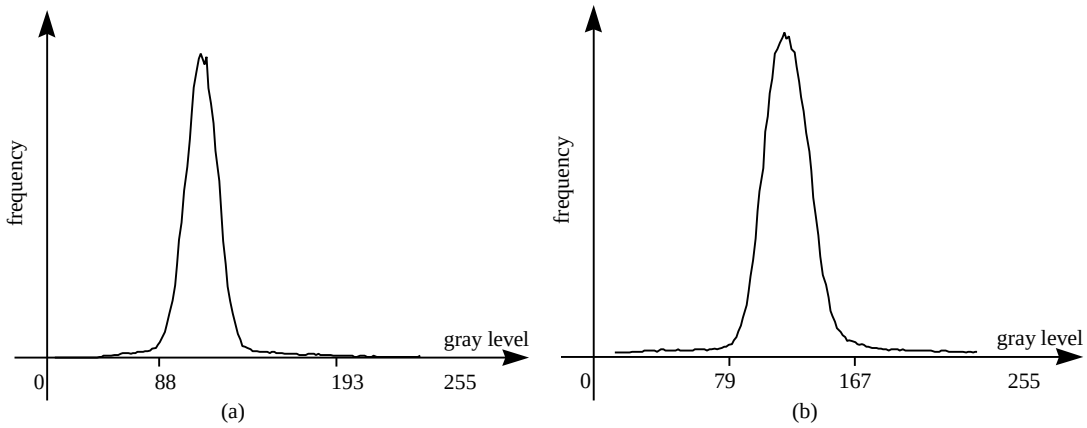


Figure 7.8 Histograms of two X-ray images of boards

was selected by the thresholding program to be 193. This means that pixels whose transmission values are 193 or greater are from low-density components.

Table 7.5 Intervals found in the histogram shown in Figure 7.8 Part (a)

Index	Interval	Area
1	[55, 88]	4345
2	(88, 193)	269850
3	[193,219)	1634
4	[219,249]	784

For the histogram in Figure 10 Part (b), seven intervals were found. The range of these intervals and the number of pixels contained in each are given in Table 7.6. In this instance the fourth interval represents the clear wood since this interval contains most of the pixels within the board image. The thresholds T_H used to identify high-density (low transmission value) components and T_L used to identify low-density (high transmission value) components, were selected by the thresholding program to be 96 and 167, respectively. The other intervals correspond to different high-density or low-density areas detected by the algorithm but these intervals are not utilized. Only the boundaries between high transmission, clear wood transmission, and low transmission are needed for the analysis of the X-ray image.

Figure 7.9 shows the original X-ray images of the boards and the thresholded images that were created for these boards. The histogram of the X-ray image shown in Figure 7.9(a) is the histogram shown in Figure 7.8(a) and the histogram of the X-ray image shown in Figure 7.9(c) is the histogram shown in Figure 7.8(b). Each thresholded image was created using the thresholds T_H and T_L that were described above. For Figure 7.9(b) the T_H = 88 and T_L = 193 while for Figure 7.9(d) the T_H = 96 and T_L = 167. As can easily be observed, most pixels in the thresholded images of both Parts (c) and (d) that were labeled as being high density were correctly identified. Similarly pixels labeled as being low density were also correctly identified. Since these pixels are somewhat hard to spot, the low-density pixels all lie in the upper right corners of Parts (c) and (d) of Figure 7.9. Both are also separated from pixels that are believed to be clear wood. Hence, pixels from the major defect components (e.g., knots) are separated from the clear wood in the X-ray images.

