AN ABSTRACT OF THE THESIS OF

Yuan Zhong for the degree of Doctor of Philosophy in Forest Products presented on November 1, 1994. Title: Image Segmentation for Defect Detection on Veneer Surfaces.

Abstract approved: Redacted for privacy

James W. Funck

Machine vision is widely used in scientific areas and non-wood using industries, but the extreme variability of wood has limited its adoption by forest products industries. However, it is now becoming a key factor in further automation of the forest products industry. As a very important part of machine vision, developing image segmentation algorithms that can be used for wood products is an ambitious undertaking. The focus of this research was to adapt existing and develop some new segmentation algorithms which could be used to detect defects on veneer surfaces.

Nine algorithms covering three segmentation technique categories were explored. Three existing edge detection algorithms were modified for use on veneer images, and four existing thresholding algorithms were adapted in both global and local versions. Two new region extraction algorithms were developed specifically for defect detection on veneer surfaces.

The performances of these nine algorithms were tested and compared under the combinations of two camera resolutions (5-bit and 8-bit), three color spaces (RGB, Lab, and gray-scale), and seven surface features (clear wood, blue stain, loose knot, pitch pocket, pitch streak, tight knot, and wane). Ten sample images for each of seven surface features on Douglas-fir veneer [Pseudotsuga menziesii] were used. Ten measures were proposed for performance evaluation. A multi-factor factorial ANOVA was used in the performance tests and comparisons.

The best combinations of camera resolution and color space for each of the algorithms were determined. The 5-bit and 8-bit camera resolutions were not significantly different for the three edge detection and two region extraction algorithms, but the 8-bit camera resolution was better for all but one of the thresholding algorithms. That exception was the global Otsu thresholding algorithm, for which the 5-bit camera resolution was better. The RGB color space was the best for all algorithms. Overall, the two region extraction algorithms were the best. Under the best combination of factors, those two algorithms provided the highest defect detection accuracies of 91% for pitch streak samples and over 95% for loose knot, tight knot, and pitch pocket samples. These results were accomplished while still providing clear wood accuracies of over 95%. The one performance exception was blue stain, for which no satisfactory algorithm was found.

Image Segmentation for Defect Detection on Veneer Surfaces

by

Yuan Zhong

A THESIS

submitted to

Oregon State University

in partial fulfillment of
the requirements for the
degree of

Doctor of Philosophy

Completed November 1, 1994
Commencement June 1995

Doctor of Philosophy thesis of Yuan Zhong presented on November 1, 1994

APPROVED:

Redacted for privacy

Major Professor, representing Forest Products

Redacted for privacy

Head of Department of Forest Products

Redacted for privacy

Dean of Graduate School

I understand that my thesis will become part of the permanent collection of Oregon State University libraries. My signature below authorizes release of my thesis to any reader upon request.

Redacted for privacy

Yuan Zhong, Author

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to my major professor, Dr. James Funck, for his guidance, encouragement and support through the course of my graduate study. Without him keeping me off of the river and out of the brush, I would have never completed this thesis.

I would like to thank my committee members, Dr. Ronald Mohler, Dr. David Butler, Dr. Charles Brunner, and Dr. Eldon Olsen who gave their time to give suggestions, read my thesis, and be present at my defense.

Thanks also goes to Mr. Alberto Maristany. His help and patience made possible my work with the hardware and software in the laboratory.

I am grateful to Mr. Johannes B. Forrer for his valuable suggestions. Talking with him is always helpful.

I am deeply indebted to my husband, Yimin Zeng. He is the one who gave the love, care, support and help in my daily life and studies.

I would like to express my deep gratitude to my mother for her love, help, and encouragement.

TABLE OF CONTENTS

Page

LIST OF FIGURES

Page

Page

Page

LIST OF TABLES

Page

IMAGE SEGMENTATION FOR DEFECT DETECTION ON VENEER SURFACES

CHAPTER 1 INTRODUCTION

Machine vision largely deals with the acquisition and computer analysis of images. It has already been widely used in many areas. Examples include the well known CT (computer tomography) used in medical diagnosis, satellite image analysis for weather forecasting and forest fire inspection, and robots in work places where the conditions are hazardous for a human operator. Whatever the application, machine vision always includes three basic parts: image acquisition, image processing, and image analysis. Image segmentation is one of the most important elements in image processing component because it is at this step that objects or other entities of interest are extracted from an image.

1.1 MACHINE VISION AND ITS APPLICATIONS IN THE FOREST PRODUCTS INDUSTRY

Substantial economic losses occur in some wood products processes due to the lack of a system that can accurately detect, locate, and classify defects or features on wood surfaces. Therefore, advancements of machine vision systems are a key to further automation in forest product industries. Automation utilizing machine vision in the forest products industry is an ambitious undertaking. Applications of a few automatic systems equipped with machine vision have been attempted in the forest products industry, for instance, the automatic sorting chain in a sawmill. They have not been popular due to unsatisfactory accuracy and reliability. As the wood resource becomes increasingly limited and of lower quality, more technology is required. These technology improvements will depend on machine vision.

There are many application areas for machine vision. Log breakdown into lumber or veneer has significant potential for optimization gains if the defect distributions are known. Combining machine vision with conventional profile scanning in the board edging and trimming processes can raise the lumber grade. Automatic plugging and patching in a plywood plant can relieve human labors. Automatic sorting can increase speed and accuracy. For instance, Fahey (1987) found that in green veneer sorting, about 20 percent of the AB veneers were downgraded into CD veneers and 3 percent of the CD veneers were upgraded into the AB grade. The money lost in the downgrading is significant because AB veneer is about twice as valuable as CD veneer. Szymani and McDonald (1981) reviewed a number of approaches (ultrasound, x-ray, infrared, and visible light) for interpreting lumber surface characteristics and determined that no one approach was superior. However, they felt that systems employing visible light had certain advantages that would yield higher reliability.

As mentioned before, machine vision consists of three basic parts: image acquisition, image processing, and image analysis. Image acquisition involves obtaining a representation of an object and in an optical system has three elements: a sensor, lighting, and frame-grabber. Each element has an impact on the properties and quality of the acquired image. Image processing is the part in which various mathematical techniques are applied to an image in order to identify the features of interest. The phrase image processing is used in its broadest sense to include many data processing techniques such as enhancement, transformation, restoration, compression, and segmentation. Image analysis is the part in which quantitative assessments are made of interesting features on the basis of information in the segmented image.

1.2 IMAGE SEGMENTATION

Image segmentation is the subdivision of an image into its constituent parts or objects. In the other words, it is a processing or grouping of information in an image

into units corresponding to similar objects. Grouped units are homogeneous with respect to one or more image characteristics such as intensity, color or texture. Image segmentation is one of the most complex and difficult tasks in image processing, even though the identification of objects in an image comes naturally to the human observer. In many application areas, the quality of the final output depends largely on the quality of the segmented output.

Segmentation algorithms are generally based either on the concepts of similarity or discontinuity of image features such as color, intensity, tone, or texture. Hundreds of segmentation methods or algorithms have been developed for different types of images and different applications. However, there is no single method which can be considered good for all images nor are all methods equally good for particular types of images. Almost all image segmentation techniques proposed so far are ad hoc in nature. Also, it is not easy to compare different image segmentation algorithms because only a small amount of effort has been placed on how to measure segmentation error other than the simple criteria of percentage of pixels misclassified.

1.3 STUDY OBJECTIVES

The goal of this project was to find, adapt, and improve or develop image segmentation algorithms which could be used to detect defects on Douglas-fir [Pseudotsuga menziesii] veneer surfaces.

The specific objectives of this study were twofold: (1) to find and adapt or develop a variety of segmentation algorithms for wood products applications, and (2) to develop a systematic method to quantitatively evaluate the performances of those algorithms. The algorithms were chosen to represent different categories of image segmentation techniques, that is edge detection, thresholding, clustering, region growing, and splitting and merging, so that there was a better chance of finding good segmentation algorithms. If possible, the algorithms also needed to have gray-scale and color versions because previous research work had shown that color information was very important for some surface features such as stain and pitch streaks.

Therefore, in the performance evaluation, gray-scale, RGB, and Lab color spaces were considered to compare their capabilities with different algorithms and different defect types. Also, two camera resolutions (5-bit and 8-bit) were considered because camera resolution can influence segmentation results.

CHAPTER 2 LITERATURE REVIEW

Segmentation is the first step of image analysis which aims at either a description of an image or classification of an image. In many applications, segmentation quality determines the quality of final performance of the machine vision system. A wide variety of image segmentation techniques, methods and algorithms have been proposed for various purposes and types of images. In this chapter, segmentation techniques and some specific methods or algorithms will be discussed.

2.1 INTRODUCTION TO SEGMENTATION

Image segmentation is the process of partitioning an image into regions that are meaningful for their correspondence to visual objects or their parts. Each segment has a homogeneous property such as intensity or gray-level, color, texture, and other spatial or geometric characteristics. Accordingly, almost all segmentation techniques are based on two basic image properties, discontinuity and similarity of image features. When based on discontinuity, an image is partitioned when abrupt changes in certain features occur. The principal methods in this category detect boundaries and edges in an image and are therefore called edge detection techniques. In contrast, when based on similarity, image pixels whose properties are similar are grouped together to form regions. The principal approaches in this category are thresholding, clustering, region growing, and region splitting-and-merging.

In segmentation, the algorithms commonly use a variety of approaches, such as statistical techniques, geometrical analysis, characteristic feature analysis, information theory, filtering, graphical searching, fuzzy logic theory, dynamic programming, and expert systems. In spite of significant efforts over a long period of time, there is still no "perfect" segmentation algorithm available. According to Marr (1982), there are two reasons why the theory and practice of segmentation remain primitive. First, in most cases the exact goals of segmentation are impossible

to precisely formulate. Second, regions that have segmentation importance do not always have any particular visual distinction. Therefore, for different purposes and types of images, different algorithms have been developed with varying degrees of performance. Since there is no single method which can be considered good for all images nor are all methods equally good for any particular type of image, discussions about algorithms in most segmentation surveys must be considered only as references and not direct comparisons.

Fortunately, some of the shortcomings of segmentation can be partially rectified at the high-level processing stage of image analysis by using more domain-specific knowledge. Recently, various expert systems for image processing have been proposed to develop image understanding systems needed for the creation of intelligent machines with visual perception capability (Chen et al., 1989; Draper, 1989; Levine, 1985; Matsuyama, 1989; Shu, 1987; Stansfield, 1986). In such a way, the accuracy of image analysis can be increased.

2.2 INTRODUCTION TO COLOR

Color is a perceptual phenomenon related to human response to different wavelengths in the visible electromagnetic spectrum. It is a function of incident light, absorption selectivity of an object, and sensitivity range of a receptor. It has long been recognized that the human eye can only detect in the neighborhood of one or two dozen intensity levels at any one point in an complex image due to brightness adaptation but can discern thousands of color shades and intensities (Gonzales and Wintz 1987; Overheim and Wagner 1982).

There are three psychological attributes that are generally used to represent color, namely hue, saturation and intensity. To quantitatively describe color, the C.I.E. in 1931 (Commission Internationale de L'Eclairage, 1986) defined the tri-stimulus standard consisting of three monochromatic primaries red (R), green (G), and blue (B) (referred to as the CIE RGB color space). Since that time, quite a few color models or color spaces have been established to quantify and measure color.

Examples include CIE XYZ, CIE Lab, CIE Luv, H-V-S, NTSC YIQ (Commission Internationale de L'Eclairage, 1986), and Ohta's O_1-O_2-O_3 etc (Ohta, 1980). Each color space has its strengths, weaknesses, and suitable application areas; no single space has been proven universally superior. Since the review of color literature is beyond the scope of this thesis, more general information about color, color spaces, and transformations between the various spaces may be found in references such as Hunter and Harold (1987), Billmeyer and Saltzman (1981), and Wyszecki and Stiles (1967).

Research has shown that the use of color information can be very important in image segmentation, since intensity is the only available information from a monochrome image. In contrast, a color image also provides chromaticity information that can overcome some of the difficulties in image segmentation. Thus, applications involving color imaging are becoming increasingly prevalent. In image processing, the data describing a color image is usually about three times as large as that for a gray-scale image. This does present a greater burden on the image processing software. However, the rapidly increasing speed of the computer compensates for that load and makes color image processing feasible.

2.3 OPTICAL PROPERTIES OF WOOD SURFACES

For defect detection on veneer surfaces using image segmentation techniques, it is essential to understand the optical properties of wood surfaces. The way the color of wood is perceived by humans appears to depend on three main characteristics: dominant wavelength, brightness, and purity. Using CIE color measures to quantify natural wood colors, researchers have found that wood can be qualitatively described as light reddish brown or yellowish gray. The dominant wavelength is roughly 580 nm. Research has shown that the color of wood varies from one type of defect to another (such as tight knot, loose knot, pitch streak, pitch pocket, wane, stain, and decay), from defects to clear wood, from defect to defect of the same type, from sapwood to heartwood, and from latewood to earlywood.

However, this color variability is in brightness and purity, not in the dominant wavelength (Gray, 1961; Lakatosh, 1966; Moon and Spencer, 1948; Conners et al., 1985).

There are a number of factors affecting the optical characteristics of wood surfaces, such as fiber angle, orientation of incident light relative to the wood surface, surface roughness, moisture content, sapwood and heartwood in the same species, wood cut from different portions of the tree, and different species. (Conners et al., 1985; Brunner et al., 1989).

Sullivan (1967) noticed that wood color is basically two-dimensional within the standard CIE color space, and this is further reinforced by investigations of Conners et al., (1985), Funck et al., (1987), Butler et al., (1989, 1991), and Forrer et al., (1988, 1989) in their research of defect detection of wood surfaces. The color properties of wood and defects have been discussed in more detail by Conners et al., (1985), Brunner et al., (1989, 1992, 1993), and Maristany et al., (1991).

While image analysis technology has been extensively used in areas such as metallurgy, medicine, and biology, only limited applications have been reported in wood science and the wood industry. The great variability of wood surfaces becomes a significant obstacle in defect detection using optical imaging. Like a finger-print, no two pieces of wood are the same. Each has its own unique grain pattern and defect distributions. The optical properties of the wood surface, such as reflection and color, vary between and within different parts of wood surfaces. In softwoods, the latewood is usually darker and smoother in the same piece than earlywood. However the latewood in one piece may be lighter than earlywood in another piece and the latewood area in a species may be rougher than earlywood areas in a different species. The reflection and color of defects exhibit significant variation from part to part and species to species.

2.4 SEGMENTATION METHODOLOGY

In general, segmentation techniques can be divided into three categories: (1) thresholding and clustering, (2) region extraction, and (3) edge detection. In reality, the thresholding and clustering techniques belong to the region extraction category. However, because they were used earlier and have quite a large number of algorithms, they are usually considered as a separate category. It should be kept in mind that these are only general classifications. Some algorithms combine several techniques together. For instance, the spatial clustering technique is a combination of two or three basic techniques. Consequently, different reviewers classify segmentation techniques and methods in different ways.

2.4.1 Thresholding and Clustering

Thresholding and clustering techniques are the most popular approaches to image segmentation. They are good for images having more or less clear modals or clusters.

2.4.1.1 Thresholding

The characteristic feature thresholding technique is the earliest one used. It is comparatively simpler than edge detection and region-dependent techniques, and it is more immune to noise than edge detection techniques. It also gives closed boundaries, although it is sometimes necessary to smooth out some of the noisy boundaries. However, since most of the methods in this approach are based on the assumption that different classes of segments of an image are represented by distinct modes in the distribution of suitably chosen features extracted from the image, the technique will fail when that assumption is not true. Another drawback of thresholding is the difficulty in using color information.

Using Weszka's classification scheme (1978), thresholding methods can be divided into three types depending on the functional dependencies of threshold operators: global, local and dynamic. Dynamic thresholding is very similar to local thresholding except that it finds a threshold for every pixel in an image instead of for the subimages or windows of an image. Therefore, Taxt et al., (1989) included it with local thresholding and so does this segmentation survey. The relationships of representative thresholding techniques are shown in Figure 2.1.

2.4.1.1.1 Global thresholding

Global thresholding considers information from the whole image and uses it to find proper threshold values. It is one of the basic thresholding techniques. When an image consists of only a few objects and background (i.e., a two-class, tri-class, or quad-class image) and the objects have a certain degree of difference in characteristic values than the background, global thresholding is the best and easiest segmentation method. There are two types of global thresholding algorithms: (1) histogram-based thresholding, and (2) nonhistogram-based or spatial information-based thresholding (Glasbey, 1993).

Histogram-based global thresholding is the most commonly used approach. In these types of algorithms, the threshold values are selected by histogram analysis. A histogram is computed for the image intensity values, color values or any other feature values and analyzed to determine a threshold setting to separate the object from its background. Most of the histogram-based algorithms are simple to understand and easy to implement. However, all the histogram-based algorithms have a common drawback because they take into account only the histogram information. As a result, such algorithms are only good at thresholding images which contain significant degrees of contrast between the objects and its background and in which the objects do not occupy too small a portion of the image.

Various histogram-based algorithms have been developed. Prewitt and Mendelsohn's mode method (1966) uses thresholds that are simply chosen at the

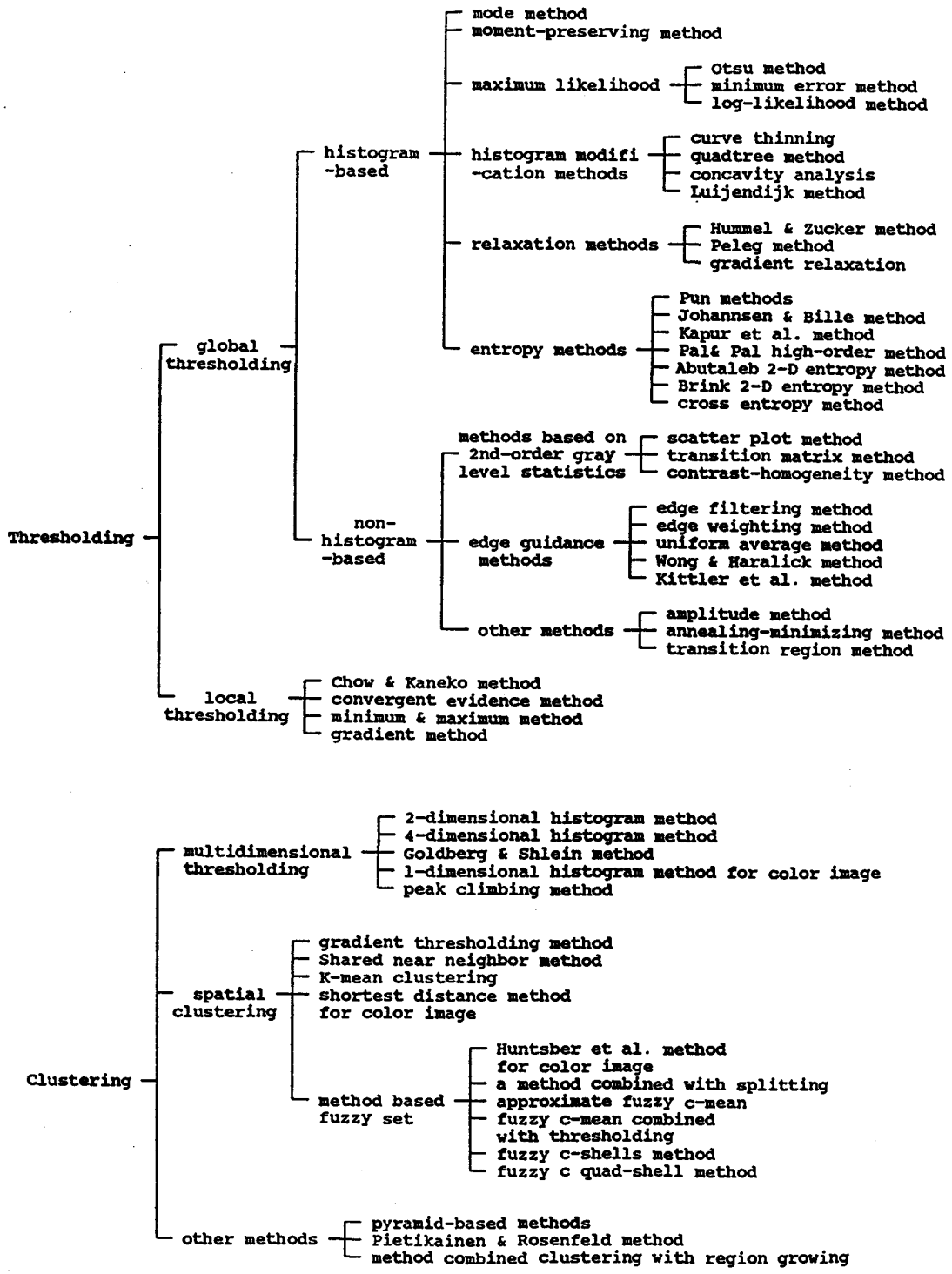

Figure 2.1. Examples of thresholding and clustering algorithms.

valleys on the gray-level histogram of images. Since this method requires modes in a histogram, it can not be applied to images with extremely unequal peaks or to those with broad and flat valleys. Tsai (1985) suggested the moment-preserving method in which the thresholds were selected in such a way that the binary image had the same first three moments as the gray-level image.

Several approaches utilize the maximum likelihood-based method on population mixture models under the assumption of a normal distribution. It is claimed that these methods are feasible even for images with unimodal histograms. Otsu's method (1979) is based on discriminant analysis and also minimizes the mean square errors between the original image and the resultant binary image. The threshold is determined by maximizing the between-class variance of gray-scales. The minimum error method by Kittler and Illingworth (1986) minimizes a criterion related to the average pixel classification error rate. The method by Kurita et al., (1992) maximizes the log-likelihood of the joint distribution and uses dynamic programming to search for optimal thresholds.

For histograms without clear modes, a common procedure is the modification of a histogram into one with deeper valleys and sharper peaks. Rosenfeld and Davis (1978) used a curve thinning scheme to sharpen peaks on an image histogram. The quadtree method by Wu et al., (1982) subdivides the entire image into quadrants if its gray-level standard deviation is high so that the gray-level histogram of the resulting image will have sharper peaks and deeper valleys because of the homogeneity of each block. Rosenfeld and Torre (1983) suggested that a threshold could be determined by analyzing the concavity structure of the histogram and a good threshold could be found at the root of the "shoulder". Luijendijk (1991) introduced a set of new histograms based on the count of 4-connected regions that had a better affiliation with the ideas of histogram-based methods for automatic threshold selection than conventional gray-level histograms.

The relaxation operation was primarily developed to reduce the local ambiguity in interpreting objects in a scene by iterative parallel operations (i.e., relaxation operations). This process is used for thresholding to enhance the original image so

that the resulting histogram will be better suited for choosing an appropriate threshold. Hummel and Zucker (1977) suggested the probability updating method in which the initial probability estimates of the light and dark classes at each pixel were iteratively updated according to a predefined updating equation. For line enhancement, Peleg (1980) proposed a Bayesian-based probability updating scheme in which the probabilities at neighboring nodes of a graph with a vector of probabilities associated with its every node were used iteratively to update the probabilities at a given node based on statistical relations among node labels. Bhanu and Faugeras (1982) suggested using an iterative gradient relaxation operation to separate a unimodal histogram into a bimodal histogram.

The entropic method includes a large group of thresholding methods where the optimal threshold is obtained by applying information using the theory of entropy. Pun (1981) proposed an algorithm in which the histogram was classified by its entropy and the threshold was determined as a function of an anisotropy coefficient. Johannsen and Bille (1982) used the conditional probability of a gray-level and the threshold was selected by minimizing the interdependence between two parts of a thresholded image which was measured by the sum of two entropies. Kapur, Sahoo, and Wong (1985) proposed a method in which the optimal threshold was defined as the gray-level which maximizes the sum of the entropies of an object and its background. Pal and Pal (1988) suggested using higher order entropy and conditional entropy. The threshold is selected in such a way that the sum of entropies of two quadrants in a co-occurrence matrix is maximum. Abutaleb (1989) extended the common one-dimensional entropy method to a two-dimensional one where the entropy was the function of a pair of gray-level variables which were the gray-level of each pixel and the average gray-level value of its neighborhood. The threshold is selected in such a way that the entropy-based function is maximized. Brink (1992) also used two-dimensional entropies based on the local average gray-level histogram. The two-dimensional threshold vector that maximizes both background and object class entropies was then selected. Li and Lee (1993) solved the threshold selection problem by minimizing the cross entropy between the image and its segmented version.