Table 7.6 Intervals found in the histogram shown in Figure 7.8 (b)

Index	Interval	Area
1	[0, 40]	747
2	(40, 79]	2291
3	[79,96)	2436
4	[96, 167)	228830
5	[167,197)	2220
6	[197,212)	799
7	[212,255]	1190

However, as one might notice in the thresholded image shown in Figure 7.9(a), some small low-density (brighter) components that are inside a large high-density (darker) component do not appear in the thresholded image shown in Part (b). This happened because the total area of low-density components is too small to form any significant peak in the transmission value histogram of the board (See Figure 7.8[a]). As a consequence, the algorithm picked the threshold T_H to be 193. This was not the best threshold to select. The threshold value for T_H is much higher than it should have been in relation to the average transmission value of clear wood. In contrast, for the transmission value histogram of Part (b) of Figure 7.8, the algorithms selected the threshold T_H to be 167. This value is very close to the optimum and, hence, a much better

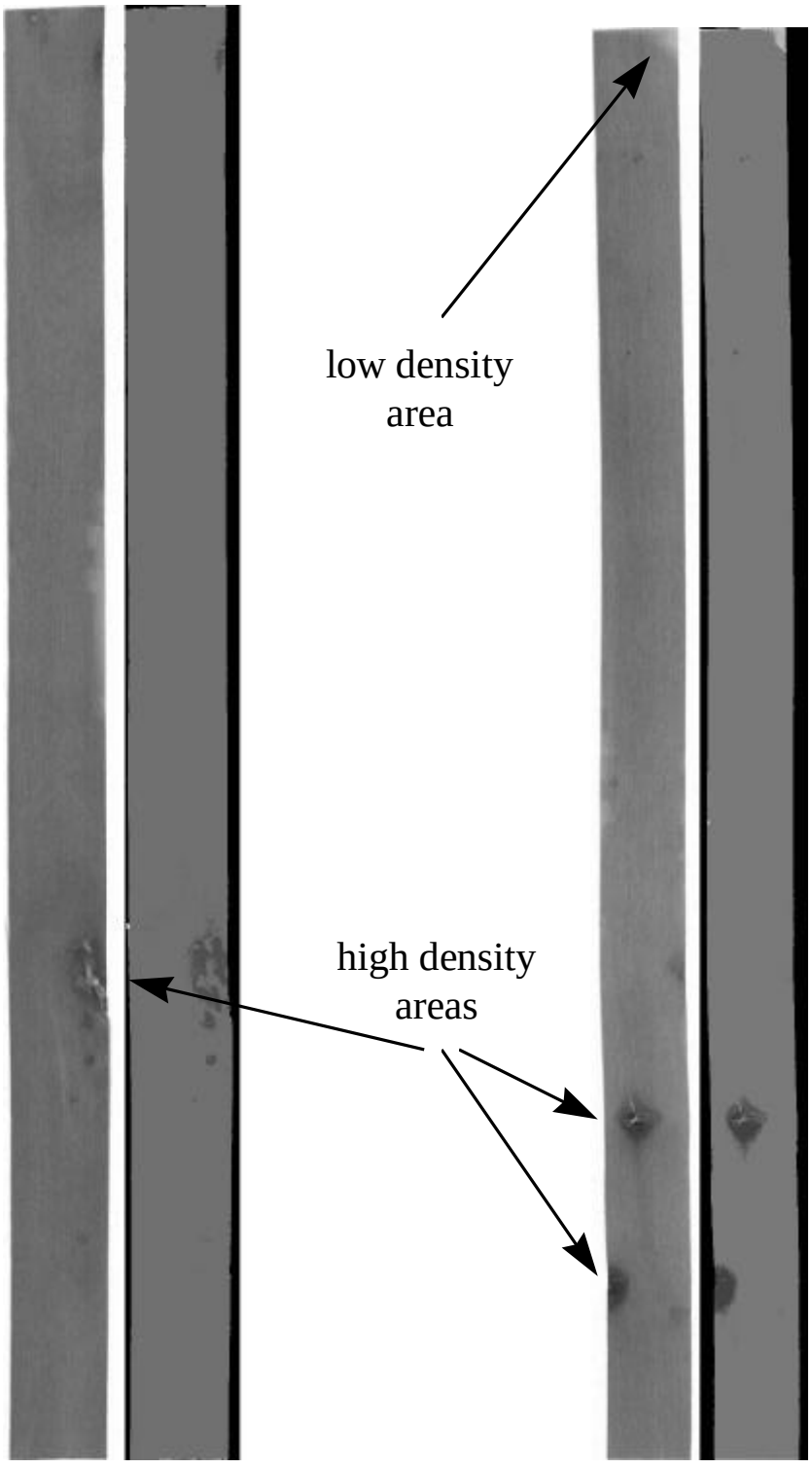


Figure 7.9 Original X-ray images (a), (c) and their thresholded images (b), (d)

value to select. Fortunately, the small components that were incorrectly excluded in Figure 7.1(b) often are small worm holes or small checks that lie inside a large knot. Since large knots are easily segmented for clear wood and are always removed, any small checks or worm holes within large knots are also removed whether they were correctly segmented or not. In addition, it is worth noting that small worm holes can usually be correctly segmented from clear wood during the analysis of the color images.

After thresholding, a connected-component labeling is applied to the thresholded image, $X_T(i, j)$, to form the connected-component labeled image, $X_C(i, j)$. Figure 7.10 shows the original images, i.e., the same X-ray images that appeared in Figure 7.9, and their connected-component labeled images. The different connected components are show as various shades of pink. The pink shade used was determined by the value of the label that was given to the connected component by the connected-component labeling operation. If the label assigned to the clear wood component, an assignment that is determined dynamically during the connected-component labeling operation, is smaller than the labels of other components, then the clear wood component will appear darker than other components when it is displayed. This is the case in Figure 7.10(d). Moreover, the clear wood of one board may appear to be darker than the clear wood of another board because of the dynamic assignment of label values during the labeling process. See, for example, the comparison between the clear wood components of Figure 7.10(b) and (d).

The connected-component labeled image, $X_C(i, j)$, is the output of low-level processing. The connected-component labeled image contains components that possibly represent all or part of a possible defect. Pixels in each component are marked with a unique number-component label. The connected-component labeled image is the input to the next level of processing--the intermediate-level of processing.

7.6.3 Measurement Extraction from X-ray images

The first step of the intermediate-level processing of X-ray images is to extract a number of measures from each component in the connected-component labeled image, $X_C(i, j)$, that has

not been identified as being clear wood. These measures are then used to eliminate small components and to merge similar components. The measures are as follows:

- The area of the component;
- The height and width of minimum inscribing rectangle of the component;
- The average transmission value of the component;
- The upper left and lower right corners of the minimum inscribing rectangle of the component;
- The perimeter, i.e., the total number of pixels on the boundary, of the component;
- The touch_edge, i.e., the total number of pixels on the boundary of the component that touch an edge of the board.

The initial components obtained from the connected-component labeling operation usually contain many small and meaningless components. Consider for example, the enlarged insertions to Figure 7.10. Since small component elimination is a simple operation and since it is very effective, it is performed first. Then, a component merging operation is performed. Both of these operations use some of the measures defined above.

Experiments have shown that a small component elimination operation, an operation that eliminates components containing one or two pixels, can reduce the total number of components that must be considered by the later processing by up to 70 percent. The merging operation typically reduces the number of components that must be considered to less than 50 in most X-ray images. To illustrate this, the columns of Table 7.7 represent the various sizes of components that are counted, i.e., one or two pixel components, components having pixel counts between three and 20 and, finally, components that have pixel counts greater than 20. The second row of this table gives the number of components by size that are in the connected-component labeled image of the board shown in Figure 7.9(a), i.e., in the connected-component labeled image shown in Figure 7.9(b). Similarly, Row 3 of this table provides the same information about the board shown in Figure 7.9(c), i.e., the connected-component labeled image of Part (d) of this figure. As will be observed from these two rows of the table, a very large percentage of the total components