Unlike the histogram-based algorithms which use only quantitative information about gray-level distributions, nonhistogram-based or spatial information-based algorithms try to use second order statistics, such as the co-occurrence matrix, to determine the position information about pixels with that distribution. However, all spatial information-based algorithms eventually threshold the histogram. Since they make use of the spatial details, it is expected that they would result in a more meaningful segmentation than the histogram-based methods. Kirby and Rosenfeld (1979) proposed a scatter plot method which was similar to the co-occurrence matrix method. The histogram by near diagonal entries has a deep valley while the histogram by off-diagonal entries in the plot should have a sharp peak. Thus, the threshold can be better determined. Deravi and Pal (1983) proposed a transition matrices method. The threshold is selected based on two measures which are the function of total transitions and indicate the spatial discontinuity of the segmented classes. Pal and Pal (1987) described an algorithm that used a "homogeneity" and "contrast" measure defined by the co-occurrence matrix.

Some algorithms employ edge information in guiding the determination of a threshold. Kartz (1965) proposed an edge filtering method. Only edge pixels are used to create a gray-level histogram that has a single peak at a gray-level between the object and the background which is then used to help in choosing a suitable threshold value. The edge weighting method by Mason et al., (1975) uses the gradient edge operator for weighting the gray-level histogram to generate a new histogram with sharper peaks and deeper valleys. Kohler (1981) proposed a recursive uniform average contrast method. This method selects the threshold which generates the highest average contrast of detected edges over the image using an expected contrast histogram. Wong and Haralick (1984) proposed a recursive technique in which the edge pixels are first classified and two histograms are then obtained from those edge pixels that are relatively light and dark on the basis of their neighborhoods. A threshold is selected corresponding to one of the highest peaks from the two histograms. Kittler et al., (1985) proposed a simple statistical method in which the threshold was equal to the ratio of two sums, the sum of the maximum gradient

magnitude of the image and the sum of the product of the maximum magnitude and gray-level value.

There are other methods that do not directly depend on the histogram and they seem to cover a wider variety of image types. The amplitude method by Boukharouba et al., (1985) is based on the intrinsic properties of the distribution function of an image. In that method, the curvature of the distribution function is approximated by a least squares method using the Chebyshov basis. The threshold is chosen at valley-like points. Bilbro et al., (1990) used a tree annealing minimization technique to optimally find the global minimum corresponding to the threshold of an image whose histogram was multimodal. Zhang and Gerbrands (1991) presented a method which was based on the determination of the transition region. The threshold value is selected as the mean or mode gray-level of the transition region.

2.4.1.1.2 Local thresholding

A fixed threshold that is chosen based on overall information about an image will not be appropriate in cases of uneven background, poor illumination or high noise. In such cases, although locally the objects will still be lighter or darker than the background, global information such an overall histogram might not be useful for differentiating objects from the background. A threshold varying over different image regions so as to fit the spatially changing background and lighting conditions is needed. This is called a local thresholding.

In local thresholding, the original image is divided into smaller regions called windows or tiles. Standard thresholding methods are then applied to determine the threshold for each region or even for each pixel of the entire image. A smoothing technique can be applied to eliminate discontinuities at the boundaries of two different regions and a cleaning procedure can be applied to eliminate the incorrectly thresholded objects.

There are some problems with local thresholding. First, if the window contains no object, then attempting to threshold is dangerous because a threshold is

always found by a global thresholding algorithm unless the region is completely uniform. Second, if more than one object is present in the window, then a single threshold may not suffice. Third, if an object overlaps several windows, then it may be represented as several objects since several different thresholds may be used. The final problem is the consumption of time and memory.

There are a few local thresholding algorithms (Figure 2.1). Chow and Kaneko's method (1972) is used to detect boundaries in very low contrast and low quality radiographic images. The original image is divided into overlapping windows; a gray-level histogram is computed for each window; and thresholds are selected for those windows that have bimodal histograms. Those thresholds are then interpolated to find the threshold for each pixel in the entire image.

Milgram (1979) suggested a method called convergent evidence. The edge map is first found from a smoothed image using a median filter, and then the smoothed image is thresholded at all gray-levels above the mode of the histogram. Finally, thresholded candidates are then accepted or rejected based on the coincidence of an edge map with a region boundary. Only the one defining the greatest coincidence of threshold region border and edge is deemed valid for a particular region.

In the minimum and maximum method, Bernsen (1986) used an overlapping window for finding the threshold for the center pixel in the window. The threshold is defined as the mean of the maximum and minimum values of the pixels in that window. Such a threshold has second-order derivative behavior.

Yanowitz and Bruckstein (1988) proposed the method employing gradient information. The locations of maximum gradient are used as pointers to positions at which the gray-level of the original image is sampled. Interpolation is used to find threshold values for every pixel in the image. Finally, the thresholded image is cleaned by a validation process to remove incorrectly thresholded objects.

2.4.1.2 Clustering

Strictly speaking, the clustering technique is a multidimensional extension of the concept of thresholding. A clustering method is used to group the points in the characteristic feature space into clusters. Typically, two or more characteristic features are used and each class of regions is assumed to form a distinct cluster in the space of these characteristic features. These clusters are then mapped back to the original spatial domain to produce a segmentation of an image. Unlike thresholding, clustering not only includes gray-scale but can use any feature, for instance, a color feature. This is why many color image segmentation algorithms belong to clustering. A number of representative algorithms are listed in Figure 2.1.

Clustering does have problems. First, adjacent clusters frequently overlap in the color space and this causes incorrect pixel classification. Second, the clustering is more difficult when the number of clusters is unknown, as is typical for segmentation applications. Third, since it uses the pixel as a unit and compares each pixel value with every other pixel value, the clustering method usually requires long computation times.

2.4.1.2.1 Multidimensional extensive threshold clustering

The multidimensional extensive threshold clustering technique is just as the name implies in that it is a direct extension of thresholding techniques because a multidimensional histogram is used for clustering the original image pixels. In this kind of technique, it is difficult in automatically choose the modes in the multidimensional feature histogram, especially when the dimension of the feature vector is large. In segmenting cervical smear images, Aggarwal et al., (1977) used a two-dimensional thresholding at a pre-set gray-level to extract the nucleus, or cluster.

To cluster large quantities of LANDSAT images, Narendra and Goldberg (1977) used a non-parametric clustering algorithm. In their method, a 4-dimensional

histogram is computed to reduce the large pixel data to a much smaller number of distinct vectors. The vectors are clustered using the histogram count as a probability density estimate. Goldberg and Shlein (1978) used the idea of the multidimensional histogram of four bands for multispectral images. The peaks in the multidimensional histogram are used as cluster centers and all the other measurement vectors in the histogram are assigned to the nearest cluster center.

Celenk (1990) used only a one-dimensional histogram for CIE Lab color space images. Image clusters are detected by estimating their distributions in some well-defined decision volumes of the constant lightness and constant chromaticity network. Then the estimated color clusters are projected onto a line using the Fisher linear discriminant criterion for one-dimensional thresholding.

The peak-climbing algorithm by Khotanzad and Bouarfa (1990) identifies the significant peaks of a multidimensional histogram of the vectors to be clustered. The feature vectors are calculated from small local areas of the color image.

2.4.1.2.2 Spatial clustering

By Haralick and Shapiro's definition (1985), spatial clustering is a technique which combines the mode seeking technique with a region growing or spatial linkage technique. In general, spatial clustering is computationally more time consuming than extensive threshold clustering since one more step of pixel clustering is needed. Since many of these algorithms also depend on the same mode seeking as is found in thresholding algorithms, their application limitations are almost the same as for thresholding.

Haralick and Dinstein (1975) proposed a gradient thresholding algorithm applicable to multi-spectral image data. The gradient image is thresholded by a threshold of each subset which is the mean of the gradient values of resolution cells in that subset. A sequential labeling procedure is used and the labeled connected regions are clustered in the cleaned image.

The shared near-neighbor clustering method is one of the typical agglomerative clustering methods and is particularly effective in detecting chain-like clusters. Jarvis's shared near-neighbor clustering algorithm (1977) links spatially adjacent elements on the basis of a similarity measure derived from considering the shared near-neighbors in feature space selected from among an extended spatial neighborhood centered upon the adjacent pair of elements. Hattori and Torii (1993) used a similarity matrix in clustering based on fuzzy set theory. However, in contrast with fuzzy clustering, no transitive relation is necessary between the elements of the similarity matrix in the algorithm.

The k-means clustering method by Coleman et al., (1979) iterates on the number of clusters and evaluates the clustering based on the parameter of clustering quality. Brailovsky (1991) introduced a probabilistic-based clustering algorithm by using the K-means algorithm and the mean square error of clustering as the estimate of clustering quality.

Fukada (1980) suggested a shortest distance method for color images. The original image is repeatedly divided into nonoverlapping squared regions until all regions are uniform. The vectors of uniform regions are used as kernel candidate vectors. These kernel candidates are then merged using the merging distance and grouped into several clusters.

The methods based on fuzzy theory play an important role in spatial clustering. Fuzzy c-means algorithms use iterative optimization of an objective function based on a weighted similarity measure between the pixels and each c cluster center. Local extrema of the objective function are indicative of an optimal clustering of the image data. This technique encompasses a large group of algorithms in spatial clustering. Huntsberger et al., (1985) modified the c-means algorithm to segment a full color image. The iterative algorithm consists of an initial clustering of a randomly chosen sample of pixels taken from the input image into 4 clusters. The cluster centers of this sample are used to calculate membership functions for all of the pixels in the image. Trivedi and Bezdek (1986) suggested an approach which utilized the region splitting concept and a pyramid data structure for hierarchical analysis of aerial

images. The fuzzy c-means clustering algorithm is also used to partition the measurement vectors. Cannon et al., (1986) exemplified an approximate fuzzy c-means implementation based on replacing the necessary "exact" values in the fuzzy c-means equation with estimates to reduce the complexity of the fuzzy c-means. Lim and Lee (1990) combined thresholding and fuzzy c-means together. They used thresholding to do the coarse segmentation while the remaining unclassified pixels were assigned to the closest class using fuzzy c-means. To detect clusters that can be described by shell-like subspace, Dave (1990, 1991) proposed the fuzzy C-shells algorithm and fuzzy adaptive C-shells algorithm. Krishnapuram et al., (1993) generalized them into a fuzzy C quad shells algorithm to overcome the difficulty in finding curves.

2.4.1.2.3 Other clustering methods

Pyramid-based methods by Burt, Hong and Rosenfeld (1981) describe a spatial clustering scheme which uses a pyramid structure. Each successively higher layer of the pyramid is an image having half the number of pixels per row and half the number of rows of the image below it. The segmentation is based on the local average gray-level, while the estimated gray-level assigned to each image node is the average computed over the entire segment containing that node. The node is linked to the corresponding "most similar" father node.

Pietikainen and Rosenfeld (1981) extend Burt et al.,'s pyramid-based method to segment an image using a texture feature. The texture feature used is a second-order gray-level statistic "contrast" which is the moment of inertia of the co-occurrence matrix about its main diagonal. The global information is obtained from the upper pyramid levels in order to get good segmentation for the texture image.

Amadasun and King (1988) proposed an approach which combines clustering with the region growing concept. Essentially, the original image is divided into a number of nonoverlapping square neighborhoods, and each neighborhood is tested for uniformity. The mean feature vectors are computed for all uniform neighborhoods.

If the number of mean feature vectors is less than the number of categories, then the uniformity criterion is considered too strict and is relaxed. If the number of mean vectors is greater than the number of categories, the two most similar mean vectors are determined. The normalized Euclidean distance is used as a measure of similarity.

2.4.2 Region Extraction

Region extraction techniques utilize the property of similarity to segment image data by grouping pixels such that they are all contiguous and similar, or homogeneous, from some well-defined mathematical point-of-view. Hence, they offer good noise immunity and do not require as much reliance on semantic knowledge as edge detection techniques. The region-based approach has the advantage of being able to detect textured objects, unlike simple thresholding and most edge detectors.

One strategy of region extraction, splitting and merging, is the dividing of an image into basic regions, either by adjacent identical points or arbitrary small regions, and then merging the adjacent regions into one region if they are similar while dividing a region into smaller regions if it is not uniform. The similarity and uniformity criteria for merging and splitting produce different algorithms. Another strategy, called region growing, involves finding the region seeds first and then using similarity criteria to grow the spatially adjacent pixels to form the regions.

Problems with the region based methods can arise in the selection of initial regions or region seeds and in the selection of the merging or splitting criteria. Also, a partially processed image would not contain a few clear, dominant regions in the image, but would contain many small unmerged regions. The merging and splitting techniques usually produce square-shaped boundaries. Some of the existing algorithms are listed in Figure 2.2.

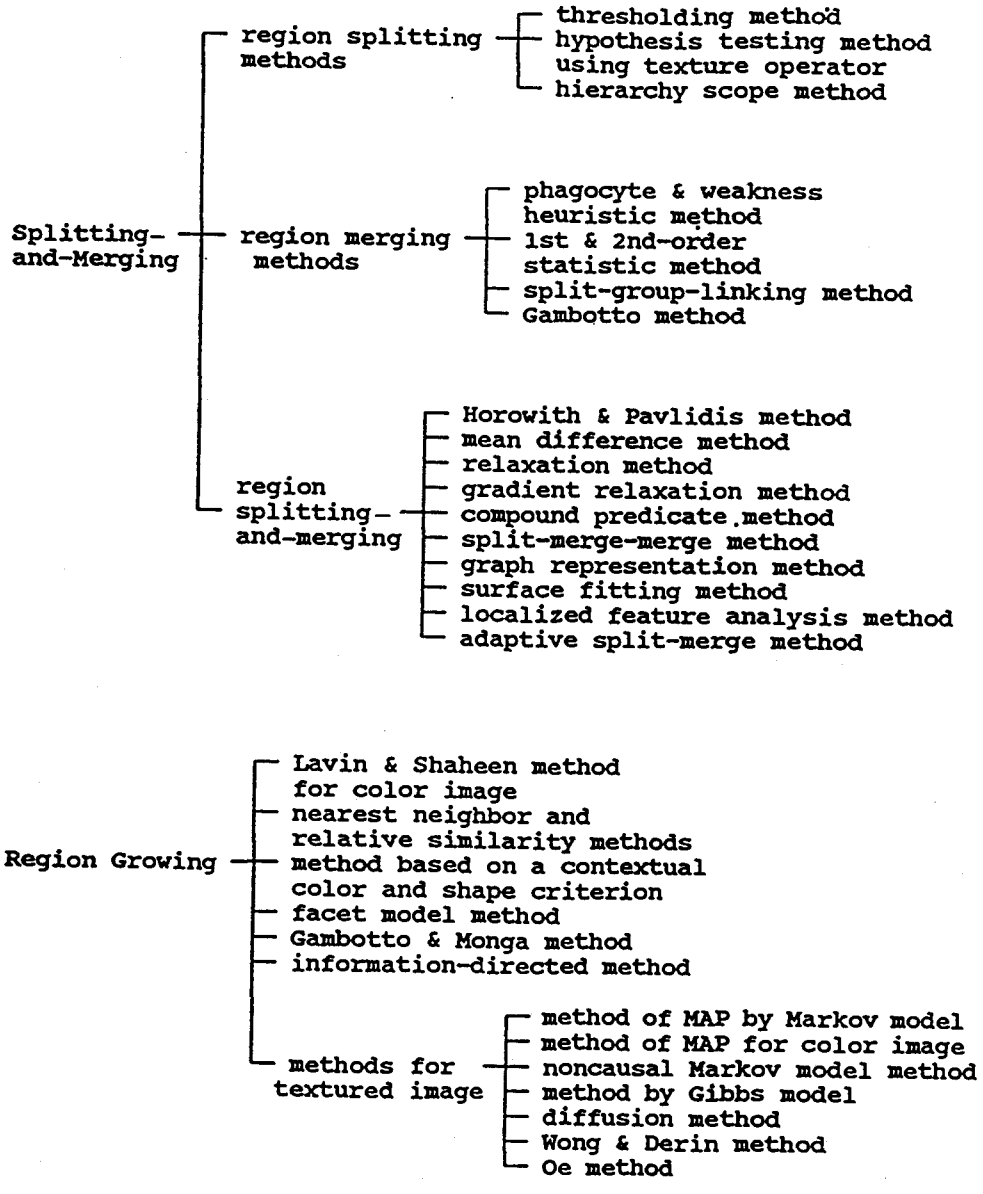

Figure 2.2. Examples of region extraction algorithms.

2.4.2.1 Region splitting and region merging

In the splitting and merging technique, the original image is first divided into small regions that are split or merged according to certain uniformity or homogeneity criteria. For region splitting and region merging, the quadtree data structure is commonly utilized; that is, if a region is not uniform, then it is divided into its four quadrants, and if any two or three quadrants are similar, then they are merged together. The uniformity and similarity criteria can be based on statistics, edge information, or histogram modes of certain image features. Usually, the splitting and merging methods are distinguished by their uniformity and similarity criteria.

Split and merge has been shown to be the most computationally efficient algorithm for region-based segmentation. It is comparatively easy by involving a variety of different methodologies, such as thresholding and edge detection. It also takes advantage of the quadtree data structure which facilitates a well-understood and concisely coded implementation. Finally, the split and merge algorithm has modularized the content of a predicate, or measure of region homogeneity. This modularization allows the implementor to experiment with a variety of predicates to handle complex data, while providing data independence in a manner analogous to machine independent programs.The main drawback of splitting and merging is that it does not provide a unique solution; a different initial partition produces different results. This is because the procedure is sequential so that the result depends on the processing order. Another problem is that almost all algorithms use local information heavily. There is no simple way to incorporate global information into the model unless the class of images dealt with is restricted. The third problem is that the resulting boundary found by the splitting and merging technique is usually more or less stepwise in shape caused by its inherent methodology that the boundaries have only two directions, horizontal and vertical.

2.4.2.1.1 Region splitting

Region splitting is a top-down process. The whole image is treated as a starting point, and then it is successively divided into smaller and smaller regions until certain uniformity criteria are satisfied.

A thresholding method by Ohlander et al., (1978) employs the best peak in some feature histograms to divide the region being considered. Tominaga (1986) described a method for segmenting a color image by a recursive thresholding method using the three histograms corresponding to H, L, and S of the Munsell color space. The whole image is partitioned into two sets of subimages using two threshold values derived from the lower bound and the upper bound of the most significant peak in the three histograms. Then the thresholding process is repeated for the extracted subregions until either no significant peak exists in the histograms or the sizes of subregions become very small.

Conners et al., (1984) proposed a method which is mathematically developed as a series of hypothesis testing using texture operators. The primitive operators used to drive the segmentation are texture measures derived from co-occurrence matrices. At every level the procedure differentiates uniform regions from boundary and unspecified regions. It then assigns a class label to the uniform regions. The boundary and unspecified regions are split to form higher level regions.

Lee (1986) described an algorithm in which a quadtree structure was used to store the split results at different levels. The histogram is used as the criterion for splitting. The segmentation will proceed to small scope views only if the result at that level is not satisfactory according to a specific criterion.

2.4.2.1.2 Region merging

Region merging is a bottom-up process. An image is initially divided into a large number of small regions which are then merged to form larger regions. The key

point for merging is the initial size of the small regions because it determines the merge order and therefore decides the result of the segmentation.

Phagocyte and the weakness heuristic method by Brice and Fennema (1970) break the image into tiny regions of uniform gray-scale. Then a pair of heuristic phagocyte and weakness are used to merge these regions. Regions having strong uniformity are called phagocytes which become the merge seed, while the regions having weak uniformity are called weaknesses which will be merged with a phagocyte.

The first-order and second-order statistics method by Gupta and Wintz (1974) uses a minimum distance classifier which interprets each initial region as belonging to one of a small predetermined number of different classes. Neighboring regions are merged based on their statistical similarity. The similarity measure is based on the hypothesis testing of first- and second-order statistical properties of the elementary regions.

Browning and Tanimoto (1982) modified the spliting-and-merging algorithm of Horowitz and Pavlidis (1974) to make the split-and-group-with-linking method. They omit the merging step and add grouping followed by a linking procedure so that the memory required is reduced. Moreover, since vertical and horizontal linkings are introduced, the edge information is incorporated into the segmentation decision.

Gambotto (1986) proposed an algorithm which simultaneously used the region means and gradient estimates over the boundaries of the regions. The region means are for merging, that is, two regions are merged together if their means are similar; the gradient is for control, that is, the merging process terminates when the true boundary of a region is reached.

2.4.2.1.3 Region splitting-and-merging

The combination of region splitting and region merging increases the quality of segmentation and increases processing speed while maintaining small memory requirements. A split-and-merge procedure begins with an arbitrary partition not

satisfying a criterion condition and produces a partition satisfying the condition by both merging and splitting operations.

The quadtree-structured method by Horowitz and Pavlidis (1974) is the first attempt to split-and-merge by using a quadtree data structure. In their method, a predetermined error tolerance of brightness of each pixel over a region is used as the criterion for merging and splitting. Many splitting-and-merging algorithms use this data structure. The main problem with this sort of split-merge is that the edge segments formed have only two orientations (horizontal and vertical), and their position is restricted by the borders of the quadtree nodes.

The mean difference method by Chen and Pavlidis (1980) regards image segmentation as an estimation problem. Its uniformity measure is defined as the estimated mean difference in brightness between two sample regions as compared to a predetermined threshold.

The relaxation method by Bahram and Parvin (1984) uses the region splitting kernel based on two class relaxation techniques to control the convergence of the segmentation process. Bhanu and Parvin (1987) improved this method by basing the criterion for splitting on a generalization of a two-class gradient relaxation method. The compound predicate method by Doherty and Bjorklund (1985) segments regions of high texture into single entities as well as those with similar gray-scale statistics. The compound predicate combines the use of a first-order average gray-level for areas with low texture with the use of a second-order inverse difference moment texture measure for areas with high texture. To improve the performance of the compound predicate method, Doherty et al., (1986) suggested a split-merge-merge procedure. An additional texture-based merge step produces improved segmentation results for regions that have been over-segmented by spliting and merging when using the mean predicate.

To improve the property of splitting-and-merging algorithms that produce results dependent on the order of processing, Cheevasuvit et al., (1986) use a graph representation of the splitting-and-merging procedure as described by Horowitz and Pavlidis. This graph provides a flexible and powerful description of the temporal

behavior of the sequence in terms of continuity, merging, splitting, birth, and decay of homogeneous and robust areas.

The intensity surface fitting method by Laprade (1988) approximates the image intensity surface by planar facets. A region is facet fitted by least-squares, then a combination of an F-test about residuals of surface fitting and a mean predicate of that region is used to test the uniformity of the region.

The method using localized feature analysis and statistical tests by Chen et al., (1991) combines the strength of characteristic feature analysis and a hypothesis model to produce an initial segmentation. All the parameters are computed automatically on the basis of the characteristic features. Those parameters provide the hypothesis model with appropriate constraints to test the region homogeneity.

Wu (1993) proposed an adaptive split-and-merge algorithm which tried to overcome the main problem of quadtree-structured methods. In his method, a recursive optimal four-way split and the RAG-guided (region adjacency graph) minimum cost merge are used to improve the two-way edges in the conventional split-and-merge. Its novelty is in splitting a nonuniform region along the strongest edge in it and merging two over-split regions if their border is not a valid edge.

2.4.2.2 Region growing

Region growing is the process of joining neighboring points, or collections of points, into larger regions subject to some criteria. The region growing technique treats an image on a pixel by pixel basis. The algorithm begins with a seed location and attempts to join neighboring pixels to this growing seed until no neighbors can be joined to it; it directly compares adjacent pixels and carries out the merging process. The main problem with the region growing approach is how to choose the region seed. Another potential problem is an inherent dependence on the order in which pixels and regions are examined. The computation burden of region growing can be significant. The different criteria used for controlling the region growing or merging

produce different algorithms. These criteria are commonly the uniformity measures for the region or for the pixel group. They are naturally ad hoc.