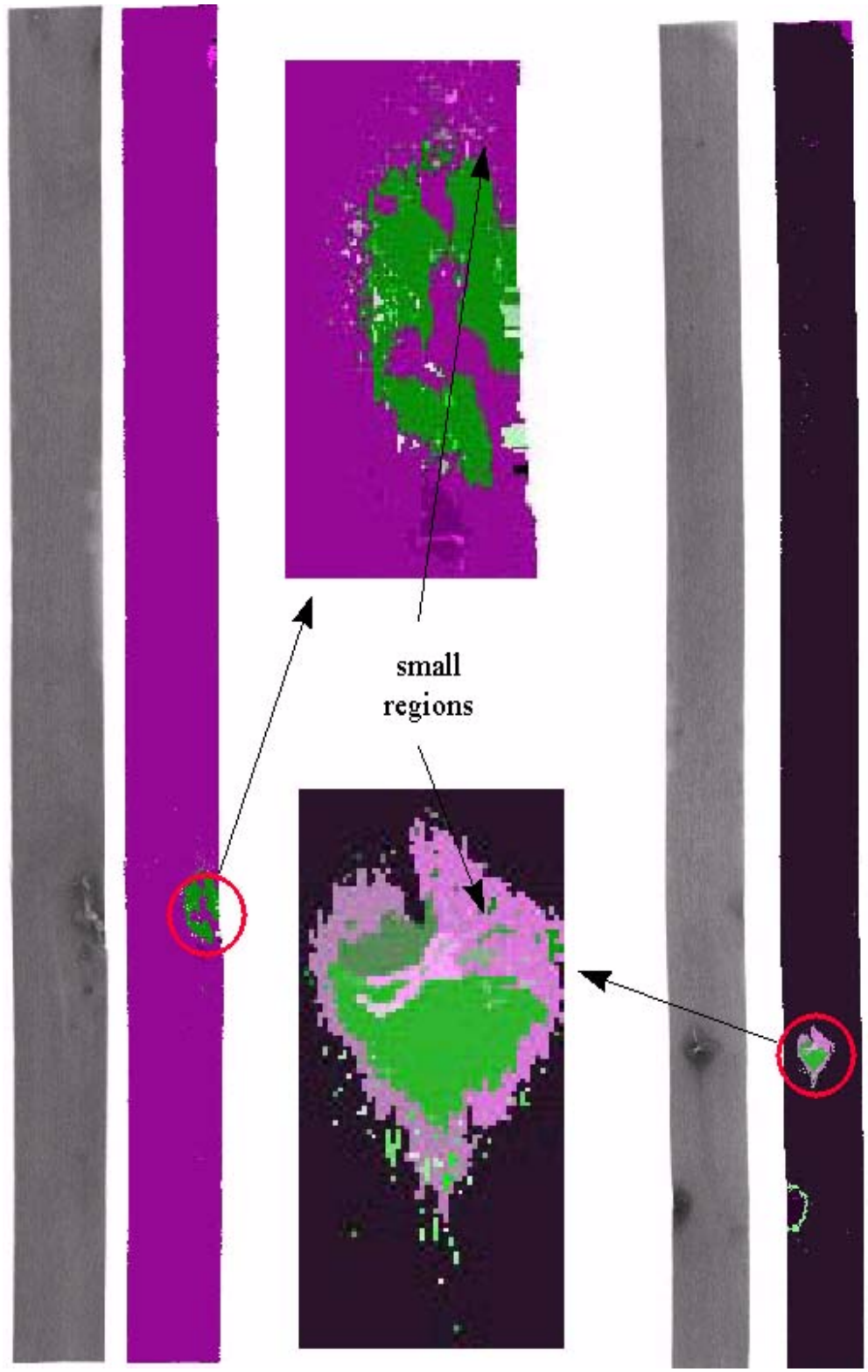


Figure 7.10 The labeled images containing many small components

found contain only one or two pixels. Removing these components through small component elimination and component merging will result in markedly reduced computational complexity. Table 7.8 shows the number of one-or two-pixel components that remain after small component elimination. As can easily be observed, all these small components have been eliminated or merged with neighboring components. Note that the number of component from 3-20 has increased in Table 7.8. This increase is because some of the one- or two-pixel components have been merged with neighbors to create larger regions, thus adding to the number of components in the range of 3-20 pixels. Table 7.9 shows the number of components left after the component merging operation is completed. As can easily be seen, the total number of components that must be considered after this operation is completed is very small in relation to the number of components that were originally found in Table 7.7. In fact, the total number of components that remain in either image is less than 15 while the number of initial components found in each image numbered in the several hundreds.

Table 7.7 Initial components found in the images of Figure 7.9

Component Size (in pixels)	<= 2	3 – 20	> 20	Comments
Number of Components	227	81	11	Image (a)
Number of Components	239	75	13	Image (b)

Table 7.8 Components distribution after eliminate small components

Component Size (in pixels)	<= 2	3 - 20	> 20	Comments
Number of Components	0	126	11	Image (a)
Number of Components	0	128	13	Image (b)

Table 7.9 Components distribution after component merging

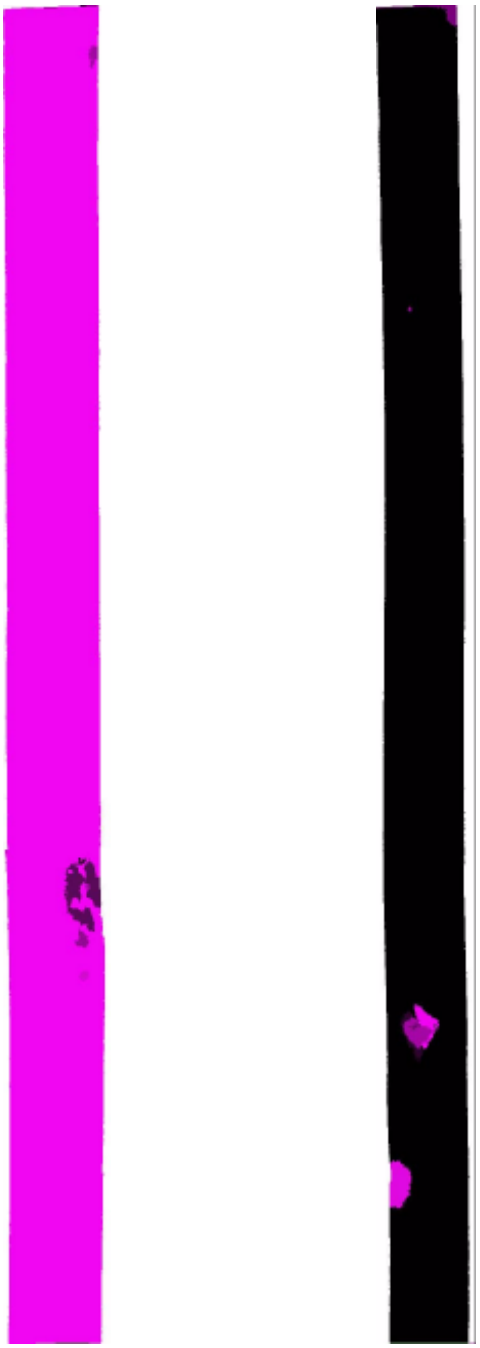
Component Size (in pixels)	<= 2	3 - 20	> 20	Comments
Number of Components	0	3	10	Image (a)
Number of Components	0	1	7	Image (b)

However, even after component merging, there still may exist small components that contain only a few pixels, e.g., less than 20. Such small components are typically caused by noise, or small worm holes, i.e., real defects. Therefore, these small components cannot be neglected since some might actually be defects. There are two options for handling the small components. These are: 1) If the area of a component is smaller than a pre-determined threshold, then mark this small component as clear wood. Because defects are smaller than a threshold, e.g., five pixels in size can be neglected based on the secondary manufacturer's specifications. 2) Process these small components as normal components, either in the first processing or in the second processing of the X-ray images. The first option was selected for this system because most manufacturers' specifications allowed for very small defects to appear in the finished product. Consequently, the intermediate-level processing module applies an adjustable threshold (nine-pixel is used in this research) to its final output. Any components with an area smaller than the threshold will be marked as clear wood and will not be forwarded to the high-level of processing.

After the component elimination and the component merging have been completed, the components that remain will be those forwarded to the high-level processing module. For example, the final components that remain for the two X-ray images of Figure 7.9 are displayed in Figure 7.11. Clearly, the major defects are all separated from clear wood.