Levine and Shaheen (1981) proposed a scheme in which the color features for the RGB color space were the means and standard deviations of the three color components for every region. Regions are grown provided that the difference between each color feature for the pixel and the average feature value over the region is less than a specified threshold.

Asano and Yokoya (1981) suggested two algorithms called nearest neighbor and relative similar methods. In the first algorithm, each pixel is compared with its four neighboring pixels and merged with its closest neighbor. In the second algorithm, pixels are considered to belong to the same region if their difference is less than the minimum value of the average of gray-level differences between the pixel being considered and its surrounding 8-pixel neighborhood.

To segment bone marrow cell images, Chassery and Garbay (1984) used a method based on contextual color and shape. It combines local information, color, and global information to control the aggregation step. The luminance, saturation and hue are used as color features. A color reference is computed as the mean color of initial points. Points displaying colors similar to the reference color are aggregated to the initial points and the reference color is then updated. The convexity shape test is performed at each step, based on a convex hull and a mask areas comparison, to restrict the region growing.

In the facet model method by Pong et al., (1984), the constants of a gray tone surface, which is a sloped plane of a region, are estimated using least-squares. The facet modal tells that regions are connected sets of resolution cells whose gray tones belong to same polynomial surface.

Gambotto and Monga's method (1985) processes all the regions in an image in a parallel and hierarchical fashion. It assumes that each pixel is a region. Merging is performed at each iteration and is based on the analysis of the mean gray-level within neighboring pixels. The merging criterion is iteratively modified in order to allow new regions to merge at subsequent iterations. The second criterion, which

involves an estimate of the mean gradient over the common boundary of adjacent regions, is used to control the region growth.

The information-directed method by Raafat and Wong (1988) uses a method directed by a resolution dependent texture information measure. This measure, indicating how typical an image block is with respect to other blocks in the image, is used for region initiation. Then it uses the information measure together with a texture distance measure to detect the growth of various homogeneous seed regions.

There is a group of methods specially developed for textured images. They use random fields to model the textured regions, estimate certain feature probability of the regions, and then grow the pixels to form regions in such a way that estimation is maximized. The maximum a posteriori method (MAP) by Therrien (1983) utilizes a Markov random field model to represent the occurrence of textured regions. Segmentation of the image is then treated as a region estimation problem. Maximum likelihood and maximum a posteriori estimation are applied to estimate regions of similar terrain. Therrien (1985) extended the MAP method to color images by considering three color components R, G, and B together. Cohen and Cooper (1984) proposed two noncausal Markovian random field model methods. In each, the data are modeled as one of C noncausal 2-D Markovian stochastic processes. Image segmentation is realized as the true maximum likelihood estimation. The first algorithm is hierarchical and uses a pyramid-like structure while the second algorithm is a relaxation-type. The Gibbs random fields model method by Derin and Cole (1986) uses the Gibbs random fields. The algorithm seeks to obtain the MAP estimate of the region process using the textured image data. The maximization is carried out recursively by making use of a dynamic programming formulation.

Reed et al., (1990) introduced the diffusion idea to texture image segmentation. This approach allows the region growing process to take place not only between adjacent pixels in a region but between randomly selected pixels in the entire image. The image is modeled by a pseudo-Wigner distribution. Wong and Derin (1992) used Markov random fields as components of the image model. Their algorithm determines the number of regions of the image through a model fitting criterion tagged

on to the segmentation algorithm. Oe (1993) proposed a new discriminant function for the segmentation that measures the statistical difference between two textures by using the two-dimensional autoregressive model and Kullback information (Kullback, 1959) which can measure the distance between two probability distributions.

Reed (1993) wrote a paper reviewing texture segmentation in which more algorithms are discussed in greater detail.

2.4.3 Edge Detection

An edge is the separation of two regions in an image that have different features, such as gray-level, color, texture patterns, etc. Because edge detection is a technique based on the detection of discontinuity, it searches for parts of the image where a transition occurs from one uniform region to another. Many edge detectors for gray-scale images can be modified for color images. This is a significant advantage of edge detection over thresholding. Since edges are local features, they are determined using local information. To utilize edge detection techniques, the image must have a certain degree of abruptness between objects and background.

In the edge detection techniques, the edge must be collected into line segments, and these line segments must then be grouped into objects. Usually these systems require a well-defined model to allow the grouping of the segments into objects. Problems can arise when edges are not well defined or too many edges occur. The greatest disadvantage of edge or line oriented segmentation systems is the need for grouping the line segments into coherent regions. Other problems with edge detection techniques are that: (1) sometimes the detected edges are not the transitions from object to background but the transitions inside the background or objects, (2) the detected edges often have gaps in them at places where the transitions between regions are not abrupt enough, and (3) edge detection is computationally expensive.

Edge detection techniques can be classified into two classes: parallel and sequential. A number of algorithms are listed in Figure 2.3.

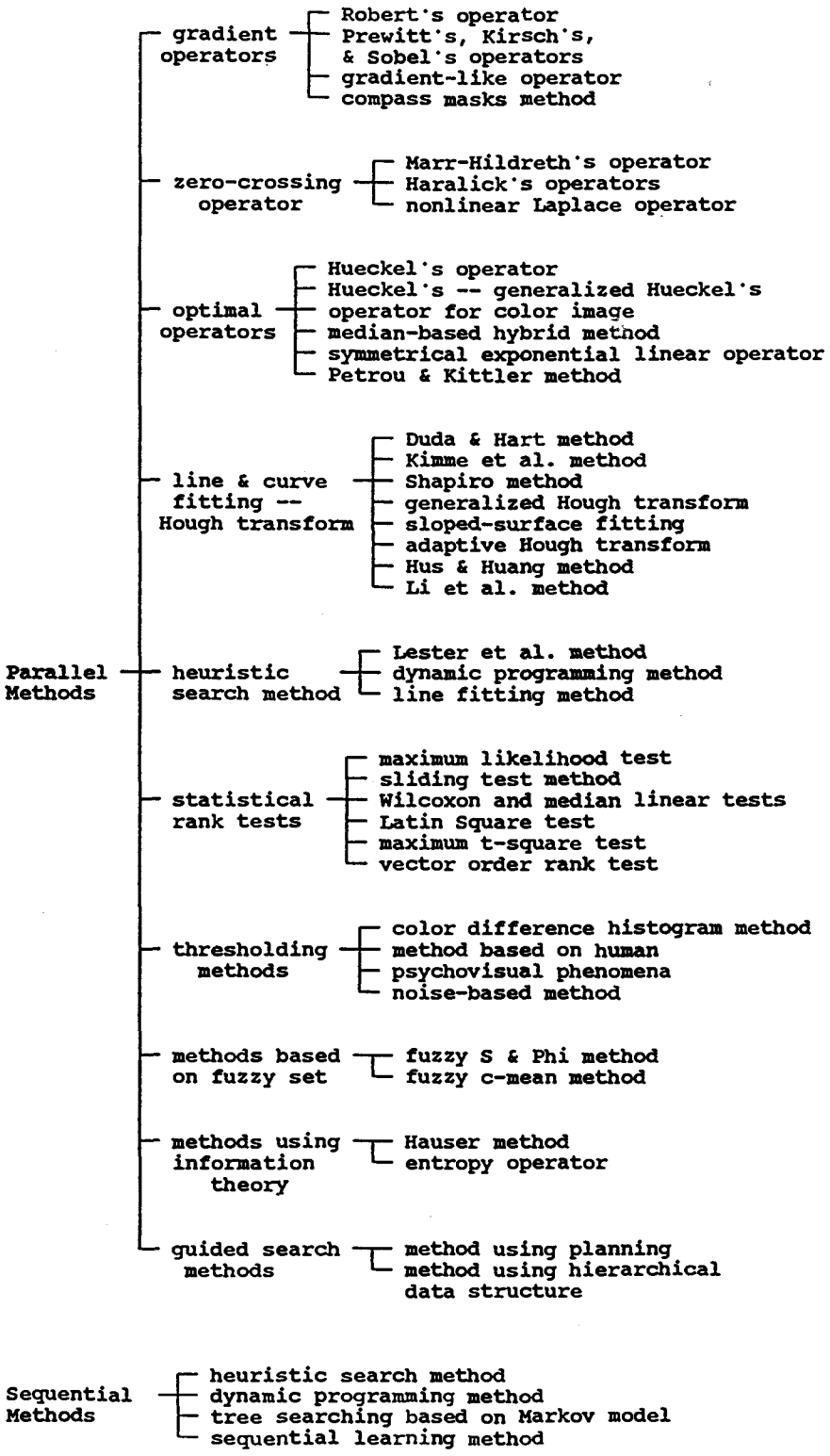

Figure 2.3. Examples of edge detection algorithms.

2.4.3.1 Parallel edge detection

In parallel edge detection, the decision of whether or not a set of points is on an edge is not dependent on previous decisions of other sets of points laying on an edge. So the edge detection operator may be applied simultaneously everywhere in the image. Parallel edge detection raises the issues of how to choose the window size or the size of the region in which the detectors operate and the local optimality for determination of edge pixels.

Probably the biggest group of the parallel edge detectors is the differential edge operators. The mathematical principle of differential operators is that the first-order derivative is big or the second-order derivative equals zero at the place where the discontinuity occurs. The gradient edge detector is good at detecting sharp edges. It is the most popular edge detection technique because it has a well-understood mathematical model, is easy to implement, and computationally is comparatively inexpensive.

Unfortunately, the gradient operators have a number of problems. One of them is the response not only to edges but also to isolated points. The second is the inherent bias in the estimate of edge direction because all gradient operators are the approximations of derivatives. The third is sensitivity to the presence of noise in the image data. That problem can be alleviated by increasing the processing neighborhood size, but this is usually at the expense of an increase in estimate bias and also in errors in the processing of small or thin objects. Usually, the bigger the neighborhood size, the bigger the inaccuracy of the edge position.

Quite a few such edge detectors exist based on the digital approximations on variations of first derivative (gradient) and second derivative (Laplacian) operations. Examples of gradient operators are Robert's cross operator with a 2 by 2 window size (1965) versus Prewitt's (1970), Kirsch's (1971), and Sobel's (1973) operators with a 3 by 3 window size. The main difference between these operators is the weights assigned to each element of the 3 by 3 or 2 by 2 template. Their performances

change depending upon the image type but quite a few researchers have found that the Sobel operator performs best.

It is worth mentioning the compass edge detector proposed by Robinson (1977a, 1977b). This is a detection system which makes use of any 3 by 3 compass gradient operator. The edge angles are quantified for eight equally spaced directions. The edge value is the maximum gradient magnitude and the phase value of the gradient is used to test the connectivity of edges. For a color image, the edges are obtained by comparing the 24 gradient values for each of the three color component's eight directions. The gradient that gives the maximum value at a point thus determines the direction of the edge, whether due to a discontinuity in the first, second or third color component. Any kind of gradient edge operators can be employed in this compass mask system. Robinson tried to consider eight directions instead of the two directions in gradient operators to increase the accuracy.

Zero-crossing-based edge detectors are also second derivative operators and were first suggested by Marr and Hildreth (1980). They determined that the appropriate edge operator is a Laplacian-Gaussian because it minimizes noise and produces continuous line-like edges. In this approach, wherever an intensity change occurs, there will be a corresponding peak in the first directional derivative, or equivalently, a zero-crossing in the second directional derivative of intensity. Haralick (1984) developed a directional second derivative step edge operator based on the cubic facet model. The measure of gradient strength is the maximum value of the integral of the first directional derivative taken over a rectangular or square neighborhood, the maximum being taken over all possible directions for the directional derivative. To improve the performance of Marr-Hildreth's operator, Vliet et al., (1989) suggested a nonlinear Laplace filter which takes the second-order-derivative-like operation on the most relevant direction perpendicular to the local edge direction. After studying general zero-crossing-based edge detectors, Lee et al., (1993) mathematically derived a new zero-crossing edge detector which is optimal in the sense of signal-to-noise ratio. The operator they found is bell-shaped but different from a Laplacian-Gaussian.

The third group of edge detectors is called optimal edge detection because the technique is based on optimizing a certain cost function or criterion which is from information theory, a signal processing technique, or from empirical results. The optimal operator is usually computationally expensive and the optimization is theoretically realistic but hard to implement. Heuckel's operator (1971, 1973) used a circular neighborhood instead of a square one and the operator performed best when given an empirical edge element characterized by a step function. Nevatia (1977) generalized Heuckel's operator for color images, in which the optimal steps were computed separately for the three components.

Neuvo et al., (1986, 1987) proposed a median-based hybrid detector based on a combination of linear filters followed by a median operator. The linear filters are designed to have an edge detection characteristic and special response properties such as small noise gain or short edge response, while the median filters have good noise cleaning and edge preserving properties. To improve noise immunity and precision of Laplacian-like edge operators, Shen and Castan (1986) derived a symmetrical exponential linear operator. They found that the optimal filter is a symmetrical exponential filter which has a similar noise eliminating effect compared to Laplacian-Gaussian filter but localizes edge points with better precision. Petrou and Kittler (1991) developed an optimal edge detector for ramp edges using an optimal filter approach established by Canny (1986). They model the ramp edges as an exponential function and find the optimal function which maximizes the good locality measure which is inversely proportional to the standard deviation of the distribution of points where the edge is supposed to be.

Another large edge detector group includes line, curve and surface fitting. They are particularly designed to detect certain types of edges such as straight lines, circles, ellipse, or parabolas. Therefore, they may behave poorly in detecting other sorts of edges. The basis for most of these methods is the Hough transformation. The Hough transformation uses an angle-radius parameter as developed by Duda and Hart (1972) as a way of detecting collinear points. Kimme et al., (1975) extended and improved the circle-finding concept by extending the Hough straight line finder.

Shapiro (1975) extended the Hough transformation to detect arbitrary curves in the presence of noise. Ballard (1981) generalized the Hough transformation method for detecting lines, circles and parabolas by use of directional information. Haralick's sloped-surface fitting method (1980) uses a surface fitting model by which edge detection can be done. His framework is based on the sloped-facet model which assumes that regions of image segments are maximal areas which are sloped planes. Edges are declared to exist at locations of local maxima in the F-statistic edge strength picture. Illingworth and Kittler's adaptive Hough transform (1987) is a more flexible variation of the Hough transformation. Hsu and Huang's partitioned Hough transformation (1990) uses parameter space decomposition to detect three-dimensional shape. Li et al., (1993) published an algorithm called the linear generalized Hough transform. For a more detailed discussion and review, see the survey about Hough transforms by Yuen et al., (1990) and Leavers (1993).

Heuristic search methods can be guided by other techniques embedded in the cost functions. There are a few algorithms in this group. The gradient combined with a thresholding method as proposed by Lester et al., (1978) incorporates both threshold and gradient information in the cost function to guide the search. The basic cost is lowest at points where the gradient is comparatively high and the average gray-level in the immediate area is close to the threshold value which is the average gray-level of nine neighborhood pixels.

Montanari (1971, 1972) suggested using dynamic programming techniques to perform low-curvature edge detection. A figure of merit representing the heuristic information is used to determine the relative value of different paths but is not used to guide the search as in the heuristic search case. A line fitting method by Furst and Caines (1984) locally fits the best line. The regions for the application of the dynamic programming technique are selected by first decomposing the entire image into equal square regions and then breaking into four equal subsquares any region where the error count exceeds a fixed tolerance.

Statistical testing is a common tool to detect the discontinuities in an image. The main problem with these methods is that an assumption about feature distribution

of an image is made (commonly assumed to follow a normal distribution). Therefore, the method will fail if the real distribution of the feature in an image is different. Yakimovsky (1976) assumed that regions are normally distributed and used a maximum likelihood test to determine edges. Souza (1983) discussed the problem of detecting edges in one-dimensional profiles in his sliding statistical tests method. He suggested that edges can be detected by locating the positions where a function computed over the window undergoes a particular type of turning point. The function is dependent upon the nature of the problem and five different functions are based on statistical tests (t-test, tau test, F-test , X-test, and the log likelihood ratio test). Bovik et al., (1986) introduced two edge detectors based on nonparametric test statistics such as the Wilcox test and median linear rank test. To find the edge of an image imbedded in correlated noise rather than Gaussian noise, Stern and Kurz developed (1988) an edge detector using a 5 by 5 latin-square mask. Huang and Tseng (1988) proposed a maximum t-square test method which was the maximum of the t-square distribution over all meaningful partitions of a n by n subimage. This statistic is a function of mean and variance. Trahanias and Venetsanopoulos (1993) suggested using vector order statistics to detect edges in a color image. In this approach, a color image is treated as a vector field and the edge information carried directly by the vectors is exploited. The edge is confirmed if its vector range is in highest rank from the vector order.

Thresholding methods have their limitations in edge detection since they heavily depend on global information. Generally speaking, they perform about the same as the previously discussed thresholding methods. Pietikainen and Harwood (1986) described a thresholding method for detecting color edges. The cumulative histogram of absolute color differences is computed and connected to percentiles of the distribution of cumulative frequencies. Then the color edge-value at a pixel is the maximum percentile of the cumulative frequencies of the color differences between the center pixel and its eight neighbors. Kundu and Pal (1986) proposed an algorithm which is based on the human psychovisual phenomena. The threshold value adapts with the background intensity according to the criterion governed by a characteristic

of one of the De Vries-Rose, Weber's saturated regions. Haddon (1988) developed a method in which the threshold was dependent only on the noise of the image and independent of the edge content of the image. The threshold is set such that the volume integral outside the threshold is equal to the proportion of nonedge pixels tolerated as edges.

To detect the different regional boundaries of X-ray images, methods based on fuzzy set theory proposed by Pal and King (1983) use the Fuzzy S and Phi function. The edge detection algorithm uses a minimum or maximum operator which defines the edge pixel intensity as the absolute value of the difference between that pixel and the minima or maxima of a set of n neighboring pixels within a circle centered at the pixel concerned. Huntsberger and Descalzi (1985) developed a color edge operator by using the fuzzy c-mean. In their algorithm, the fuzzy membership value for each pixel comes from Bezdek's fuzzy c-mean algorithm (1973). The color edge is defined in the representation as the zero crossing of differences between membership values at each pixel.

A method using information theory developed by Hauser (1984) approximately determines the relative information content of two disjoint parts of a stochastically modeled image which is a stochastic matrix. Shiozaki (1986) presented an entropy operator using the entropy of brightness in a local region of an image. The entropy of brightness (or intensity) is a probabilistic information measure of change rate in brightness or intensity in a region. Since the entropy is small when the change of brightness is severe and large when the change of brightness is small, the edges may be extracted by the detection of the regions where the entropy is small. A color edge can be detected in a similar way.

2.4.3.2 Sequential edge detection

In sequential edge detection, the result at a point is contingent upon the results of the operator at previously examined points. For sequential detectors, global optimality is considered but this may be a disadvantage to have some global

information about the image to be processed, such as the number of objects within the image, or the maximum allowable curvature. Another disadvantage is that if a mistake is made in any particular step, it will lead to further mistakes in the following steps. The performance of a sequential edge detector depends on the choice of an appropriate starting point. Sequential edge detection techniques are applicable for images containing objects with comparatively simple shapes, a more or less uniform background, and a priori information about edges to be detected.

The number of sequential edge detection algorithms is much less than for parallel algorithms. The guided search method by Kelly (1971) uses the planning for guidance. There are three steps in this procedure: first, extracting a new small picture from the original image; second, locating the desired edges in the new small image; and third, finding edges in the original image using the edges found in the small image as a plan. Tanimoto and Pavlidis's method (1975) using a hierarchical data structure generalizes the idea suggested by Kelly. A pyramid as a tree is produced in their algorithm. The low spatial frequencies preserved in coarse pictures are useful in finding regions of interest in fine pictures at low cost. Edge detection is accomplished by using a recursive refining algorithm from near the top (coarse picture) of the pyramid to the bottom (fine picture) of the pyramid.

Heuristic search techniques use state space search methods with heuristic information used to limit the space to be searched. Martelli (1976) formulated the edge detection problem as a heuristic search for the shortest path on a graph. Properties of edges are embedded in an objective function and the edge which minimizes this function is sought.

Elliott and Srinivasan (1981) proposed a dynamic programming boundary detection algorithm based on a Markovian boundary generation model which is a sequence of states in a Markov chain. The algorithm uses dynamic programming to estimate the short boundary segments, and then these short segments are pieced together to form complete boundary estimates by use of a graph search procedure.

Eichel and Delp (1985) proposed a special type of the Markov random field model, the D-SIM field, for the correlated image data in sequential tree searching

algorithms. Their technique explores and ranks potential edges according to a log-likelihood statistic. The sequence of observations along a path in a correlated random field is modeled as an autoregressive moving average process. At each iteration of the algorithm, one of the paths explored so far is extended by one node.

Ghelli (1988) proposed a sequential learning method which was particularly efficient in the cases where the background level was not strictly uniform. This method uses the information supplied by the pixels while they have been classified as the boundary to obtain a Bayesian updating of the information on the two classes of object and background. This updating allows the decision rule to "learn" slow variations of both the background level and the gradient near the boundary in the figure region. This method works sequentially, referring to the likelihood of all the neighbors of the previous boundary pixel, by minimizing the "a posteriori" probability of error.

2.5 SEGMENTATION ALGORITHMS FOR IMAGE OF WOOD SURFACES

In recent years, a number of researchers have investigated the identification of features of wood products using optical scanning technology. In 1983 and 1984, Conners et al., produced a prototype optical scanning system to locate and identify surface defects on boards. Their algorithm has two steps. In the first step, they differentiate clear regions from potential defect regions using a standard Chi-squared test. That is, if the mean, variance and skewness of the gray-levels of a region are less than some predetermined thresholds, the region is considered to be clear wood. Otherwise the region is considered to contain a potential defect. They reported their algorithm to reach 99.67% accuracy in this step. In the second step, they differentiate the types of defects by texture analysis using spatial gray-level dependence. They reported accuracies varied from 57.35% to 88.33% for different types of defects in the second stage.

In 1986, Poelzleitner (Austria) used the Hough transformation to identify and classify defects, such as knots, resin pockets, bark, cracks, and regions of skip. He

stated that the defects were successfully separated from the clear wood in about 100 test images, but no real application results nor exact accuracies were reported..

In 1987, Forrer et al., developed three algorithms for detecting defects on Douglas-fir veneers. A statistical based algorithm, SISM, uses statistical moments derived from the intensity and color channels and a test based on probability estimation. This algorithm produced the best defect accuracy of 94.1%, while maintaining a clear wood accuracy of 86.3% but did not perform well when detecting pitch pockets and pitch streaks. A color clustering algorithm, CISM, is similar to Fukada's algorithm but introduces spread and compactness of the clustering to mark the outlier tile. CISM produced lower defect and clear wood accuracies in all cases. A morphological image processing algorithm, MISM, uses the gray-level generalized dilation and erosion described by Werman and Peleg (1985). The defect accuracy for MISM was comparable to the SISM algorithm, but it produced a lower clear wood accuracy. The authors showed that color information is very important in increasing defect accuracy.

In 1989, Butler et al.,improved the SISM algorithm (NSISM) to reduce the overall error rate in clear wood identification by 17.3% compared to the original SISM results. The main difference between NSISM and SISM was that a secondary threshold was added to the conventional single threshold (primary threshold). Since the secondary threshold was larger than the primary threshold, neighbors of marked tiles with respect to the primary threshold were more readily marked. Also, the secondary threshold allowed the primary threshold to be smaller so that fewer clear wood tiles were marked as defects.

In 1989, Conners et al., described a vision system aimed at developing an automatic cut-up system for use in the rough mills of the hardwood furniture and fixture industry. In that system, the image size was 512 by 480 pixels with a spacial resolution of 64 pixels per inch. While a full color image was input, only the red and blue channels were used. The two-dimensional histogram (R and B) is formed for the initial segmentation of the board. This 2-D histogram is thresholded to remove points that have a relatively low frequency of occurrence on the board. The threshold is

used to find regions of the red/blue plane associated with peaks in the 2-D histogram. Then a region growing algorithm is applied to regions obtained by thresholding in an effort of fill in any holes that might exist. Finally, the regions are expanded and labeled. The region whose elements produce the largest sum of 2-D histogram elements is assigned the clear wood label. The area of the red/blue plane that does not belong to one of the expanded regions is assigned the label "unknown". These labeled regions form a look-up table used to initially segment the board. The isolated points on the board after initial segmentation are eliminated using a thinning algorithm. A simple line-by-line algorithm was used to obtain information about how many defects are in a region and the boundaries of the regions that contain a defect.