After all the components with similar measures are merged, each newly merged component must have its associated measurement values recomputed. That is, the area of the component, the height and width of its minimum inscribing rectangle, its average transmission value, the upper left corner and lower right corner of its minimum inscribing rectangle, its perimeter, and its touch_edge value must all be recomputed. At the same time a series of new measures, ones that can be immediately derived from the above measures, are computed for each of the remaining components. These new measures are:

- The center of the mass of the component
- The compactness of the component
- The elongatedness of the component



(a) (b)

Figure 7.11 Final results of intermediate-level of processing for images in Figure 7.8

After these new measures have been computed from each of the remaining components, these components together with their associated measurement sets and the connected-component image $X_C(i, j)$ are all input to the high-level processing module.

The low-and intermediate-level processing modules have been applied to more than 300 images of red oak, maple, and pine boards. In almost all cases, the segmentation results detect the majority of defects such as knots, mineral streak, holes, and split/check that appeared in these boards. This suggests that the dynamic methods used to do the segmentation are, at least in part, species independent since they seem to work about equally well on all the species to which they have been applied [FOR96].

7.6.4 Defect Identification

In the high-level processing module, each of the components forwarded to it by the intermediate-level processing module is sent to each of the expert defect-recognition modules that have been created for this system. The expert defect recognition modules include ones for identifying members of the *knot* class, members of the *mineral streak* class, members of the *high-density* class, members of the *hole* class, members of the *split/chec*k class, and members of the *low-density* class. For each component, each of the expert defect recognition modules returns an approaching degree as to how close the component resembles the defect class the module was created to identify.

The approaching degrees between each component and defect classes are given in Table 7.10 for the board shown in Figure 7.9(a). Each row describes one component, its approaching degree between defect class knot, mineral streak, high-density, split/check, hole and low-density. The maximum approaching degree is shown in bold face. The final result or decision, which is based on the maximum approaching degree, is given in the last column. If the maximum approaching degree of a component is less than 0.4, the component is considered to be clear wood.

After all the components have been processed by all the expert recognition modules, i.e., after each component has been assigned an approaching degree to all the defect classes for which there is a defect recognition module, the final defect label assigned for each component is made by the decision-making module. Subsequently, the post-processing module merges the same kind of

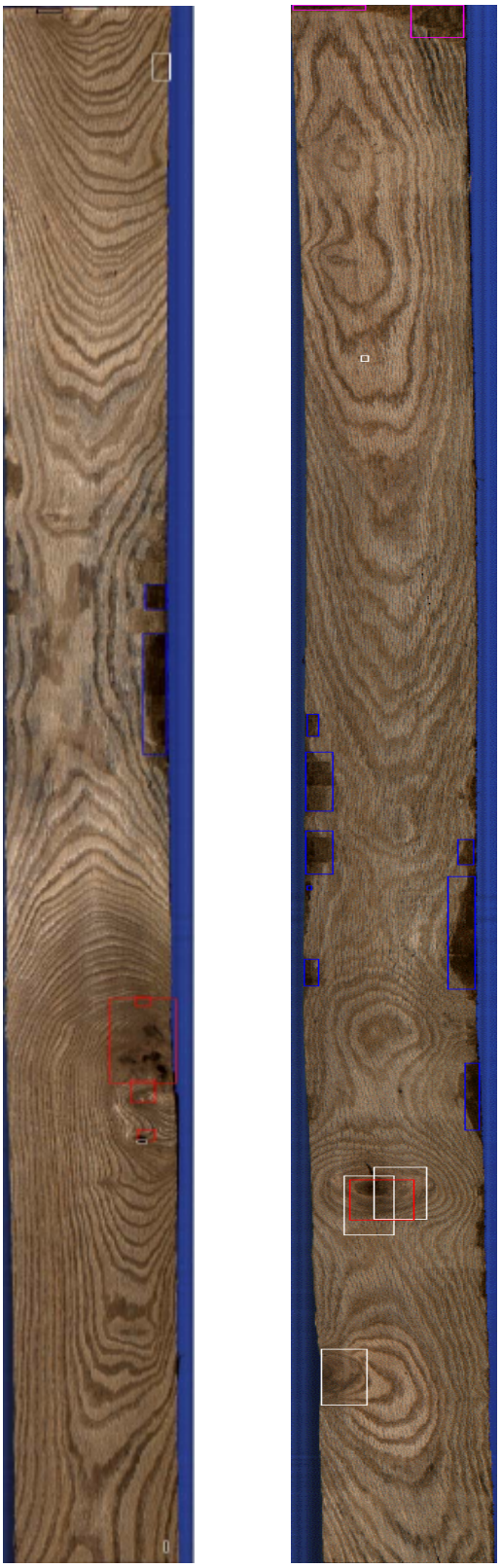
defects and checks each component to see if it lies inside another component. If it does, the defect is neglected unless the larger defect is a high-density or low-density defect. If it is not, the component is added to the final defect list.