2.6 QUANTITATIVE EVALUATION OF PERFORMANCE IN SEGMENTATION

The performance evaluation technique is necessary in order to establish measures of effectiveness, reliability, and accuracy for comparisons of the different algorithms in terms of application goals. However, the general performance evaluation is difficult in the sense of two points. First, no measures can represent all of the different cases in the resulting images. Second, many of the segmentation algorithms have different purposes. Any evaluation method, therefore, must be relatively specific. These are the reasons why visual evaluation is still a very common method for determining image segmentation performance.

In the development of image processing techniques, a few evaluation methods have been proposed. Most of them deal with the performances of different edge operators and are used as quantitative comparisons of the effectiveness of various types of preprocessing operations as facilitators of edge detection. Fram and Deutsch (1974, 1975) proposed two measures, one for estimating what fraction of the detected edge pixels are actual edge points and another for estimating what fraction of the vertical extent of the edge is covered by detected edge pixels. Pratt's figure of merit (1979) is based on the displacement of each detected edge pixel from its ideal position

with a normalization factor to penalize for too few or too many edge points being detected. Kitchen and Rosenfeld's evaluation method (1981) uses local edge coherence which combines two qualities of well-formed edges, good continuity and thinness. Unfortunately these three methods are only useful for special images which are syntactic vertical, horizontal, and diagonal lines, or in the most extreme case, circular.

Levine and Nazif's general purpose performance scheme (1985) is designed for any kind of image segmentation algorithm including both region extraction and edge detection. In their scheme, four measures are introduced: uniformity within each region and contrast between adjacent regions serving as parameters for region analysis, and contrast across lines and connectivity between them for line analysis. However, these measures are not as simple or good as Peli and Mahah (1982).

Peli and Mahah (1982) and McLean and Jernigan (1987) proposed seven quantitative performance measures: correct detection rate (the ratio of the number of correctly detected edge pixels to the total true edge pixels), ambiguity rate (penalizing operators which assign multiple pixels to a desired single edge pixel), percentage of missed edges (the number of missed edge pixels to the number of the total true edge pixels), percentage of wrong edges (the number of incorrectly detected edge pixels to the number of total true edge pixels), mean width of detected edge (the ratio of the number of total detected edge pixels to the number of the total true edge pixels), average squared deviation (or distance) of a detected edge pixel from the true edge, and mean absolute value of the deviation (distance).

Haralick and Lee (1990) formulated the edge evaluation problem as a Bayesian problem and generalized the edge evaluation of Kitchen and Rosenfeld. Their general edge evaluator combines local edge coherence, edge correctness, and edge thinness together so that it is based on the continuity, thinness, and positional accuracy of edges.

Sahoo et al., (1988) proposed two measures for evaluation of the thresholding algorithms, a uniformity measure adopted from Levine and Nazif and a shape measure

which is the function of the generalized gradient value of a pixel and the average gray value of the neighborhood of the pixel.

For the purpose of evaluating the performances of algorithms used to detect defects on veneer surfaces, Forrer et al., (1989) used two measures, the type I error (the percentage of "missed pixels") and type II error (the percentage of "wrong pixels"); more precisely, the ratio of real defect pixels which are not detected to the total number of real defect pixels and the ratio of the detected defect pixels which are really clear wood.

Besides the measures and criteria in direct connection with quantitative performance evaluation, shape discrimination in pattern recognition possesses some potential for performance evaluation. Bribiesca and Guzman (1980) described how to measure differences in shapes using a shape number. The shape number is a description that changes with skewing, anisotropic dilation, and mirror images. The order of a curve shape number is the length of the perimeter of a discrete shape closely corresponding to the curve. To find how close in shape two curves are, the degree of similarity between them is introduced. You and Jain (1984) describe the performance evaluation of shape matching using chord length distribution. In that method, chord lengths between all pairs of boundary points are computed and normalized by the length of the longest chord. The set of all chord lengths is treated as a sample distribution set. Then a Kolmogorov-Smirnov statistic is used as the dissimilarity measure between the two shapes. Bryant and Bryant (1989) suggested five measures for discriminating similar shapes in binary images. They are the ratio of the boundary perimeter to the area enclosed, variance of the derivative of the boundary, the variance of the boundary, symmetry measure, and normalized first- and second-order moments. These measures are invariant under translations, rotations, magnifications, and reflections. The measures are found to be error-free in classification in low noise binary images.

CHAPTER 3 ALGORITHMS USED IN THIS STUDY

It is essential that the algorithms used in the project are suitable for optical images of wood surfaces, particularly for images of Douglas-fir [Pseudotsuga menziesii] veneer. Unfortunately, research and applications of machine vision in the area of wood products have been limited; hence, no systematic survey, analysis, or comparison of image segmentation techniques or algorithms has been conducted for images of wood. As mentioned in Chapter 2, the performance of a segmentation algorithm varies from image type to image type; even the performance of different algorithms in the same segmentation category (thresholding and clustering, edge detection, and region extraction) can be extremely different for the same types of images. In other words, the performance of a segmentation technique is algorithm and image dependent.

The selection and development of algorithms in this project were based on the limited previous research. Two points were considered in selecting segmentation algorithms. First, the algorithm itself had to be based on a mathematical or statistical model that was not too ad hoc or specific for certain image types, and if possible, the algorithm should have a possibility of being extended to color images. Second, the characteristics of the subject image (intensity or color, contrast, background uniformity, object size, and texture) had to be more or less similar to veneer images. Therefore, in order to obtain a more general knowledge for the successive research, several algorithms in each segmentation category had to be considered.

In this chapter, the justification, development, adaptation, or improvement of each algorithm used in the project will be discussed in detail. The algorithms chosen were:

(a) Characteristic feature thresholding:

1. the Otsu method
2. the Kapur et al., entropy method
3. the moment-preserving method
4. the transition-matrix method

(b) Edge detection

1. compass gradient mask edge detector
2. nonlinear Laplace edge detector
3. Shiozaki's entropy edge detector

(c) Region extraction

1. a method combining splitting-and-merging with region growing
2. a method combining clustering with region growing

3.1 CHARACTERISTIC FEATURE THRESHOLDING

The advantages of characteristic feature thresholding techniques are that they give closed boundaries, they are comparatively simple in both computation and implementation, they are more immune to noise than edge detectors, there are many existing methods, and there are many surveys and comparison studies.

The main problem with thresholding techniques is that they have limited application areas because these methods are all based on an assumption that different classes of segments of an image have different modes in the distribution of the chosen features, for instance, intensity. In general, that assumption is not met by many images of wood. Another main drawback of thresholding is that it is difficult in extending gray-scale thresholding to color spaces. The few thresholding algorithms for color images are either too complicated without significant performance gains or not really color image thresholding as their name implies.

In the investigation of defects on veneer surfaces, it was found that thresholding techniques could be used to find holes, most of the loose knots, tight knots, some dark pitch streaks, and some dark blue stains even though most of the veneer images did not have clear bimodal histograms. Compared with some first-order differential edge operators or gradient edge operators such as the Sobel and Roberts operators, manually chosen thresholds worked better for the defects mentioned above. Therefore, although automatic thresholding would not be as good as manually chosen thresholding, it still could be expected to have potential in detecting most of

the common defects on veneer surfaces and might be useful in an application where only knots and holes would be taken into account.

In 1987, Sahoo et al., gave a good survey of thresholding techniques. In their survey, they evaluated nine global thresholding methods, namely the co-occurrence matrix method (1978), Otsu method(1979), Pun method (1980), Johannsen and Bille method (1982), histogram concavity analysis method (1983), Kapur et al., entropy method (1985), Deravi and Pal method (1983), moments preserving method (1985), and minimum error method (1986). A set of test images which had different types of histograms was used. The shape measure and uniformity measure were used for evaluation, where the uniformity measure was the amount of spread of gray-levels from the average gray-level in each region for selected thresholds, and the shape measure was an average of gradients corresponding to selected thresholds. Both measures gave an indication of the homogeneity of segmented regions. From visual inspection of the binary images as well as an objective evaluation based on the uniformity and shape measures, Sahoo et al., concluded that the Otsu method was the best and the Johannsen and Bille, Kapur et al., entropy, and moment-preserving methods were reasonably good methods for different types of histograms such as unimodal, bimodal with a flat valley, and bimodal.

Lee et al., (1990) did a further investigation about performance evaluation of five global thresholding algorithms, namely the Otsu method (1979), Wu et al., quadtree method (1982), Kittler and Illingworth's simple images statistic method (1985), Kapur et al., entropy method (1985), and the Tsai moment-preserving method (1985). Two well chosen images were used. One was a high contrast disk image with large background variance, a rather small object variance, and a large mean gray-scale difference between the object and its background. Another was an image of a crane with gradual shading and roughness in the background, relatively large variances in both object and background, and a small mean difference. Both images had 256 by 256 pixels and 256 gray-levels. The histogram of the "disk" had a sharp peak and a bimodal shape while the "crane" histogram had no clear bimodal shape. The criteria for evaluation were probability of error which reflected the missed

percentage and wrong percentage and Sahoo et al.,'s shape and uniformity measures. Comparisons were made of the change of object size, the mean gray-level value difference between object and background, and the signal to noise ratio. They concluded that: (1) the performance of global thresholding techniques were limited by the object size, the mean difference, the contrast, the variances of object and background, noise etc., and the performances were image dependent, (2) no single algorithm was uniformly top-ranked across test images and performance criteria, and (3) among the five algorithms, Otsu's and Kittler and Illingworth's statistical method showed relatively good performances.

Optical images of softwood veneer have relatively low contrast between defect and clear wood because the dark latewood often has very similar intensity and color values when compared to the defects. In addition, some defect sizes are very small, and the background clear wood is nonuniform because it is comprised of latewood and earlywood. By considering the veneer image characteristics, the algorithm itself, and the limited comparison studies previously discussed, four global thresholding algorithms were selected and adapted. They are the: (1) Otsu method, (2) Kapur et al., entropy method, (3) moment-preserving method, and (4) transition-matrix method.

3.1.1 The Otsu Method

The Otsu method (1979) is one of the maximum likelihood methods utilizing population mixture models under the assumption of a normal distribution. It is based on discriminant analysis and also minimizes the mean square errors between the original image and the resultant binary image. The threshold is determined by maximizing the between-class variance of gray-scales. The mean and variance of gray values are used as features.

Otsu tested his algorithm with five images that were all 64 by 64 pixels in size: a picture of the letter character "A" at 16 gray-levels, two textured pattern images at 64 gray-levels, and two cell images at 256 gray-levels. All showed good results. One of the texture images had a corresponding histogram with two peaks and a very broad

and flat valley. This was usually difficult to threshold. Another texture image had a histogram with one peak and a long left tail. Cell images have been widely investigated, and in the two images used by Otsu, the boundaries both between the nucleus and cytoplasm and between the cytoplasm and background were blurred. Histograms of those two cell images had one clear peak, long tails and a few ambiguous small peaks along the tails. Results of using those two images with three-level thresholding were satisfactory. Cell nucleus, cytoplasm and background were separated well.

The Otsu method has some advantages. First, the procedure is very simple; only the zeroth and the first order cumulative moments of the gray-level histogram are utilized. Second, a straightforward extension to multi-thresholding problems is feasible by virtue of the criterion on which the method is based. Finally, the method is said to be general because it has made no assumptions about the shape of the histogram. It performed well in comparison studies by Sahoo et al., (1987) and Lee et al., (1990).

The cell image is similar to gray-scale veneer images in that they both have comparatively low contrast, no sharp boundaries between objects and background, and very nonuniform backgrounds.

To describe the Otsu method, let the pixels of a given image be represented in L gray-levels {0, 1, 2, ..., L-1}; the number of pixels at level i denoted by n_i; and the total number of pixels by $N = n_0 + n_1 + \ldots + n_{L-1}$. For two-level or two-class thresholding, let 0 denote class 0, 1 denote class 1, and k denote the threshold. The variance of each class is then defined by equations 3.1 and 3.2.

$$\sigma_0^2 = \frac{1}{\omega_0} \sum_{i=0}^{k} (i - \mu_0)^2 p_i \qquad (3.1)$$

$$\sigma_1^2 = \frac{1}{\omega_1} \sum_{i=k+1}^{L-1} (i - \mu_1)^2 p_i \qquad (3.2)$$

Where the mean value for each class is given by

$$\mu_0 = \sum_{i=0}^{k} i\,P_i / \omega_0$$

$$\mu_1 = \sum_{i=k+1}^{L-1} i\,P_i / \omega_1$$

The probability distribution of the ith level p_i and the probability of class occurrence ω are defined by equations 3.3 to 3.5.

$$p_i = \frac{1}{N} \sum_{j=0}^{i} n_j \qquad (3.3)$$

$$\omega_0 = \sum_{i=0}^{k} p_i \qquad (3.4)$$

$$\omega_1 = \sum_{i=k+1}^{L-1} p_i \qquad (3.5)$$

The discriminant function is given by equation 3.6, where the between-class variance is given by equation 3.7 and the within-class variance by equation 3.8.

$$\eta = \frac{\sigma_b^2}{\sigma_w^2} \qquad (3.6)$$

$$\sigma_b^2 = \omega_0(\mu_0-\mu_t)^2 + \omega_1(\mu_1-\mu_t)^2 \qquad (3.7)$$

$$\sigma_w^2 = \omega_0\sigma_0^2 + \omega_1\sigma_1^2 \qquad (3.8)$$

The discriminant criterion η is defined as

$$\eta(k) = \frac{\sigma_b^2(k)}{\sigma_t^2} \qquad (3.9)$$

where σ_t^2 stands for the total variance and is the sum of between-class variance and within-class variance, i.e.,

$$\sigma_t^2 = \sigma_w^2 + \sigma_b^2$$

where σ_t^2 is independent of the threshold level k. The optimal threshold k^* is selected such that η is maximized, or equivalently, σ_b^2 is maximized; that is, select threshold value k^* such that

$$\sigma_b^2(k^*) = \max[\sigma_b^2(k), k=1, 2, \ldots, L-1] \ .$$

The extension of the method to multi-thresholding problems is straightforward. For example, in the case of three-class thresholding, two thresholds are assumed: k_1 and k_2 for each of the three classes, C_0 for [0, 1, ..., k_1], C_1 for [k_1+1, ..., k_2], and C_2 for [k_2+1, ..., L-1]. The criterion measure σ_b^2 (also η) is then a function of the two variables k_1 and k_2, and an optimal set of thresholds k_1^* and k_2^* is selected by maximizing σ_b^2 using equation 3.10.

$$\sigma_b^2(k_1^*, k_2^*) = \max[\sigma_b^2(k_1, k_2), 0 \le k_1 < k_2 < L-1] \qquad (3.10)$$

It should be mentioned that the selection of thresholds becomes quite difficult as the number of classes increases because the criterion measure, σ_b^2, defined in a one-dimensional scale, may gradually lose its meaning as the number of classes increases. A diagram of the algorithm for two-class thresholding is given in Figure 3.1.

3.1.2 The Kapur et al., Entropy Method

Of the entropy thresholding algorithms reviewed in Chapter 2, the Kapur et al., (1985) entropy algorithm performed best in the comparison study by Sahoo et al., (1987). In this method, the probability distributions of the object and background are derived from the original grey level distribution of the image. Then the optimal threshold is defined as the grey level which maximizes the sum of the object and background entropies.

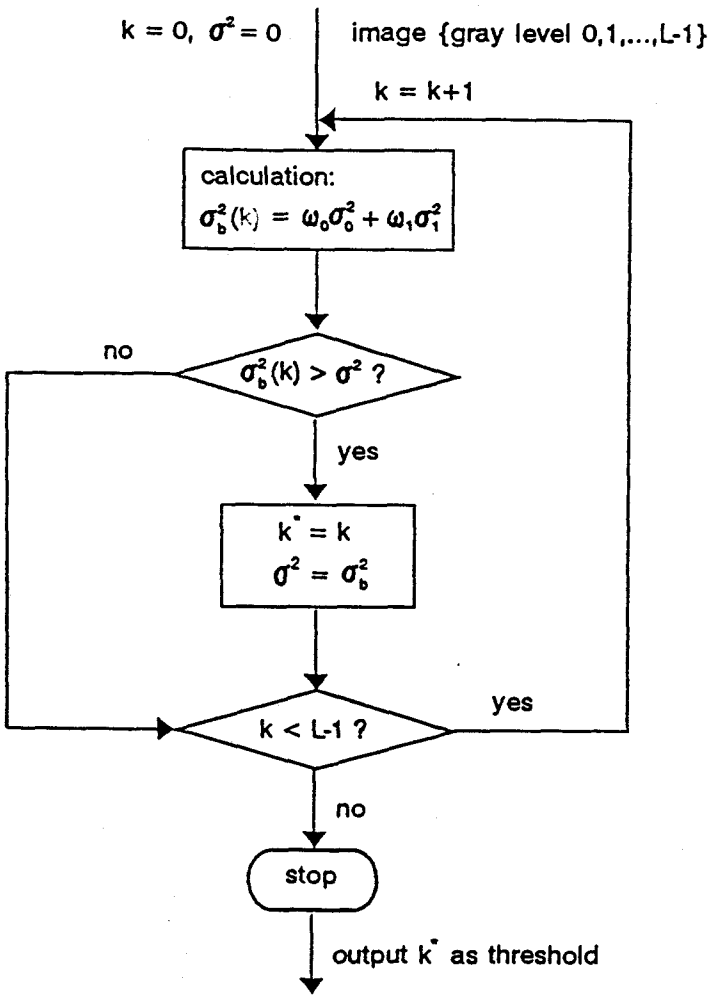

Figure 3.1. A diagram of the Otsu thresholding algorithm for the two-class case.

Kapur et al., tested four real world pictures, an image of a building, an image of a cameraman, and two images of human models. All images had 256 gray-levels and the histograms of three of the images had clear modes while one image had no distinct bimodal histogram. The "building" and "cameraman" histograms had broad

flat valleys. Kapur et al., showed that their method gave a better threshold value when compared to others such as Pun or Johannsen and Bille.

The advantages of the method are that it belongs to a large group of thresholding techniques, it has a well defined information theory model and statistical criterion, it can be extended to multi-thresholding without difficulty, and it does not require clear modals in the histogram.

To describe the Kapur et al., entropy method algorithm, let the pixels of a given image be represented in L gray-levels {0, 1, ..., L-1}. The number of pixels at level i is denoted by n_i and the total number of pixels by $N = n_0 + n_2 + \ldots + n_{L-1}$. For two-level or two-class thresholding, let 0 denote class 0, 1 denote class 1, and k denote the threshold. The entropy of each class is then defined by equations 3.11 and 3.12.

$$H_0 = -\sum_{i=0}^{k} \frac{p_i}{p_k} Log \frac{p_i}{p_k} \qquad (3.11)$$

$$H_1 = -\sum_{i=k+1}^{L-1} \frac{p_i}{1-p_k} Log \frac{p_i}{1-p_k} \qquad (3.12)$$

The probability of level i is defined by equation 3.13.

$$p_i = \frac{1}{N} \sum_{j=0}^{i} n_j \qquad (3.13)$$

The optimal threshold k^* is selected such that the discriminant function defined by equation 3.14 is maximized:

$$H(k) = H_0 + H_1 \qquad (3.14)$$

For the multi-thresholding, suppose $k_1, k_2, \ldots, k_m$ are wanted, then the discriminant function is defined as in equation 3.15.

$$H(k_1, k_2, \ldots, k_m) = H_0 + H_1 + H_2 + \ldots + H_m \qquad (3.15)$$

where

$$H_{i-1} = Log\left(\sum_{j=k_{i-1}+1}^{k_i} p_j\right) - \frac{\sum_{j=k_{i-1}+1}^{k_i} p_j \; Log \; p_j}{\sum_{j=k_{k-1}+1}^{k_i} p_j} \qquad (3.16)$$

$$i = 1, 2, \ldots, m, m+1.$$

and define

$$k_0 = 1, \quad k_{m+1} = L - 1$$

A diagram of the algorithm for two-class thresholding is presented in Figure 3.2.

3.1.3 The Moment-Preserving Method

In the Tsai moment-preserving method, the thresholds are selected in such a way that the moments of the thresholded image are preserved, that is, the moments of the original image are equal to those of the thresholded image. The moments of the original image are first computed, then the moments of the thresholded image are set equal to those moments. The threshold or thresholds are selected by solving the linear moments equations.

The method was applied to four images: letters on a newspaper, a three-color wheel pattern, an image of a cell, and a picture of a girl. The newspaper image and wheel image had bimodal histograms; while the cell and three color wheel images had trimodal histograms. The image of a girl had a four-modal histogram. Results indicated that the pixel classes had been successfully thresholded.

All images tested had clear bimodal or polymodal histograms and no difficult histograms showed. However, in the study by Sahoo et al., (1988), they used an image of a "building" without a distinct bimodal histogram and the moment-preserving algorithm still performed well.

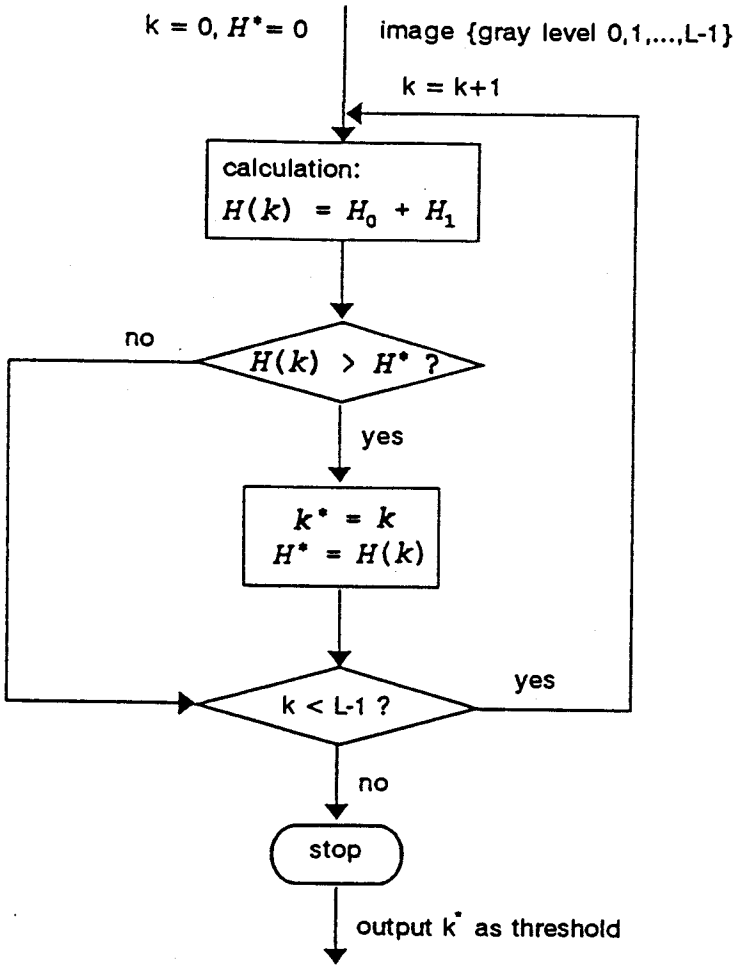

Figure 3.2. The diagram of Kapur et al., entropy thresholding.

The reasons for choosing the moment-preserving algorithm are that it is well defined from a mathematical and statistical point-of-view, it is applicable for multi-level thresholding by solving 2N linear moments equations for N-1 threshold values, and it covers a variety of images.

To describe the moment-preserving method algorithm, when given an image with L gray-levels {0, 1, 2, ..., L-1} and N pixels, where $N = n_0 + n_1 + \ldots + n_{L-1}$

(n_i denotes the number of pixels at level i), then the ith moment m_i of the image is defined as equation 3.17,

$$m_i = \frac{1}{n} \sum_{j=0}^{L-1} p_j j^i \qquad (3.17)$$

where the probability of the jth gray-level is

$$p_i = \frac{n_j}{N}$$

For two-class thresholding, an image can be considered as a blurred version of an ideal bilevel image which consists of pixels with only two gray levels z_0 and z_1 ($z_0 < z_1$). The moment-preserving thresholding is to select a threshold value such that if all gray values below the threshold in the image are replaced by z_0 and all gray values above the threshold are replaced by z_1, then the first three moments of the original image are preserved in the resulting bilevel image. Let p_0 and p_1 denote the fractions of the below-threshold pixels and the above-threshold pixels in the bilevel image, respectively, and m^r denote the moment of resulting bilevel image, then the first three moments of the resulting bilevel image are given by the following equation.
And preserving the first three moments in the resulting image means the following equalities:

$$m_i^r = m_i, \qquad i = 1, 2, 3.$$

where m_i is given by equation 3.17.

$$m_i^r = \sum_{j=0}^{1} p_j z_j^i, \qquad i = 1, 2, 3.$$

Note that

$$p_0 + p_1 = 1$$

and also define m_0 to be 1. To select threshold k, we can then solve the following set of four moment equations 3.18 through 3.21.

$$p_0 z_0^0 + p_1 z_1^0 = m_0 \qquad (3.18)$$

$$p_0 z_0^1 + p_1 z_1^1 = m_1 \qquad (3.19)$$

$$p_0 z_0^2 + p_1 z_1^2 = m_2 \qquad (3.20)$$

$$p_0 z_0^3 + p_1 z_1^3 = m_3 \qquad (3.21)$$

where:

$$p_0 = \frac{1}{N} \sum_{i=0}^{k} n_i$$

and:

$$p_1 = \frac{1}{N} \sum_{i=k+1}^{L-1} n_i = 1 - p_0$$

To find the desired threshold value k^*, equations 3.18 to 3.21 are first solved to obtain p_0 and p_1, and then the threshold k^* is chosen as the p_0-percentile of the histogram of the image as in equation 3.22.

$$p_0 = \frac{1}{N} \sum_{j \le k^*} n_j \qquad (3.22)$$

In practice, there may exist no discrete gray value exactly at the p_0-percentile of the histogram. In such a case, the threshold k^* should be chosen as the gray-level closest to the p_0-percentile. This algorithm is diagramed in Figure 3.3.

The two-class thresholding can be easily extended to multi-class thresholding. To threshold an image into M classes, M-1 threshold values are needed and 2M-1 moment-preserving equations must be solved.

3.1.4 The Transition-Matrix Method

All three of the previously discussed algorithms are histogram-based and utilize information about gray-level distributions only. Since non-histogram-based thresholding algorithms are proposed to overcome the shortcomings of histogram-

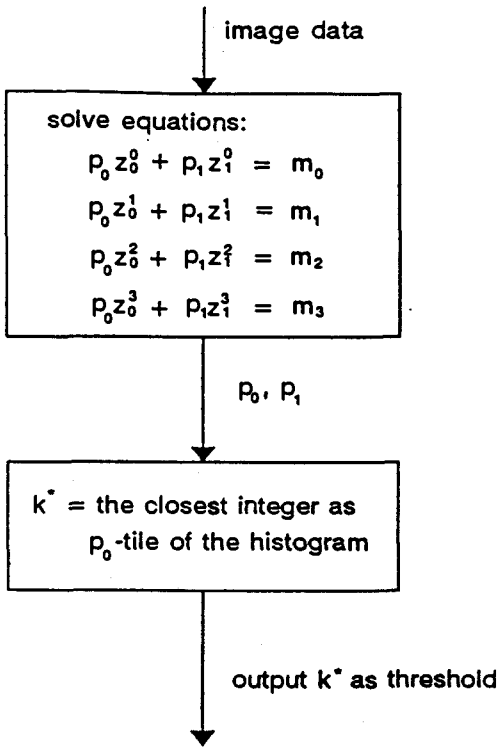

Figure 3.3. A diagram of the moment-preserving thresholding algorithm for the two-class case.

based ones by employing spatial information, they are expected to result in more meaningful segmentation. The special information-based or non-histogram-based algorithms try to use second-order statistics rather than just the first-order histogram. The commonly used co-occurrence matrix contains position information about pixels with the gray-level distribution. The Deravi and Pal method (1983) used the transition matrix similar to the co-occurrence matrix but reduced the computational burden. The element (i, j) of the transition matrix specifies how frequently the ith gray-level is followed by the jth gray-level in the specified horizontal and vertical spatial displacements. For two-class thresholding, the total number of transitions of each class is computed. Two measures, which are the estimates of the joint and conditional probabilities of the intensity transition between the classes, are used as discriminant

A 64-levelfunctions. These measures indicate the spatial discontinuity of the segmented regions. Therefore, it is conjectured that meaningful sets of thresholds would correspond to the minima of the two measures. Since one of the measures is not directly related to the histogram, it is expected that it would exhibit minima even for unimodal histograms. picture of a boy on a boat with an unimodal histogram was tested. Such a histogram is typical for images with many different small objects such as natural outdoor scenes or aerial photographs. The result was acceptable that the main object (boy, building, and bank) and background (water and air) were well separated. The extension to multi-thresholding is straight-forward by finding the second or third minima of the discriminant functions.

There were no comparison studies about non-histogram-based global thresholding. The reasons for choosing this algorithm are that first, the co-occurrence matrix is widely used by many researchers; second, the criteria for threshold selection is well defined and easy to understand; and third, it is claimed that this algorithm is particularly designed for a unimodal histogram which is the case for images of veneer.

In an experiment for the research reported in this thesis, there were some cases where a conflict occurred in the two measures when applying Deravi and Pal's transition-matrix method, that is, the threshold values given by those two criteria were unequal. In such cases, it is difficult to decide which threshold value should be used. The better way to do it is using one criterion. It was noticed that for a bimodal histogram, two measures always gave the same threshold value. However, since the curve of conditional probability will exhibit minima even for unimodal histograms, while the curve of joint probability would not, the conditional probability minima was chosen as the only measure for finding the threshold.

In the transition-matrix method algorithm, given an image with M x N pixels and L gray-levels

$$X = [X_{m,n}: m = 1, \ldots, M; \; n = 1, \ldots, N]$$

where $x_{m,n} = k$; k = 0, 1, 2, ..., L-1, an L x L transition matrix is defined for the intensity transition between adjacent pixels on a horizontal direction from left to right and the vertical direction from bottom to top as shown in equation 3.23.

$$T = [n_{ij}]$$

$$n_{ij} = \sum_{n=1}^{N-1} \sum_{m=1}^{M} [x_{m,n} = i \wedge x_{m,n+1} = j] + \sum_{n=1}^{N} \sum_{m=1}^{M-1} [x_{m,n} = i \wedge x_{m+1,n} = j] \quad (3.23)$$

$$i, j = 0, 1, \ldots, L-1$$

The first part of the summation equation is in the horizontal direction from left to right and the second part is in the vertical direction from bottom to top. If a threshold value k is chosen , then the image is divided into two classes C_0(k) and C_1(k):

$$C_0(k) = [x : x = 0, 1, \ldots, k]$$

$$C_1(k) = [x : x = k+1, \ldots, L-1].$$

Accordingly, it leads to the definition of four regions in the transition matrix

$$T = \begin{array}{c} \\ L-1 \\ \vdots \\ k+1 \\ k \\ \vdots \\ 0 \\ \underbrace{0\ 1\ \_\ k}_{C_0}\ \underbrace{k+1\ \_\ L-1}_{C_1} \end{array} \begin{array}{c} \\ \left.\begin{array}{c} \\ \\ \\ \end{array}\right\} C_1 \\ \left.\begin{array}{c} \\ \\ \\ \end{array}\right\} C_0 \end{array} \begin{bmatrix} a & c \\ d & b \end{bmatrix}$$

where

$$a = \sum_{i=0}^{k} \sum_{j=0}^{k} n_{ij}$$

$$b = \sum_{i=k+1}^{L-1} \sum_{j=k+1}^{L-1} n_{ij}$$

$$c = \sum_{i=0}^{k} \sum_{j=k+1}^{L-1} n_{ij}$$

and

$$d = \sum_{j=k+1}^{L-1} \sum_{j=0}^{k} n_{ij} \quad .$$

The discriminant function is given by equation 3.24.

$$p(k) = \frac{1}{2}\left(\frac{c}{a+c} + \frac{d}{b+d} \right) \qquad (3.24)$$

The first part of the summation in the equation is an estimate of the conditional probability of C_0 to C_1, and the second part is an estimate of the conditional probability of C_1 to C_0.

The threshold is selected such that p(k) is a minimum because the lower the value of p(k), the lower the probability that the next transition will be a significantly different intensity class. This algorithm is shown in Figure 3.4.

3.1.5 Global Thresholding Versus Local Thresholding

The four algorithms selected are all global thresholding approaches. In this thesis research, both global and local thresholding were considered in order to possibly obtain better performance because previous research in the Department of Forest Products had found that local thresholding provided better results than global thresholding for defect detection on veneer surfaces (Maristany et al., 1993). Because the clear wood in an image of veneer includes both latewood and earlywood, the backgrounds found in veneer images are far from uniform. The latewood is obviously darker than earlywood in Douglas-fir. The threshold value selected by an algorithm usually will either not only find the defect and some dark latewood or it will not find the defect at all.

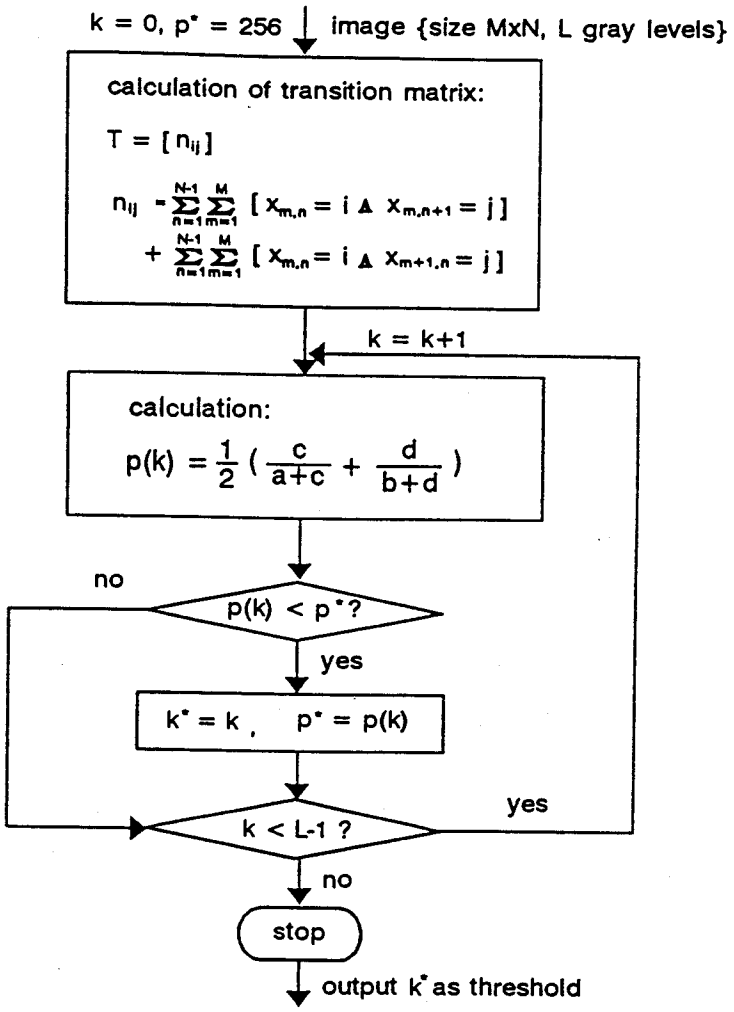

Figure 3.4. A diagram of the transition-matrix thresholding algorithm.

To improve performance, the strategy of clear wood testing is used to eliminate the latewood from the thresholded image. To do this, the Sobel edge detector is used to generate the edge strength map of the whole image. Then a critical edge strength value is found. It is determined by establishing a threshold such that a given percentage of pixels will have values equal to or larger than that critical edge

strength value in the edge strength map. This is called the percentage of upper-tail edge strengths.

In the clear wood testing, the whole image is divided into square tiles. For tiles whose edge strengths are equal to or larger than the critical edge strength value, the number of pixels with values greater than a certain predetermined threshold are counted. Any tile with the high pixel count is then considered as a potential defective tile and not a clear wood tile. The idea is that the edge strength should be high when defects appear in a region since most of the defects have slightly sharper boundaries as compared to those inside the clear wood.

Global thresholding has four steps:

(1) global thresholding is applied to the whole image to select a threshold value;

(2) the Sobel edge operator is applied to the whole image to obtain the edge strength map and find the critical edge strength value corresponding to a given percentage of the upper-tail edge strengths;

(3) the whole image is divided into square windows or tiles (except that the right most and top most tiles may be rectangular depending on the whole image size and the tile size), and a clear wood test is applied to each tile to see whether or not the tile contains only clear wood; and

(4) if the tile is a non-clear wood region, the threshold value found in (1) is used to threshold the tile into a binary image (white to the clear wood and black to the defect); otherwise, set all pixels in the tile to white as the tile is a clear wood tile.

The procedure for global thresholding is shown in Figure 3.5. As mentioned in the literature review, a fixed threshold chosen based on overall information about an image, no matter how well-chosen, sometimes can not perform well in cases of uneven background, poor illumination or high noise. This seems to be the case in images of veneer where the background is very uneven. Moreover, the intensity of some defects is very similar to or even lighter than that of the latewood. Hence, local

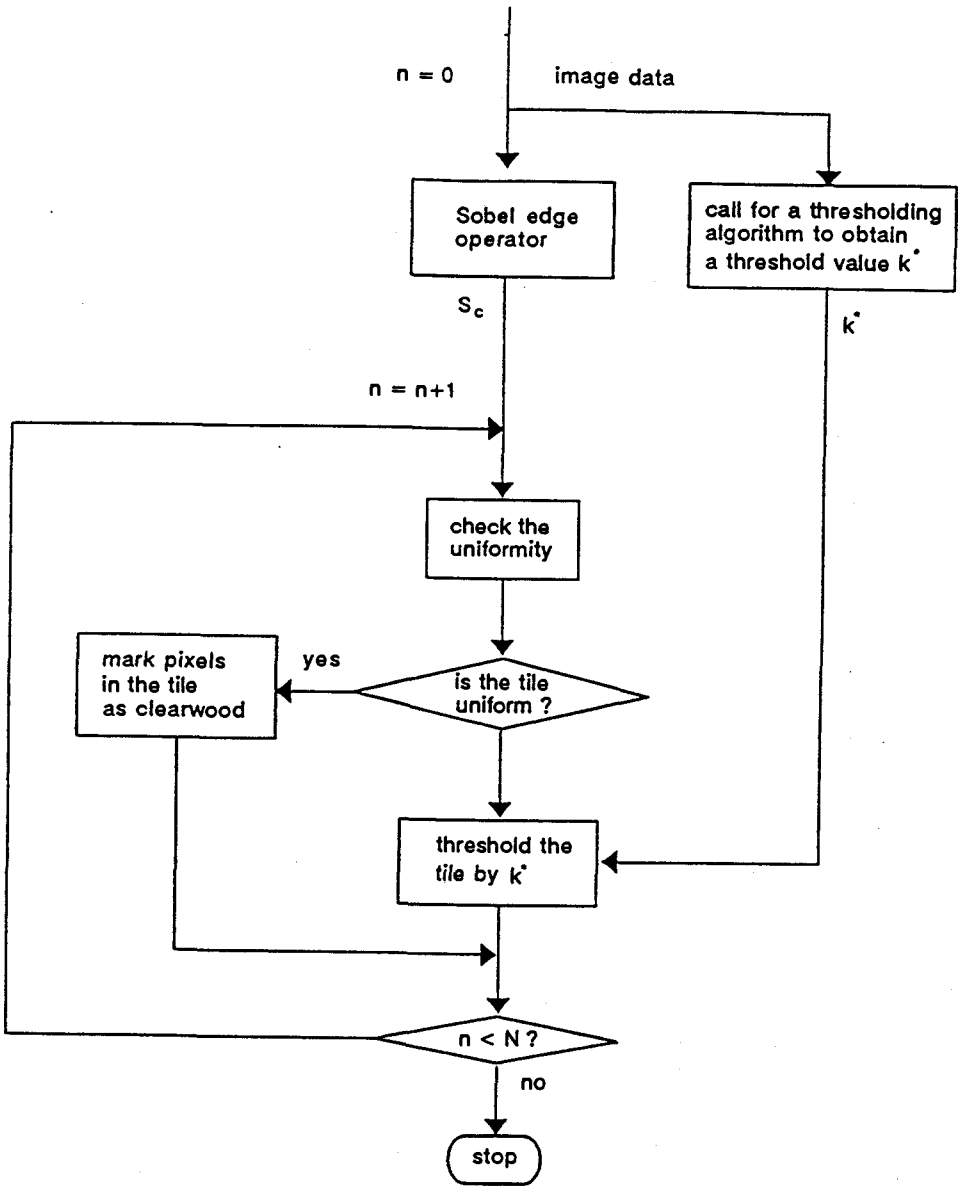

Figure 3.5. A diagram of the global thresholding approach.

thresholding was considered to eliminate false defect detection and to increase the possibility that light defects can be detected. All four global thresholding algorithms were adapted to their corresponding local thresholding versions.

In local thresholding, the original image is divided into smaller regions or tiles, and the global thresholding method is used to determine the threshold for each region.

As mentioned in Chapter 2, there are some problems in local thresholding. First, if the window contains no object, then attempting to threshold is dangerous, since many of the algorithms will select threshold values if the region is not perfectly uniform. Second, if the value of the threshold is set, then it might be a problem because the number of objects is different in each different window. Third, if an object overlaps several windows, then it may be represented as several objects since several different thresholds may be used. The final problem is the computational time and memory.

The main problem in this thesis project is the first one, thresholding the regions containing only background or clear wood. Therefore, the same clear wood testing strategy used in the global thresholding was used to determine the clear wood tiles. The second and third problems do not present any difficulty because only two-class thresholding is considered in this study.

Local thresholding has three steps:

(1) the Sobel edge operator is applied to the whole image to obtain the edge strength map and to find the critical edge strength value corresponding to a given value of the percentage of upper-tail edge strengths;

(2) the whole image is divided into square windows or tiles (except that the right most and top most tiles may be rectangular depending on the whole image size and the tile size), and a clear wood test is applied to each tile to see whether the tile contains only clear wood; and

(3) if the tile is a non-clear wood tile, a global thresholding algorithm is used to find a threshold value for that tile, and the tile is thresholded; otherwise, set all pixels of the tile to clear wood.

The procedure for local thresholding is given in Figure 3.6.

3.1.6 Post-Processing After Thresholding

The binary image obtained by thresholding (either global or local) is not clean; there are small spots in the background corresponding to the very dark latewood and

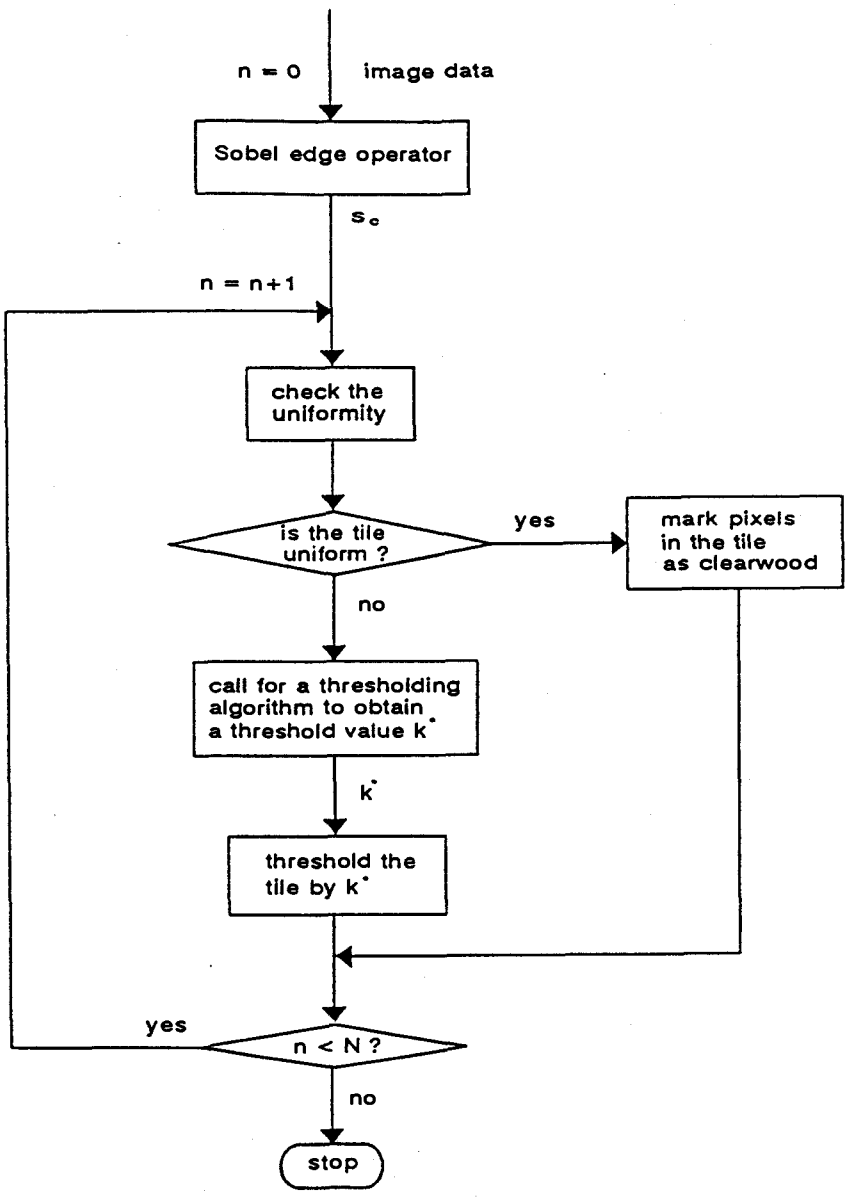

Figure 3.6. The diagram of the local thresholding.

some small white spots or holes inside the detected defects corresponding to light defect portions. Therefore, post-processing is required to clean up the image to obtain a better result.

One of the functions of a post-processing algorithm is to clean the image by removing incorrectly marked spots in the clear wood. This is achieved by setting an acceptable size of defect according to the smallest defect size discernable in the application. In this project, the acceptable size was set to be five pixels. If the size of a black blob is less than the predetermined acceptable size, then the blob is deleted. Another function of post-processing is to fill the holes inside the detected defects because all defects concerned in the study are supposed to be solid. This was done by filling the holes of a defect blob.

3.2 EDGE DETECTION

One of the most important visual cues in nature or in any image is the existence of sharp dark or light edges. Therefore, many edge detection techniques have been developed in image segmentation. An important advantage of edge detection as compared to region techniques is that most of the edge detectors have relatively good mathematical models. Thus, the validity of edge detection methods holds over a larger variety of images than the region methods.

One problem with edge detection techniques is that they do not perform well when the transition between regions is not abrupt enough. This might be the case where some defects on veneer surfaces have no obvious changes in intensity and color, such as pitch streak or blue stain. Another problem is that the determination of the meaningful regions under detected edges can be very difficult because the detected boundaries of objects are often not closed. Therefore, edge following and thinning processes must be utilized to obtain the one-pixel width closed boundary of an object. Edge detection algorithms are sensitive to noise and can be computationally expensive.

Though it sounds as if edge detection algorithms might not be very suitable for some types of detect detection on veneer surfaces, they worked for knots and pitch pockets in previous research in the Forest Products Department. Therefore, they are still considered in this project.

Although there are many edge detection algorithms, not many systematic comparison studies exist. The references for methods of choosing algorithms are even less than those for thresholding.