The above identification results for the two boards shown in Figure 7.9 are given in Figure 7.12. Figure 7.12(a) shows the results obtained from the board image shown in Figure 7.9(a) and Figure 7.12(b) for the image in Figure 7.9(b). For the image in Figure 7.9(a), four components were accurately identified as knot (marked with red rectangles) and all of them are located in the middle of the board. One component is identified as a hole at the top of the board and is marked by a black rectangle. From the X-ray image (see Figure 7.9), it is easy to see the brighter spot at the top. Two components are correctly detected as high-density components marked by white rectangles, one at the top and one in the middle.

Table 7.10 Approaching degrees for the components in image shown in Figure 7.9, Part (a)

Component	Knot	Mineral	High density (HD)	Hole	Split/ Check	Low density	Final result
1	0.0	0.0	0.0	**0.50**	0.44	0.0	hole
2	0.0	0.0	0.0	0.23	**0.35**	0.0	wood
3	0.39	0.31	**0.59**	0.0	0.0	0.0	HD
4	**0.48**	0.41	0.39	0.0	0.0	0.0	knot
5	**0.55**	0.29	0.41	0.0	0.0	0.0	knot
7	**0.57**	0.24	0.1	0.0	0.0	0.0	knot
8	0.34	0.19	**0.41**	0.0	0.0	0.0	HD
9	**0.33**	0.19	0.0	0.0	0.0	0.0	wood
10	**0.46**	0.28	0.0	0.0	0.0	0.0	knot

For the image in Figure 7.9(b), two components on the top part of the board, marked with pink rectangles, are identified as low-density components. A small component in the top part,



(a) (b)

Figure 7.12 The defect recognition results obtained from the X-ray image analysis

marked by a white rectangle, is also correctly identified as a high-density component. Wane areas inside blue rectangles were positively identified in the previous chapter. In the lower part, one component is correctly detected as a knot while the other two are identified as high-density regions that are overlapped with the knot. These two components are also correctly identified since they are areas of high density adjacent to the center of the knot, and there are no color variations in these areas. Finally, another component, close to the bottom, is correctly identified as a high-density area since it is a knot that is not visible from the color image.

7.7 Summary

The analysis of the X-ray image of a board is aimed at identifying knots, holes, mineral streak, low-density defects (such as internal voids), splits, and high-density objects based on the density variation of these defects. This analysis consists of low-level, intermediate-level, high-level, and a post-identification processing. The goal of the low-level processing is to find image components that might contain a defect. The output of the low-level processing module is then fed into the intermediate-level processing module, where measures that characterize each of these components are computed. In this intermediate-level processing, small components are eliminated and similar components are merged. The final components and their associated measures are then sent to a high-level processing module, where each component is examined with additional measures extracted from the color image. If a component resembles a defect class, the component is then assigned the label of the defect class. The post-identification processing merges those components that are adjacent to each other and have the same defect label. Finally, all components which have been assigned a defect label are put on a final defect list.

Once the X-ray image and the laser image are processed, most defects are detected, but not all of them. Defects such as stains and smaller, more subtle, defects are not detectable at the low-level processing of the X-ray image. Hence, the next chapter will deal with the detection of these defects by processing the color image.