In sequential edge detection, the already-detected edge pixels exert an influence on both the position of the next potential edge pixel and the result of its acceptance test. Therefore, appropriate application areas for sequential edge detection techniques are where a considerable body of a priori knowledge about the shape or the position of the edges in an image is known and also the background must be more or less uniform as compared to the objects. Since most of the defect edges in veneer images are mixed up with lots of edges between latewood and earlywood and the position and shape of a defect are not predictable, the sequential edge detection techniques did not appear to be proper for defect detection on images of veneer.

In parallel edge detection, the algorithms which are most popular and have good mathematical foundations, were considered first. Three edge detectors were chosen, the compass gradient mask algorithm, nonlinear Laplace algorithm, and Shiozaki's entropy algorithm.

3.2.1 Compass Gradient Mask Edge Detection

Robinson (1977a) described an edge detection algorithm which made use of 3 by 3 compass gradient masks. The edge angles are quantized in eight equally spaced directions. A set of compass gradient masks can be formed by rotating any first-order differential operator such as the Kirsh, Prewitt, or Sobel. To reduce the burden of computation, he proposed a set of masks called five-level simple masks such that only four of them were needed to obtain eight directions of gradients. The compass edge masks were applied to three images (a tank, a girl's picture, and an aerial scene) and all had acceptable results.

Robinson (1977b) extended the compass gradient masks to color images. The color edges are obtained by comparing the 24 gradient values in three color components and eight directions. The gradient that gives the maximum value at a

pixel point thus determines the direction of the edge, whether due to a discontinuity in the first, second or third color component. The binary edge map is generated by using local connectivity in the edge direction map and a threshold map which uses a locally adaptive threshold at each point of the image. A color image of the face of a model was used to demonstrate the effectiveness of the method.

A common problem of the gradient operator is its dependency on predetermined thresholds to get the edges from the edge map. Robinson solved this problem by automatically selecting local thresholds. Also, the local connectivity checking was said to improve the performance of the compass masks.

Gradient edge operators or first-order differential operators are a large class of edge detection methods because the use of a gradient to detect discontinuity is mathematically natural and feasible. The different digital approximation schemes of a gradient or differentiation produce different gradient edge detectors. Gradient edge operators have been proven to perform well for many types of images.

In understanding defects on veneer surfaces, Sobel and Roberts operators were investigated in the project. It was found that the edge strength maps they generated showed the edges of most tight knots, loose knots, and pitch pockets but not light blue stain or light pitch streaks. Some of the boundaries between latewood and earlywood also appeared and the defect boundaries often had gaps. If both edge pixel strength and edge pixel angle are considered (the Sobel and Roberts operators considered only the edge strength), better results could be expected. Also, the compass gradient masks are capable of detecting the edge element in six directions other than just north-south and east-west as the Sobel edge detector does. Therefore, the compass edge detection method was chosen to be used in this study and five different gradient operators were included in it.

In the course of the research, it was found that the locally adaptive threshold suggested by Robinson did not work well for veneer images because too many small edges between latewood and earlywood were inside the image background. Instead, a fixed threshold for the gradient image was found to be more suitable. The threshold

for the edge strength map was chosen by hand as was the case in most of the edge detection algorithms.

One potential improvement that could be made is the preprocessing of the original images. Because of the appearance of many small edges inside the clear wood and also inside some defects, a preprocessing algorithm would be helpful in eliminating them. By applying median, mean, and Gaussian filtering, it was found that a median filter performed best. This is because the median filter can smooth an image without blurring the main edges.

The compass edge detection has the following three steps and its block diagram is shown in Figure 3.7.

(1) Smooth the original image using a color median filter;

(2) Obtaining the gradient image and edge direction map for the original image. The gradient image is obtained by taking the maximum gradient magnitude of twenty-four convolution results (i.e., eight compass gradient masks in three color components) at each pixel. The mask which produces the maximum gradient magnitude determines the direction of the edge at that pixel, which is one of the eight numbers from 0 to 7 as shown in Figure 3.8. Eight compass gradient masks are formed by rotating the differentiation mask in the x or y direction. Some of typical gradient operators' compass sets are shown in Figure 3.9.

(3) obtaining the binary edge map. If the edge direction vectors in a 3 x 3 grid surrounding a point satisfy the local connectivity conditions and its gradient value is greater than the threshold value, then that point is confirmed as an edge pixel in the binary edge map. The edge direction map is used to determine the local connectivity as shown in Figure 3.10. If the direction at the center of the 3 x 3 grid is k (k= 0, 1,..., 7), and if the directions of the preceding and succeeding edge vectors are k-1, k, or k+1 (mod 8) for any of the eight compass directions, then the edges are considered as connected.

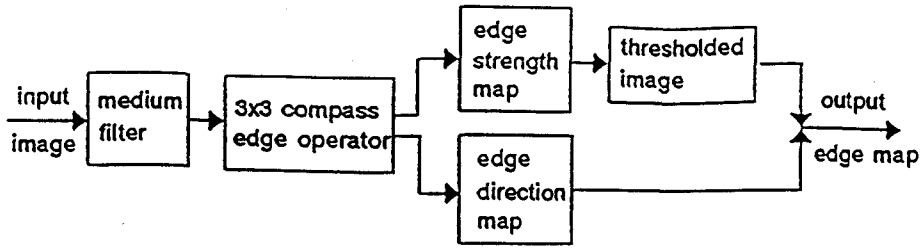

Figure 3.7. A block diagram of the compass gradient mask edge detection algorithm.

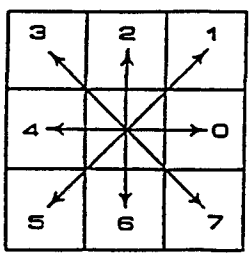

Figure 3.8. The eight principal edge directions on a 3 x 3 grid.

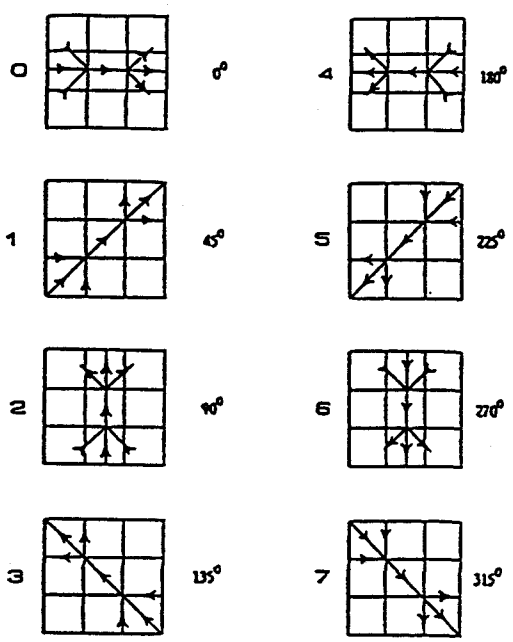

Figure 3.9. Local connectivity rules.

direction of edge	Prewitt masks	Kirsch masks	Sobel masks	Five-level masks
0	$\begin{bmatrix} 1 & 1 & 1 \\ 1 & -2 & 1 \\ -1 & -1 & -1 \end{bmatrix}$	$\begin{bmatrix} 5 & 5 & 5 \\ -3 & 0 & -3 \\ -3 & -3 & -3 \end{bmatrix}$	$\begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$	$\begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$
1	$\begin{bmatrix} 1 & 1 & 1 \\ 1 & -2 & -1 \\ 1 & -1 & -1 \end{bmatrix}$	$\begin{bmatrix} 5 & 5 & -3 \\ 5 & 0 & -3 \\ -3 & -3 & -3 \end{bmatrix}$	$\begin{bmatrix} -1 & -2 & 0 \\ -1 & 0 & 1 \\ 0 & 2 & 1 \end{bmatrix}$	$\begin{bmatrix} 2 & 1 & 0 \\ 1 & 0 & -1 \\ 0 & -1 & -2 \end{bmatrix}$
2	$\begin{bmatrix} 1 & 1 & -1 \\ 1 & -2 & -1 \\ 1 & 1 & -1 \end{bmatrix}$	$\begin{bmatrix} 5 & -3 & -3 \\ 5 & 0 & -3 \\ 5 & -3 & -3 \end{bmatrix}$	$\begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix}$
3	$\begin{bmatrix} 1 & -1 & -1 \\ 1 & -2 & -1 \\ 1 & 1 & 1 \end{bmatrix}$	$\begin{bmatrix} -3 & -3 & -3 \\ 5 & 0 & -3 \\ 5 & 5 & -3 \end{bmatrix}$	$\begin{bmatrix} 0 & 2 & 1 \\ -1 & 0 & 1 \\ -1 & -2 & 0 \end{bmatrix}$	$\begin{bmatrix} 0 & -1 & -2 \\ 1 & 0 & -1 \\ 2 & 1 & 0 \end{bmatrix}$
4	$\begin{bmatrix} -1 & -1 & -1 \\ 1 & -2 & 1 \\ 1 & 1 & 1 \end{bmatrix}$	$\begin{bmatrix} -3 & -3 & -3 \\ -3 & 0 & -3 \\ 5 & 5 & 5 \end{bmatrix}$	$\begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$	$\begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$
5	$\begin{bmatrix} -1 & -1 & 1 \\ -1 & -2 & 1 \\ 1 & 1 & 1 \end{bmatrix}$	$\begin{bmatrix} -3 & -3 & -3 \\ -3 & 0 & 5 \\ -3 & 5 & 5 \end{bmatrix}$	$\begin{bmatrix} 1 & 2 & 0 \\ 1 & 0 & -1 \\ 0 & -2 & -1 \end{bmatrix}$	$\begin{bmatrix} -2 & -1 & 0 \\ -1 & 0 & 1 \\ 0 & 1 & 2 \end{bmatrix}$
6	$\begin{bmatrix} -1 & 1 & 1 \\ -1 & -2 & 1 \\ -1 & 1 & 1 \end{bmatrix}$	$\begin{bmatrix} -3 & -3 & 5 \\ -3 & 0 & 5 \\ -3 & -3 & 5 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix}$	$\begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$
7	$\begin{bmatrix} 1 & 1 & 1 \\ -1 & -2 & 1 \\ -1 & -1 & 1 \end{bmatrix}$	$\begin{bmatrix} -3 & 5 & 5 \\ -3 & 0 & 5 \\ -3 & -3 & -3 \end{bmatrix}$	$\begin{bmatrix} 0 & -2 & -1 \\ 1 & 0 & -1 \\ 1 & 2 & 0 \end{bmatrix}$	$\begin{bmatrix} 0 & 1 & 2 \\ -1 & 0 & 1 \\ -2 & -1 & 0 \end{bmatrix}$

Figure 3.10. Examples of compass gradient masks.

3.2.2 Nonlinear Laplace Edge Detection

The Laplacian operator itself is a mathematical operator in continuous space and must be represented by an appropriate digital filter. A Laplacian operator is rotation invariant. This has the advantage that high spatial frequencies in all orientations are equally enhanced, but this may also be a disadvantage in that useful directional information is not available. One of the disadvantages is its sensitivity to high frequency noise so the Laplacian operator is generally a useless tool in images with medium or low signal-to-noise ratios.

The Marr-Hildreth operator consists of a combination of a band pass filter, (i.e., Gaussian smoothing filter), and a high pass filter (i.e., linear Laplace filter). The use of a Gaussian filter improves the accuracy of the Laplace filter since the

Gaussian satisfies two conflicting constraints, which are the bandlimited smoothing filter in the frequency domain and localized smoothing filter in the spatial domain. This edge operator forms the base for a second-order derivative edge detector and is widely used.

Beckers (1986) proposed a nonlinear Laplace-like operator in a local 3 x 3 neighborhood which may perform better than a linear Laplace filter. To remove the disadvantage of the linear Laplace filter, Vliet et al., (1989) suggested a nonlinear Laplace filter based on the Marr-Hildreth model combined with a nonlinear Laplace operator to develop an edge detection scheme robust enough to perform well over a wide range of signal-to-noise ratios. It takes the second-order derivative in the most relevant direction (perpendicular to the local edge direction), the operator can have either a square or circular shape, and the filter size can be any value rather than just three, thus increasing the noise immunity.

In an experiment of using a chessboard image and a circle image with Gaussian noise added (signal-to-noise ratio from 1 to 100) and based on Pratt's figure-of-merit performance measure, Vliet et al., concluded that their nonlinear Laplace edge detector performed better than the Haralick cubic polynomial fitting model, Rosenfeld algorithm, Roberts gradient operator, and was comparable to the Marr-Hildreth operator but offers a faster implementation and any size of square and circular shapes of the filter window could be used (in their experiment, sizes 3, 5 and 7 were studied). Two real world images were tested, a picture of a model and a picture of a woman wearing a hat. They reported that the nonlinear operator performed at least as well as the Marr-Hildreth operator and better than the other operators mentioned.

One reason to use this edge operator is that it is basically from the Marr-Hildreth operator. Marr and Hildreth's edge detection theory on which the operator was proposed is one of few mathematical and theoretical descriptions and analyses of edge detection. The performance of nonlinear Laplace edge detector is comparative to the Marr-Hildreth operator but has an advantage over the Marr-Hildreth operator in that it can use any window size needed for the noise and edge type. The window size is a useful factor if it is possible to choose it because usually, the smaller the

window size is, the more accurate the detected edge position is, while the larger the window size, the more sensitive the edge detector is to ramp edges.

In this thesis, the nonlinear Laplace edge detector described by Vliet et al., that was only for gray-scale images was extended to color images. In the tri-parameter color image, the Gaussian smoothing filter is applied to each channel of the color image, the nonlinear Laplace filter with a square shape and sizes 3, 5 and 7 are also applied to each channel of color image, and then the maximum and minimum among all three channels in a filter window are used to calculate the output of the nonlinear Laplace operator, that is, the zero-crossing and the edge strength of the center point of that window.

One of the important tasks in applying Laplacian-like edge detectors is detecting the zero-crossings. It is the task of the zero-crossing detector to assign the changes in the sign of the Laplace filtered image. In Vliet's zero-crossing detector, when regions of opposite sign touch each other and these regions have been separated by a one-pixel width strip of zero value pixels, these pixels are the edge position, or zero-crossing. If the zero value pixels are over one-pixel wide then each zero value pixel will be assigned to the nearest region and the border of the positive region represents the zero-crossing. In such a way, a one-pixel width edge map will be guaranteed.

In this thesis, the outer most pixels of edges of detected objects were used as the boundary pixels (detailed discussion in 3.2.4.) so no one-pixel width edges were requested. It was also noticed that the edge strength maps obtained by the nonlinear Laplace edge detector represented desired defect edges very well. The use of the pixels coincident in both the thresholded edge strength map and the zero-values in nonlinear Laplacian filtered image as the resultant edges were acceptable. Therefore, zero-crossings were not very beneficial and were omitted. As in the compass edge detector, the threshold for the edge strength map was selected manually.

The diagram of the nonlinear Laplace edge detector is shown in Figure 3.11. The Gaussian smoothing filter is given by equation 3.25,

$$G_\sigma(x,y) = \frac{1}{2\pi\sigma^2} \sum_{i=x-S_g}^{x+S_g} \sum_{j=y-S_g}^{y+S_g} \exp\left[-\frac{i^2+j^2}{2\sigma^2}\right] I(i,j) \qquad (3.25)$$

where σ is the standard deviation of the Gaussian filter and I(i, j) is the image value of the pixel (i, j) in one of the three-parameter color coordinates. The S_g is chosen equal to the smallest integer larger than 2σ for the following reasons: (1) the digital filter must be a good fit relative to the continuous Gaussian function, because choosing too small an S_g would produce artifacts attributable to the side lobes that occur in the frequency domain, (2) large values of S_g slow the computational speed considerably, and (3) 95% of the area under a one-dimensional Gaussian curve lies between plus and minus 1.96σ from the mean. The window size is $2S_g + 1$ in the horizontal and vertical directions. The square-shaped nonlinear Laplace operator can be obtained using equation 3.26,

$$NLLAP(x,y) = G_{max}(x,y) + G_{min}(x,y) \qquad (3.26)$$

where

$$G_{max}(x,y) = \max[I(i,j), i=1,2,\ldots,n, j=1,2,\ldots,n] - I(x,y)$$

$$G_{min}(x,y) = \min[I(i,j), i=1,2,\ldots,n, j=1,2,\ldots,n] - I(x,y)$$

where n stands for an n x n window, I stands for the image value of the central pixel in the window, G_{max} stands for the maximum gradient value, and G_{min} stands for the minimum gradient value. The edge strength is given by equation 3.27,

$$I_{strength}(x,y) = \min[G_{max}(x,y), -G_{min}(x,y)] \qquad (3.27)$$

where $I_{strength}$ stands for the edge strength value of the pixel (x, y).

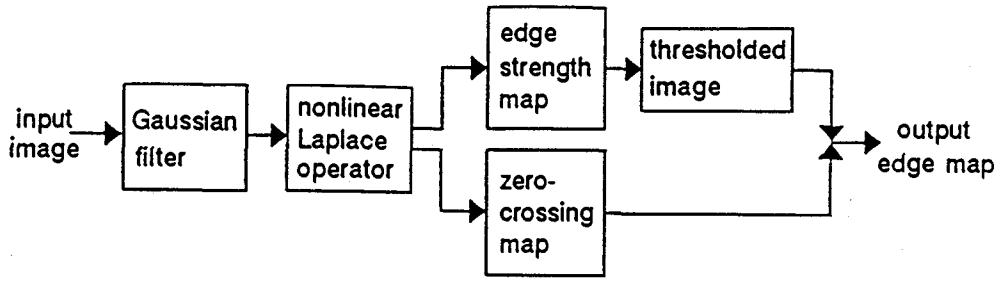

Figure 3.11. A block diagram of the nonlinear Laplace edge detection algorithm.

3.2.3 Shiozaki's Entropy Edge Detection

Shiozaki (1986) presented an edge operator called the entropy operator which extracts edges using the entropy of intensity in a local region of an image which is a 3 x 3 operation window. Entropy of intensity is a probabilistic information measure of the rate of change in intensity in a region. Since the entropy is small when the change in intensity is severe and large when the change in intensity is small, the edges may be extracted by the detection of the regions where the entropy is small. The entropy of color is defined and the color edge can be detected in a similar way.

Shiozaki applied the operator to two monochrome pictures, "GIRL" and "COUPLE", both with 256 by 256 pixels and 256 brightness levels. These images are popularly used to verify the effectiveness and reliability of algorithms since their gray-scale histograms are complicated, typical, and the edges of the images are complicated, too. The image "GIRL" has somewhat low contrast, and the image "COUPLE" has high contrast. The results showed that the entropy operator could create a smooth edge map without noisy pixels. Many details were clearly shown in the image "GIRL". Another image used was a popular aerial picture of size 256 by 256 with 256 gray-levels, showing the streets and buildings of a city. It is a high contrast picture but contains very complicated small edges. Results on this image were worse than those of "GIRL" and "COUPLE" with many small parts and details

lost. This is a common result in aerial images because of their complexity, high noise and lack of detail. Finally the author applied the entropy operator to color images of "GIRL" and "COUPLE" based on the RGB color space. The detected edges of "GIRL" were not improved significantly but the results for "COUPLE" were substantially improved by utilizing the color information.

Like the Laplace operator, the entropy operator is rotation invariant. However, unlike the Laplacian and the gradient operators which depend only on the rate of change in brightness, the nonlinear entropy edge detector depends not only on the rate of change but also on the average value in a local region or operation window. Though this average window information is very little when compared to that found in thresholding where a larger window or even the whole image is employed, it is still a kind of compensation for edge detection, because one of the shortcomings of the edge detection technique is its heavy use of local information and difficulty in utilizing global information.

This algorithm was chosen because it is sensitive to the edges in dark regions, it utilizes the change rate and average value in the operation window, and because it is simple to implement. The sensitivity to dark region may be helpful in the veneer images because most of the defects are darker than clear wood so that the comparatively lighter edges between latewood and earlywood inside the clear wood would be compressed so that the defect edges would stand out.

However, in the course of this thesis research, it was determined that the differences of entropies in a veneer image were very small due to the normalization of entropy and the low contrast edges found in veneer images. Therefore, a weighting factor for entropy was introduced in order to enhance the entropies corresponding to the stronger edges between defect and clear wood. A color median filter was also used for preprocessing the original image to eliminate some small weak edges between latewood and earlywood. The threshold for the binary image was selected manually.

The entropy edge detector for a color image is described in the following paragraphs.

Let the pixel value in the center of the 3 x 3 window be a_0, and its 8 neighboring pixel values be a_1, a_2, ..., a_8, respectively. Then the entropy of pixel the center pixel is defined by equation 3.28,

$$H = -\frac{1}{\log\ (n+1)} \sum_{i=0}^{n} p_i \log\ p_i \qquad (3.28)$$

where

$$p_i = \frac{a_i}{w_1 \sum_{j=0}^{n} a_j}$$

where n = 8, and w_1 is a weighting factor and set to be 9.0.

Similarly, for a color image, let H_R, H_G, and H_B be the entropies of the R, G, and B components of a color image, respectively. Let a_R, a_G, and a_B be, respectively, the values of the pixel of the R, G, and B components of the color image. Then the entropy H of the center pixel in the 3 x 3 window of the color image is defined by equation 3.29,

$$H = w_2(q_R H_R + q_G H_G + q_B H_B) \qquad (3.29)$$

where w_2 is a weighting factor, and

$$q_R = \frac{a_R}{a_R + a_G + a_B}$$

$$q_G = \frac{a_G}{a_R + a_G + a_B}$$

$$q_B = \frac{a_B}{a_R + a_G + a_B}$$

The entropy H is normalized so that the range of H is from 0 to 1. For the purpose of display and thresholding, the weighting factor w_2 is used. A threshold value can be used to find a binary edge map. The diagram for the entropy edge detector is depicted in Figure 3.12.

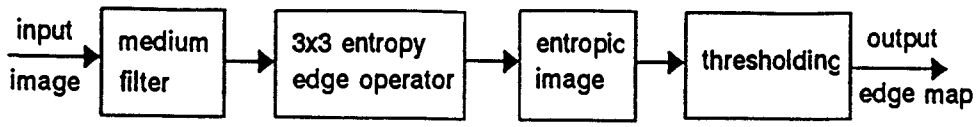

Figure 3.12. A block diagram of the entropy edge detection algorithm.

3.2.4 Edge Following

As mentioned before, the detected boundaries of objects as determined by edge detection techniques are often not closed or not one-pixel in width. Therefore, edge following and thinning processes must be utilized to obtain the one-pixel width closed boundary of an object. For edge thinning, there are many algorithms and most of them are developed for machine recognition of patterns. Lam et al., (1992) wrote a survey about thinning methods, which discussed a wide range of algorithms including iterative deletion of pixels and nonpixel-based methods. Most commonly, a priori information about the object and end-use purpose are required. There are some methods to perform edge following, for instance image intensity interpolation by Nalwa (1987), to avoid the gaps occurring in a binary edge map. However, most commonly edge following for closing the boundary of object is object driven and a priori information about objects in the image is also usually used.

In this thesis, the outer most pixels of edges of the detected defect were used as the pixels of the boundary of that defect so that no thinning process was required. The edges found by this boundary confirmation were one or two pixels in width. The

reason for using this strategy is based on two facts of edge detection algorithms used in the study. One is that most of the defect edges found by the three edge detectors were far from accurate in edge positions because the actual defect boundaries had big transition bands and the edge detection algorithms themselves were inaccurate. The second is that no overall and consistent tendency of over- or under-estimating the defect was found, and no a priori knowledge about the defect shapes helped, so the edge thinning was ineffective in finding exact edge pixels.

Edge following of a binary edge map in this study included two steps:

(1) delete isolated pixels by setting a threshold for the acceptable defect size (it was to be 5). If the size or the numbers of pixels of a detected object blob is less than that threshold, then delete it; and

(2) if a detected defect blob is not closed or has gaps between the boundary edges, then interpolate the missing pixels by using a cubic spline approach.

Using a cubic spline interpolation meant that the shape of the resulting object or defect is inaccurate if the missing pixels are too great in number.

3.3 REGION EXTRACTION

Region extraction techniques utilize similarity properties, so they offer good noise immunity and give closed objects. One of the main advantages of region extraction is that different image characteristics and similarity criteria can be utilized according to the image characteristics and segmentation method. This means that the region extraction algorithms are much more flexible than thresholding and edge detection algorithms in terms of application goals. Another advantage of region extraction is that use of color information is not as difficult as in thresholding. The edges of the detected regions can be easily found by differentiating the segmented image.

Problems can arise in the selection of initial regions and in selecting the merging criterion. Also, a partially processed image would not contain a few clear,

dominant regions in the image but would contain many small unmerged regions, thus requiring procedures to eliminate the unmerged small regions.

As in edge detection, region methods use local information heavily and have difficulty utilizing global information. This drawback was overcome by developing two new algorithms.

3.3.1 A Method Combining Splitting-and-Merging with Region Growing

Horowitz and Pavlidis (1974, 1976) first proposed the combination of splitting and merging to improve the performance of both methods and reduce the computation cost by use of a quadtree data structure. The criterion for merging and splitting is a predetermined error tolerance of brightness or intensity. The merging starts with a partition that fulfills the requirement that the brightness of all pixels in a region satisfy the error tolerance and continues until no two adjacent regions can be merged together while still satisfying the error tolerance. The splitting of a region into its four quadrants starts with a partition satisfying the criterion that the brightness of all pixels in that region satisfy the error tolerance. The splitting and merging processes are exclusive over a region.

The main advantage of splitting-and-merging is its flexibility because any proper criteria for merging and splitting can be utilized. The main problem of this sort of split-merge is that the edge segments formed have only two orientations, horizontal and vertical, and their positions are restricted by the borders of the quadtree nodes.

As long as the quadtree data structure is determined, the main part of splitting-and-merging is to find criterion for the splitting and merging procedures. Because of the variability both in clear wood and defects of a veneer image and between veneer images, the criterion should be individual image dependent. The global information about the whole image should be utilized in an appropriate way.

The utilization of global information of a whole image was realized by finding the tri-parameter (mean, standard deviation, and minimum value) clear wood statistics.

The splitting-and-merging criteria were proposed as a comparison of local region tri-parameter statistics with the global clear wood tri-parameter statistics.

The procedure of the algorithm has three steps:

(1) find the clear wood tri-parameter feature of each color channel in a tri-parameter color space;

(2) splitting-and-merging: starting with the whole image, compare a region with those clear wood tri-parameter features. If the difference or ratio is over certain tolerances (see Chapter 6 for details about how to determine those tolerances), then a nonuniform or defective region is claimed and the splitting process should be carried out on that region. Otherwise, the region is uniform or nondefective, and no splitting is needed but merging should be checked. If any two adjacent quadrants are uniform and meet the merge criterion, then they are merged into one. This process is repeated until no splitting-and-merging processing can be done or the size of regions is less than a predetermined value; and

(3) grouping: group all resulting regions together to form clear wood and defect using the mean values of resultant regions.

The clear wood statistics are found based on the idea of region growing, where a seed pixel is found first and then as many adjacent pixels as possible are grouped to form a region. Amadasun and King (1988) proposed this idea for locating uniform areas in an image. In their region growing, the mean feature (gray value) values of each region in an image are computed, and then the two most similar mean features are merged together to form the desired number of mean features.

The procedure of clear wood tri-parameter feature finding in the proposed algorithm included two steps:

(1) the whole image is divided into square windows or tiles (except that the right most and top most tiles may be rectangular depending on the whole image size and the tile size), a seed tile with the minimum standard deviation among all tiles is found, and then the tri-parameter

features (mean, standard deviation and minimum) of that tile are calculated, and

(2) certain numbers of tiles whose tri-parameter features are most similar to that of the seed tile are combined to obtain the clear wood tri-parameter feature. In this region-growing-like process, the tiles chosen to form the clear woGæ tri-parameter feature are not necessarily adjacent to the seed tile, which is different from most other region growing algorithms.

Images of veneer can be described in terms of statistics such as mean, variance, and minimum value. The average intensity for knots, wane, pitch pockets, and some dark pitch streaks and blue stain are smaller than those of clear wood, so that they look darker. In the tri-parameter color space, the defects' tri-parameter color values also differ from those of the clear wood. These differences over a region can be used to discern the nonuniform region (a region that may contain the defect) from the uniform region (regions contain only clear wood). The standard deviation over a defective region is commonly larger than that over a clear wood region for many images of veneer. In addition, the minimum value of each color channel over a defective region is significantly different from that of the clear wood region. Consequently, the comparison means, standard deviations, and minimums will eventually result in the separation of the defect from clear wood. The criteria for those differences in tri-parameter statistics were found heuristically.

The criteria for the splitting-and-merging are described in the following paragraphs:

(1) mean difference: assume that the pixel values of a region in each color parameter channel (i.e., R, G, or B) are normally distributed with mean m and standard deviation s. Let the value of a pixel be denoted as x_i and the number of pixels in the region as n, then the estimated value of m and s are

$$m = \frac{1}{n} \sum_{i=1}^{n} x_i$$

$$s = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(x_i - m)^2}$$

Let the subscript clear stand for the clear wood values and the subscript r stand for the current region values. Then the test statistic for the mean difference is

$$H_0: \quad m_r = m_{clear}$$

$$H_1: \quad m_r \neq m_{clear}$$

For

$$t^* = \frac{m_r - m_{clear}}{\sqrt{\frac{s_r^2}{n_r} + \frac{s_{clear}^2}{n_{clear}}}}$$

While the test statistic t^* does not exactly follow a t distribution, it will be well approximated by t if v is used as the degrees of freedom

$$v = \frac{(\frac{s_r^2}{n_r} + \frac{s_{clear}^2}{n_{clear}})^2}{\frac{(\frac{s_r^2}{n_r})^2}{n_r - 1} + \frac{(\frac{s_{clear}^2}{n_{clear}})^2}{n_{clear} - 1}}$$

If

$$|m_r - m_{clear}| > 2.33\sqrt{\frac{s_r^2}{n_r} + \frac{s_{clear}^2}{n_{clear}}}$$

then H_0 is rejected, that is, there is a significant difference between two means. Otherwise, H_0 is accepted, that is, there is no significant difference between two means at the 1% level (where t = 2.33 for the 0.01 significant level is used and the degrees of freedom v > 120 is assumed) (because the tile size is over 11 will be used),

(2) the ratio of standard deviations: the idea is borrowed from the inference about the variances of normal distributions, in which the ratio of the sample variances can be used to test whether the two variances are equal using the F test. A predetermined value c is used to test for variance equality, that is, if

$$\frac{s_r^2}{s_{clear}^2} > c$$

then the testing region is a possible nonuniform one. Otherwise, it is a uniform region, and

(3) the ratio of minimums: if

$$a_{high} < \frac{\min_r}{\min_{clear}} < a_{low}$$

then the region is considered to be a nonuniform region. Otherwise, it is considered as an uniform region, where a_{high} and a_{low} are predetermined values.

Overall, if any one or more of the above tests gives the result of a nonuniform region, then the region is claimed as a nonuniform region. On the other hand, if all three tests give the result of an uniform region, then the region can be said to be uniform and all pixels in the region are set to equal the mean value of that region.

If a region is tested as a nonuniform one, then it is split into its four quadrants and the tests are carried out again. If any two adjacent quadrants are tested as uniform regions and they also satisfy the merge criterion, then the two regions are merged together. The merge criterion is only the mean difference as described above because the uniform region only has this statistic.

In the final step of the algorithm, grouping, the criterion for testing for clear wood or defect is the function of mean values of the region, the maximum and minimum values over the original whole image and the range of the camera resolution (31 for a 5-bit camera resolution and 255 for 8-bit camera resolution). This criterion is a heuristic result. If

$$|x_r - m_{clear}| > w \frac{\max_{original} - \min_{original}}{range}$$

then the region is thought to be defective. Otherwise, it is clear, where w is the weighting factor, and x_r is used because all pixels in the resulting region are the same. The binary image is thereupon obtained with black representing the defect and white the clear wood. A diagram of the algorithm is shown in Figure 3.13.

3.3.2 A Method Combining Clustering With Region Growing

Amadasun and King (1988) proposed an approach which combines clustering with the region growing concept. Essentially, the technique involves the computation of the mean feature values of each region in an image. If the number of mean feature vectors is larger than the number of categories which are to be segmented, the two most similar mean features are considered to be from regions belonging to the same category. Then these mean feature vectors are used to classify the pixels.

The problem they mentioned is that the choice of two parameters used in the scheme, neighborhood size and uniformity criterion, is critical. The uniformity criterion and the window size determine the accuracy of the final results. Another characteristic of the algorithm is that the number of categories is specified by the user. It is a problem only if there is not any priori knowledge about an image to be worked on. In defect detection of images of veneer, only clear wood and defects are considered so that the number of categories is known.

The idea of combining clustering with region growing as described by Amadasun and King is used in the study but the criteria used for region growing and clustering are completely different and global information about the image is utilized. Here, a strategy similar to the method combining splitting-and-merging with region growing is used for finding clear wood features in the region growing.

(1) The original image is divided into square tiles (except that the right most and top most tiles may be rectangular depending on the whole image size and the tile size), and a seed tile with the minimum standard deviation and minimum mean edge strength is found. The tri-

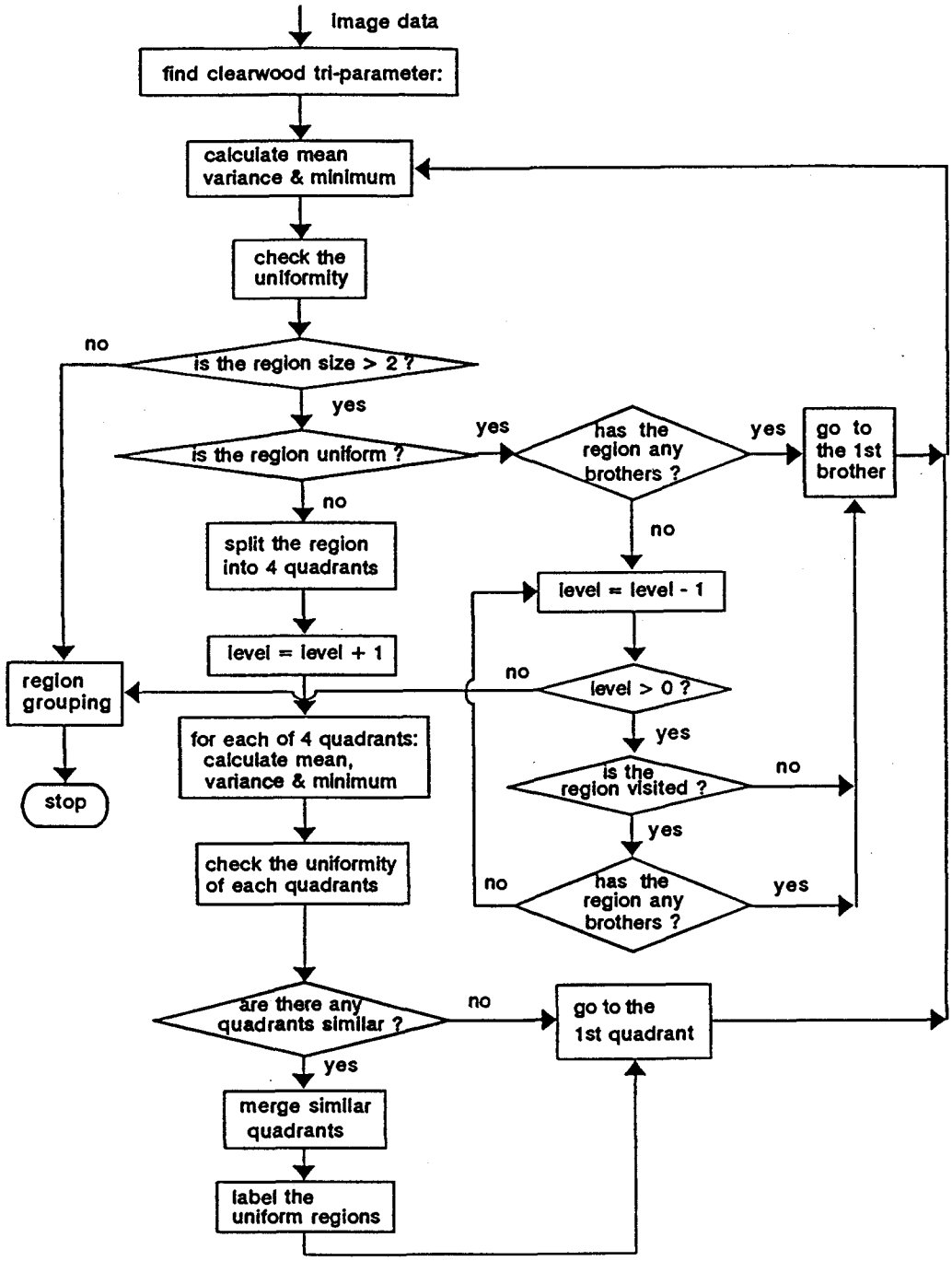

Figure 3.13. A diagram of the method combining splitting-and-merging with region growing.

parameter features (mean, standard deviation and minimum of the seed tile) are calculated; and

2) certain numbers of tiles whose tri-parameter features are most similar to the seed tile are combined to obtain the clear wood tri-parameter features.

Instead of using only the minimum standard deviation to determine the seed tile, the minimum mean edge strength is also used to determine the seed tile for the better starting decision. The edge strength map is found by the color Sobel edge detector because it performed best among Roberts, Prewitt, Kirsh and three-level gradient operators in previous investigations of veneer images. The edge strength is the maximum gradient value in the three-parameter color image and the mean edge strength is the average edge strength value of the tile of interested. The tiles used to form the clear wood feature are not to be classified further since they are already assumed to be the clear wood tiles.

In clustering, each pixel in a tile is classified as either a clear wood pixel or defect pixel by a heuristic criterion. If

$$s_r |x - m_{clear}| > w \frac{\max_{original} - \min_{original}}{range}$$

then the pixel is claimed as a defect pixel. In the criterion, s_r denotes the standard deviation of the tile, x the value of the pixel, m_{clear} the mean value of the clear wood found in region growing, w a weighting factor, $max_{original}$ the maximum value of the original whole image, $min_{original}$ the minimum value of original whole image, and range is determined by camera resolution (31 for 5-bit and 255 for 8-bit). For a tri-parameter color image, all the values become three-dimensional vectors. If the pixel value fails in meeting the criteria in any one or more color channels, then that pixel is declared as a defect pixel. Otherwise it is a clear wood pixel. The diagram of the algorithm is shown in Figure 3.14.

3.4 Summary of Algorithms Used in the Project

All algorithms discussed in Chapter 3 were applied to images of the veneer samples. These are summarized in Table 3.1.

Table 3.1. Algorithms used in the project.

Category	Algorithms
edge detection	compass edge detector entropy edge detector nonlinear Laplace edge detector
region extraction	a method combining splitting-and-merging with region growing a method combining clustering with region growing
thresholding	global Otsu method global entropy method global moment-preserving method global transition-matrix method local Otsu method local entropy method local moment-preserving method local transition-matrix method

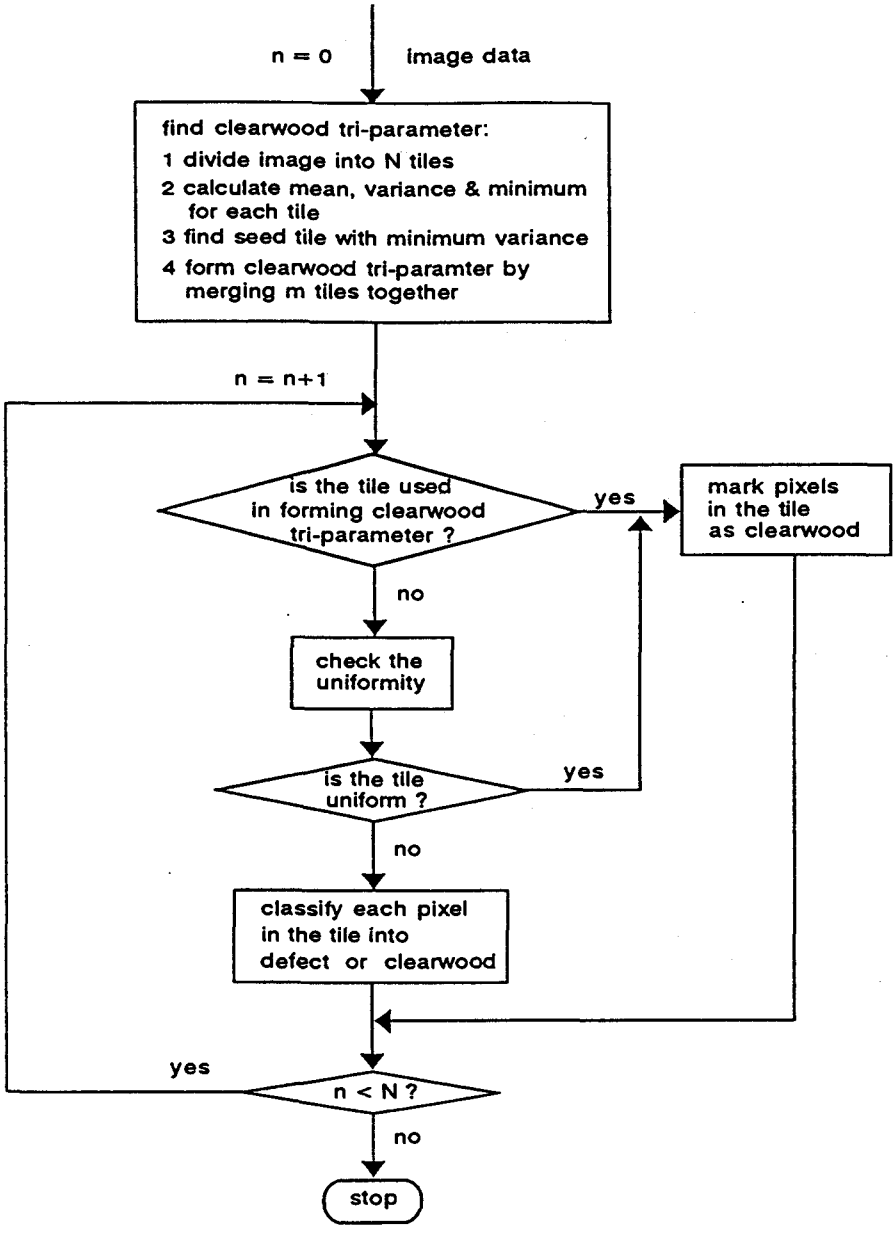

Figure 3.14. A diagram of the method combining clustering with region growing.

CHAPTER 4 ALGORITHM PERFORMANCE EVALUATION TECHNIQUES

As an important part of the project, effective and reliable performance measures had to be proposed to evaluate the algorithms used for defect detection on veneer surfaces. As mentioned in Chapter 2, a few evaluation methods have been proposed in the development of image processing techniques. Most of them deal with the performance of different edge operators and are used as quantitative comparisons of the effectiveness of various types of preprocessing operations facilitating edge detection. However, some did deal with the performance of more general segmentation algorithms.

Because many segmentation or edge detection algorithms were designed for different purposes and types of images, the previously mentioned methods are not widely used by researchers. However, they are still better than the common method of comparing performances by simple visual evaluation. To evaluate performance in this study, a few of the measures mentioned in Chapter 2 were adapted and several new measures are proposed.

4.1 EVALUATION MEASURES

Because of the application to defect scanning, the evaluation requirements for this project can be considered mainly in terms of determining the accuracy of defect detection. Therefore, measures connected with defect accuracy were emphasized.

Before discussing the evaluation measurements, two assumptions were made. First, it is assumed that the true defect edges can be accurately identified by a human expert and all true defects are closed areas. Second, it is assumed that all detected defects are closed areas (this can be made true).

There were three groups of performance measures. One was for paired defects in which the detected defect and its true defect were paired, one was for nonpaired

defect where the detected defect was not the true defect, and finally one was for all kinds (both paired and nonpaired defects). The criterion for deciding whether a detected defect corresponded to a true defect was that more than one pixel touched between the two images of the defect.

The measures will be described in detail by each of the three groups.

4.1.1 Measures for All Defects

Count percentage (M_1) is defined as the ratio of the number of detected defects to the number of true defects as given in equation 4.1,

$$M_1 = \frac{N_d}{N_t} \times 100 \qquad (4.1)$$

where N_d denotes the number of detected defects and N_t the number of true defects.

Paired percentage (M_2) is defined as the ratio of the number of detected true defects to the number of true defects as given in equation 4.2,

$$M_2 = \frac{N_{pair}}{N_t} \times 100 \qquad (4.2)$$

where N_{pair} denotes the number of paired defects.

Clear wood accuracy (M_9) is defined as the ratio of the number of true clear wood pixels detected to the total number of clear wood pixels as given in equation 4.3,

$$M_9 = \frac{P^d_{clear}}{P^t_{clear}} \times 100 \qquad (4.3)$$

where P^d_{clear} stands for detected true clear wood pixels and P^t_{clear} stands for the number of true clear wood pixels.

Defect accuracy (M_{10}) is defined as the ratio of the number of detected true defect pixels to the total number of defect pixels as given in equation 4.4,

$$M_{10} = \frac{P^{d}_{defect}}{P^{t}_{defect}} \times 100 \qquad (4.4)$$

where P^{d}_{defect} stands for number of detected true defect pixels and P^{t}_{defect} stands for number of the true defect pixels.

4.1.2 Measures for Paired Defects

The following four measures were proposed only for paired defects. Wrong percentage (M_3) is defined as the ratio of the number of clear wood pixels wrongly or incorrectly marked as defect pixels versus the total number of true defect pixels as given in equation 4.5,

$$M_3 = \frac{P^{d}_{c}}{P^{t}_{d}} \times 100 \qquad (4.5)$$

where P^{d}_{c} stands for number of true clear wood pixels detected as defect pixels and P^{t}_{d} stands for the true defect pixels.

Missed percentage (M_4) is defined as the ratio of the number of true defect pixels marked as clear wood pixels (missed defect pixels) to the total number of true defect pixels as given in equation 4.6,

$$M_4 = \frac{P^{c}_{d}}{P^{t}_{d}} \times 100 \qquad (4.6)$$

where P^{c}_{d} stands for number of true defect pixels detected as clear wood pixels.

Mean centroid difference (M_5) is defined as the mean value of the centroid differences between detected defect and true defect pairs as given in equation 4.7,

$$C = \frac{\sum_{i=1}^{N_{pair}} \sqrt{(x_{d_i} - x_{t_i})^2 + (y_{d_i} - y_{t_i})^2}}{N_{pair}} \qquad (4.7)$$

where superscript d_i and t_i stand for the ith detected defect and true defect, respectively, x and y represent the coordinates of the centroid, and N_{pair} stands for the number of defect pairs.

Mean ratio difference (M_6) is defined as the mean value of the differences between the detected defect and the true defect pairs in the ratios of the boundary lengths to the area enclosed as shown in equation 4.8 and 4.9.

$$M_6 = \frac{\sum_{i=1}^{N_{pair}} (R_{d_i} - R_{t_i})}{N_{pair}} \qquad (4.8)$$

where

$$R = \frac{boundary\ \ length}{area\ \ enclosed} \qquad (4.9)$$

4.1.3 Measures for Nonpaired Defects

The following two measures were proposed only for the nonpaired defect, that is, the detected defect was not the true defect at all so the true defect was completely missed.

Nonpaired wrong percentage (M_7) is defined as the ratio of the number of true clear wood pixels wrongly marked as defect pixels to the total number of true defect pixels as shown in equation 4.10.

$$M_7 = \frac{P_c^d}{P_d^t} \times 100 \qquad (4.10)$$

where P^d_c stands for number of true clear wood pixels detected as defect pixels and P^t_d stands for the true defect pixels.

Nonpaired missed percentage (M_8) is defined as the ratio of the number of true defect pixels marked as clear wood pixels to the total number of true defect pixels as shown in equation 4.11.

$$M_8 = \frac{P_d^c}{P_d^t} \times 100 \qquad (4.11)$$

where P^c_d stands for number of true defect pixels detected as clear wood pixels.

4.1.4 The Functions of the Performance Measures

A number of the measures are designed to quantify the overall algorithm performance (count percentage, paired percentage, clear wood accuracy, and defect accuracy). In other words, these measures indicate whether the algorithm has a tendency to over- or under-estimate the defects. The measures of clear wood and defect accuracy are the same as those used by Forrer et al. (1989).

For more detailed information, such as size, shape, and location, about the detected defects, four measures for paired defects were introduced. The wrong percentage and missed percentage tell the percentages of two types of pixels wrongly marked on the basis of true defect pixels. As shown in Figure 4.1, region 1 represents the missed defect portion and region 2 the incorrectly detected defect portion. They can be roughly explained as the type I and type II errors in terms of statistics. The total number of true defect pixels was used as the denominator for both percentages instead of the total true clear wood pixels for the missed percentage because the total number of clear wood pixels in a typical veneer image are so large in comparison that the ratio would be too small to statistically identify any differences.

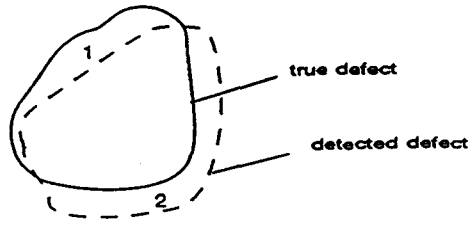

Figure 4.1. An example of the shape of a detected defect versus that for the real defect.

The mean centroid difference is used to measure the location accuracy of detected defects as well as certain types of shape differences, that is, how good the detected defect is overlapped with the true defect and how good the shape of the detected defect coincides with the true defect in such cases where the paired missed percentage and wrong percentage fail to tell the differences. This is shown in Figure 4.2. The values of missed percentage (M_4) for both cases 1 and 2 in Figure 4.2 are similar but the value of the mean centroid difference (M_5) in case 2 will be larger than that in case 1. Similarly, the values of wrong percentage (M_3) for cases 3 and 4 in Figure 4.2 are similar but the value of the mean centroid difference (M_5) in case 4 will be much larger than that in case 3. Therefore, the centroid difference clearly represents the goodness-of-fit in terms of overlapping and shape coincidence between the detected defect and the true defect.

The ratio of boundary length to enclosed area of an object was proposed by Bryant and Bryant (1989) as a measure of smoothness of an object boundary with smaller values meaning smoother boundaries. As in Figure 4.3 where the three measures, wrong percentage (M_3), missed percentage (M_4), and mean centroid difference (M_5) are very close to each other in both examples, the two results are noticeably different. Obviously, the resulting curve in case 1 is smoother than that in case 2. The value of the mean ratio difference (M_6) for case 1 is therefore, larger since the boundary length is shorter than that in case 2 but with a similar area. It should be pointed out that the mean ratio difference is meaningful only if the shape of the detected defect is similar to that of the true one.

By combining the mean centroid difference and mean ratio difference with the missed and wrong percentages, information about differences in shape, location, and boundary smoothness of detected defects can be obtained and therefore, the differences in performance between different algorithms can be compared.

Finally, two measures M_7 and M_8 were proposed for the nonpaired defects. Besides the count percentage and paired percentage, these two measures can provide more detailed information about the size of the incorrectly detected detects and the size of true defects missed. As mentioned earlier, the measures of clear wood and

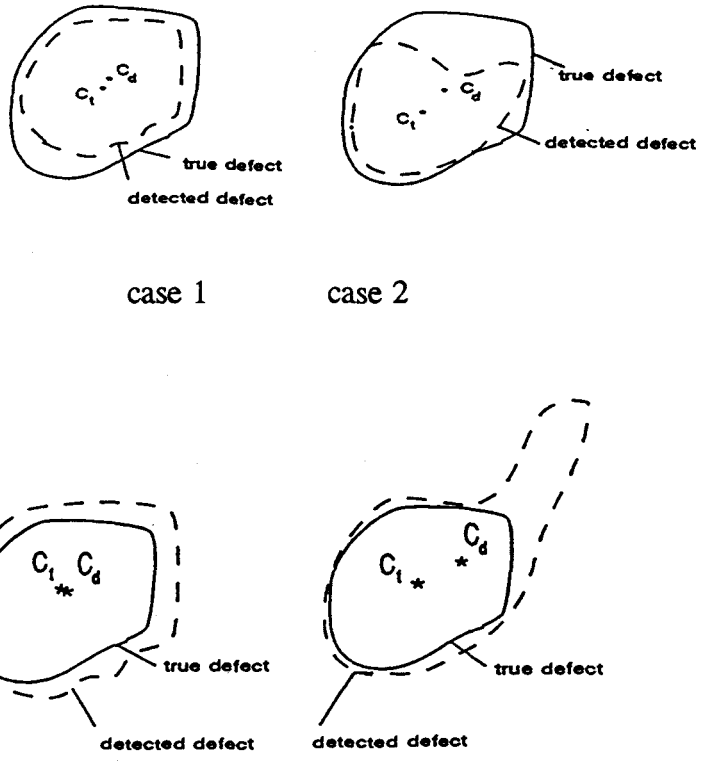

Figure 4.2. Four resulting cases of detected defect shapes versus the real defect shapes.

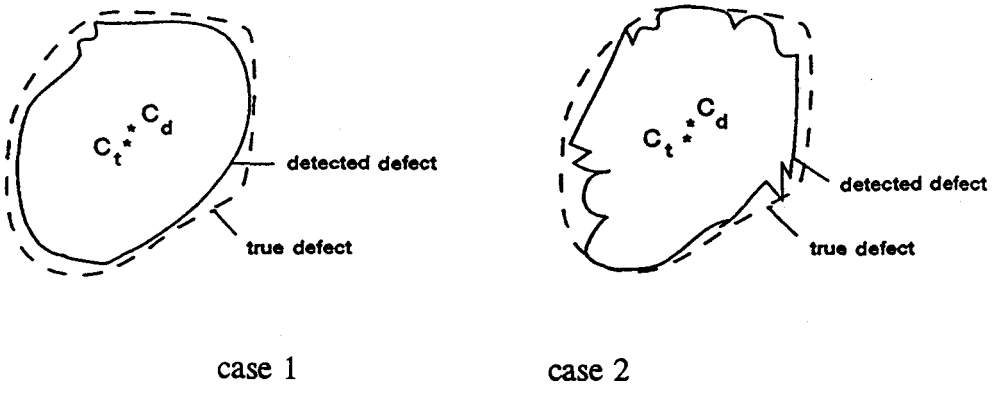

Figure 4.3. The boundaries of two resulting detected defects versus the true defect shape.

defect accuracy provide some information about incorrectly detected defects and missed defects. However, if the size of an incorrectly detected defect is not large enough, the clear wood accuracy would not clearly show this because of the large value of its denominator. The nonpaired missed defect pixels can not be distinguished by the defect and clear wood accuracies (M_9 and M_{10}) because the two values include both completely missed defects (nonpaired) and missed portions in the paired defects.

4.2 JUSTIFICATION OF THE PERFORMANCE MEASURES

To test the effectiveness and reliability of the performance measures, ten veneer images with holes and tight knots as well as five artificial defect images were used to try to produce a variety of results to be evaluated using the ten measures. For example, the images contained very dark small and large defect holes with both comparatively uniform and nonuniform backgrounds. The artificial images were obtained by changing some pixel values in real veneer images to fit the needs of the testing process. The Sobel edge operator, Roberts edge operator, and the Otsu thresholding algorithm were applied to the 15 images.

All ten measures M_1, M_2, M_3, M_4, M_5, M_6, M_7, M_8, M_9, and M_{10} (as shown in Table 4.1) were calculated based on each image pair (the master image was classified by a human expert and the test image is the resulting image obtained by applying an algorithm to the original image). Results showed that these ten measures were good indicators for comparing algorithm performance. This conclusion is based on the results obtained with the measures as well as the visual appearance.

Since the algorithms used in the test all located the defects correctly, the measure of mean centroid difference (M_5) values were all very small and the nonpaired wrong percentage and missed percentage (M_7 and M_8) were deemed to be redundant because the count percentage, paired percentage, clear wood accuracy and defect accuracy gave enough information about the results. However, in the case where some algorithms inaccurately located defects and some cases where the

nonpaired measures supplied better information, the three measures M_5, M_7, and M_8 were still kept for the statistical analysis. It was determined that all ten measures would be used in the experiment unless some of them were found to be improper or redundant so that they could be laid aside.

Table 4.1. Performance measures used in the project

Measure	Name of the Measure
M_1	Count percentage
M_2	Paired percentage
M_3	Paired wrong percentage
M_4	Paired missed percentage
M_5	Mean centroid difference
M_6	Mean ratio difference
M_7	Nonpaired wrong percentage
M_8	Nonpaired missed percentage
M_9	Clear wood accuracy
M_{10}	Defect accuracy

CHAPTER 5 EXPERIMENTAL PROCEDURES

In this chapter, details about the experimental procedures will be discussed, which includes the hardware setup for each sample image taken, software used in the experiment, selection of veneer samples, and the statistical analysis procedure utilized.

5.1 HARDWARE SETUP AND SOFTWARE

The optical video imaging system available in the Oregon State University Forest Products Department was used for this project. The hardware setup consisted of four parts: camera, digitizer, lighting, and microcomputer. A three-tube JVC model BY-110 color camera was used. It was equipped with a prism in the optical path which separates the optical image into the three primary color components, red, green, and blue. Each of the three images was sampled by a separate tube to produce an analog signal. The lens used was a JVC HZ-110 zoom lens with focal distance of 7 mm to 70 mm and the aperture was set to be 4.5. The camera was placed directly overhead of the sample at a sample-to-lens distance of approximately four feet. The camera's gamma correction was on at 0.45 and the contour correction and auto-iris were also on. The camera was black balanced, white balanced, and then registered.

The camera's analog R-G-B signals are fed to an AT&T Truevision Advanced Raster Graphics Adapter (TARGA-32), which is an image-capture board plugged into a bus-expansion slot in an IBM-PC clone microcomputer. The TARGA board converts each of the three R, G, and B analog image signals to 8-bit digital image signals for each pixel in the 200 by 256 array. The converted R-G-B readings are then packed into a 32-bit value and stored in the proper row/column position in a frame buffer. The color images, thereafter, can be manipulated and displayed on a color monitor.

All the algorithms used in the study were coded in the "C" programming language using the Microsoft "C" compiler. Basic utility software supplied with the

TARGA board and extensive image analysis software developed by Department of Forest Products personnel were used for image acquisition, preprocessing, color transformation, and enhancement.

Lighting was provided by two centered overhead quartz-halogen (tungsten) lights equipped with a light scattering diffuser on each side of the item to be imaged as well as two fluorescent lights. The veneer sample was placed on a horizontal surface within the field-of-view of the camera. No other light was allowed.

5.2 ALGORITHMS UTILIZED IN THE PROJECT

All algorithms discussed in Chapter 3 were applied to the images of the veneer samples. Those algorithms are summarized in Table 5.1.

5.3 SAMPLES

About fifty sheets of Douglas-fir [Pseudotsuga menziesii] veneer were used as samples. The samples were chosen to represent the degrees of variation found in seven types of features, (tight knot, loose knot, wane, pitch pocket, pitch streak, blue stain, and clear wood) as well as surface properties (distribution, intensity, color, and contrast in both defect and clear wood (latewood and earlywood)). No samples with the same defect type came from the same sheet of veneer but some samples with different defect types came from the same sheet.

This project primarily considered only single-defect images. However, because of selection difficulties, nine images had two or more defects. Two images had two tight knots, one image had two pitch pockets, one image had two streaks of blue stain, four images had three streaks of blue stain, and one image had four streaks of blue stain. Using grading rules published by the American Plywood Association (APA) and the Western Wood Products Association (WWPA), the replicates for each of the defect types were chosen as shown in Table 5.2 (clear wood excluded). To

Table 5.1. Algorithms used in the project.

Category	Algorithms
edge detection	compass edge detector entropy edge detector nonlinear Laplace edge detector (with window size of 3) nonlinear Laplace edge detector (with window size of 5)
region extraction	combined splitting-and-merging with region growing combined clustering with region growing
thresholding	global Otsu method global entropy method global moment-preserving method global transition-matrix method local Otsu method local entropy method local moment-preserving method local transition-matrix method

avoid introducing experimental errors, all seventy images were taken on the same day under the same lighting and video camera settings.

All sample images were RGB color images with 8-bit camera resolution (each R, G, and B value could range from 0 to 255). To test the effect of camera resolution, the corresponding RGB color images with 5-bit camera resolution (pixel values range from 0 to 31) were obtained by truncating the 8-bit image values (256 levels) to 5-bit levels (32 values). Previous research indicated that the 5-bit images produced by this conversion process contained less noise than those acquired by an actual 5-bit frame grabber. However, this was not deemed to be significant to the project results.

The spatial resolution of the sample images (number of pixels required to represent an actual real world length) was chosen to be 15 pixels per inch. This

Table 5.2. Number of samples in different grades.

Defect Type	Number of Samples
tight knot	3 small 3 medium 4 large
loose knot	3 small 3 medium 4 large
pitch pocket	4 small 4 medium 2 large
pitch streak	3 small 3 medium 4 large
blue stain	3 light 3 medium 4 heavy
wane	5 small 5 large

decision of spatial resolution was based on previous research by Forest Products Department personnel. That research showed that 15 pixels per inch were optimal on the basis of offering enough detailed information for defect detection on veneer surfaces while minimizing the number of pixels and processing time.

To exclude defects so as to consider only single-defect images whenever possible, the center portion of each sample image was used instead of the full image (200 rows by 256 columns). The window used for obtaining the center portion was set at bottom left column 51, bottom left row 36, top right column 201, and top right row 155. This was done so that the actual field-of-view would match that seen at 25 pixels per inch, which was a spatial resolution used in several other studies.

5.4 DATA ANALYSIS

For each sample and each algorithm, the performance measures were calculated according to the formulas described in the Chapter 4. To determine performance differences, a multifactor factorial analysis of variance (ANOVA) and Fisher's least-significant-difference (FLSD) test for multiple comparisons were used. The FLSD was chosen because it is a very effective test for detecting true differences in means when applied after an F-test if the analysis of variance is significant at the 5% level (Montgomery, 1988). The statistical package SAS, version 6.03, (SAS/STAT user's guide, 1988) was used to perform the statistical test.

After analyzing the output data, it was noticed that measures M_7 and M_8 for nonpaired defects do not offer much additional information regarding algorithm performance since they are very small for many of the combinations of experimental treatments. Also, the measure of clear wood accuracy (M_9) and the measure of count percentage (M_1) offer similar information. The data also showed that measure M_6, mean ratio difference, had both positive and negative values since some detected defects have smoother boundaries than those found in the actual images. The magnitudes of those positive and negative values for some veneer surface features were very large; for instance, the magnitudes of M_6 for most samples of pitch streak and blue stain as well as some wane were over 50.0. It could be easily seen that the shape of the detected defect was quite different from the actual shape when the magnitude of M_6 was over 50.0. In such cases, the comparison between values of M_6 was meaningless because of the large variability in the smoothness of the actual defect boundaries. Since comparisons of M_6 were meaningless for more than one third of the data, it was removed from the statistical analysis. Consequently, measures M_6, M_7, and M_8 were not considered in the statistical analysis.

5.5 STATISTICAL ANALYSIS PROCEDURE

In this experiment, the factors used for the statistical analysis were unbalanced. For instance, the single color channels R, G, and B of the RGB color space and L of the Lab color space were used in the thresholding algorithms, while the RGB and Lab tri-parameter color spaces were used in the edge detection and region extraction algorithms. In addition, the Lab color scale could only be utilized in the 8-bit camera resolution because the truncation caused a significant loss of information when the RGB color images were transformed into the Lab color space. Therefore, output data were separated into four subsets to make the factors balanced:

(1) thresholding algorithms under both 5-bit and 8-bit camera resolution with single color channels R, G, and B, and gray-scale for all seven image features (blue stain, clear wood, loose knot, pitch pocket, pitch streak, tight knot, and wane);

(2) thresholding algorithms with only 8-bit camera resolution with single color channels R, G, B, gray scale, and L for all seven image features;

(3) edge and region algorithms under both 5-bit and 8-bit camera resolution with the RGB color space and gray-scale for all seven image features;

(4) edge and region algorithms under only 8-bit camera resolution with the RGB color space, gray-scale, and the Lab color space for all seven image features.

Two subsets under the 8-bit resolution had three factors (algorithm, color, and feature) while the other two subsets under both 5-bit and 8-bit resolutions had four factors (algorithm, resolution, color, and feature). The purpose of the statistical analysis for data subsets using the 8-bit resolution was to discover performance differences between the color spaces RGB and Lab for edge and region algorithms and differences between L and R, G, or B for thresholding algorithms.

For each of the seven performance measures, an analysis was conducted basically following the steps for a four-factor experiment (the three factor experiment follows similar steps):

(1) Obtain a four-factor ANOVA table, its residual plot, and normal probability plot;

(2) Check the plots to see if any assumptions of the ANOVA are violated. If a transformation is required, try one of the transformation methods and go back to step (1). This step is repeated until the most appropriate transformation method isfound so that the best possible model is obtained.

(3) Check for interactions. If there are no interactions, summarize the main effects. If the four-way interaction is significant, none of the factors is acting independently. In such a case, the results should be summarized in a four-way table of means in four-factor combinations along with their standard errors or reported in a one-way ANOVA by fixing levels of other three factors. If the four-way interaction is not significant, check the three-way interaction. If the three-way interaction is significant, report the results using a one-way ANOVA with fixed levels of the other two factors. If the three-way interaction is not significant, check the two-way interaction, and go to the next step; and

(4) Analyze the interactive factors. Use the FLSD multiple comparison procedure to analyze the effect of one of those two factors at each fixed level of the other factor.

5.6 ASSUMPTION CHECKING

The error terms in the ANOVA test are assumed to be independent and normally distributed with constant variance. In general, moderate departures from normality are of little concern when using a fixed effect ANOVA. Since the F-test is only slightly affected, the ANOVA (and related procedures such as multiple comparisons) is robust to the normality assumption (Montgomery, 1988).

If the assumption of homogeneity of variances is violated, the F test is only slightly affected in the balanced, fixed effects model. However, if sample sizes are unequal, the F-tests may be affected considerably.

When the validity of the assumptions made in an ANOVA have been tested and one or more assumptions are shown to be violated, an appropriate transformation is usually necessary. The transformation is usually helpful for most applications (Montgomery, 1988).

In the statistical analysis, different data transformations were used for different measures and data subsets to meet the assumptions. Although many transformations were tried, for some of the measures and data sets, the assumptions were still not completely satisfied. In such cases, common sense was considered; that is, if the results coincided with the expectation, the results were still accepted.

Table 5.3 is an ANOVA table for the three-factor fixed effects model (factors A, B, and C with degrees of freedom a, b, and c, respectively). The test statistics for each main effect and interaction are constructed by dividing the corresponding mean square for the effect or interaction by the mean square error. The number of degrees of freedom for any main effect is the number of levels of the factor minus one, and the number of degrees of freedom for an interaction is the product of the number of degrees of freedom associated with the individual components of the interaction. If the F_0 is greater than the critical F value (it is determined by the significance level, degrees of freedom of the denominator and numerator), then the null hypothesis of no difference in treatment means or no interaction is rejected and significant differences are reported.

Table 5.3. The ANOVA table for the three-factor fixed effect model.

Source of Variation	Sum of Squares	Degrees of Freedom	Mean Square	Expected Mean Squares	F_0
A	SS_A	$a - 1$	MS_A	$\sigma^2 + \frac{bcn \sum \tau_i^2}{a - 1}$	$F_0 = \frac{MS_A}{MS_E}$
B	SS_B	$b - 1$	MS_B	$\sigma^2 + \frac{acn \sum \beta_j^2}{b - 1}$	$F_0 = \frac{MS_B}{MS_E}$
C	SS_C	$c - 1$	MS_C	$\sigma^2 + \frac{abn \sum \gamma_k^2}{c - 1}$	$F_0 = \frac{MS_C}{MS_E}$
AB	SS_{AB}	$(a - 1)(b - 1)$	MS_{AB}	$\sigma^2 + \frac{cn \sum\sum (\tau\beta)_{ij}^2}{(a - 1)(b - 1)}$	$F_0 = \frac{MS_{AB}}{MS_E}$
AC	SS_{AC}	$(a - 1)(c - 1)$	MS_{AC}	$\sigma^2 + \frac{bn \sum\sum (\tau\gamma)_{ik}^2}{(a - 1)(c - 1)}$	$F_0 = \frac{MS_{AC}}{MS_E}$
BC	SS_{BC}	$(b - 1)(c - 1)$	MS_{BC}	$\sigma^2 + \frac{an \sum\sum (\beta\gamma)_{jk}^2}{(b - 1)(c - 1)}$	$F_0 = \frac{MS_{BC}}{MS_E}$
ABC	SS_{ABC}	$(a - 1)(b - 1)(c - 1)$	MS_{ABC}	$\sigma^2 + \frac{n \sum\sum\sum (\tau\beta\gamma)_{ijk}^2}{(a - 1)(b - 1)(c - 1)}$	$F_0 = \frac{MS_{ABC}}{MS_E}$
Error	SS_E	$abc(n - 1)$	MS_E	σ^2	
Total	SS_T	$abcn - 1$			

CHAPTER 6 EXPERIMENTAL RESULTS

In this chapter, the experimental results for seven comparison tests will be discussed. The seven comparison tests are:

(1) comparison of veneer surface features;

(2) comparison of camera resolutions;

(3) comparison of color spaces;

(4) comparison of edge detection algorithms;

(5) comparison of global and local thresholding;

(6) comparison of thresholding algorithms; and

(7) comparison of seven prechosen algorithms.

6.1 PARAMETER SETTINGS OF ALGORITHMS

As discussed in Chapter 3, each algorithm has some parameters that are set heuristically. They are affected by color space (RGB, Lab or gray-scale), camera resolution, (5-bit or 8-bit), and image characteristics. All parameters were set by trial and error so that the optimal ones were obtained only for the Douglas-fir (Pseudotsuga merziesii) image data set taken under the conditions mentioned in Chapter 5. This section discusses the sensitivity of those optimal parameters.

6.1.1 Thresholding (Global and Local)

Because preliminary research indicated that thresholding algorithms did not perform well for many sample images when using the a or b components of the Lab color space, only L was utilized.

The tile size for all four thresholding algorithms (both global and local versions) was 30 pixels square. None of the thresholding algorithms were sensitive

to the tile size. Tile sizes of 20 to 40 pixels square resulted in few differences (for instance, 1% or 2% in the clear wood or defect accuracies). A tile size of 30 seemed to be the most acceptable tile size in the RGB color space with 8-bit camera resolution. For better consistency and easier comparison, this tile size was also used in the Lab color space and 5-bit camera resolution.

Both global and local thresholding algorithms are sensitive to the percentage of the upper-tail edge strength of a whole image, because this value decides whether or not a tile is detected as a defective tile. The percentage of the upper-tail edge strength should be different between camera resolutions, color components of the RGB color space or L of the Lab color space, and the image characteristics. However, the different percentage of upper-tail in the edge strength for different image features is not practical because the information about defect type is generally not known a priori. Therefore, the values were chosen by a tradeoff among seven image features (loose knot, tight knot, pitch pocket, pitch streak, wane, clear wood, and blue stain) so that they work well with all but may not be the best for any particular one. For simplicity, they were set equally for the three color components of the RGB color space and gray-scale. A 0.5 difference in these values would cause significant differences to be indicated. The percentages of upper-tail in edge strength for different combinations of algorithms and color components of color spaces were set as in Table 6.1.

6.1.2 Edge Detection

For the entropy edge detector, the weighting factor was set to be 5.0 times the maximum value of image depth (31 for 5-bit camera resolution and 255 for 8-bit camera resolution). The entropy edge detector is not sensitive to this weighting factor and a value in the range of 4.0 to 6.0 would not make more than a 2% defect or clear wood accuracy difference.

Table 6.1. The percentage of upper-tail in edge strength for thresholding algorithms.

	5-bit (gray, R, G, or B)	8-bit (gray, R, G, or B)	8-bit (L)
Otsu	2.0	1.0	1.0
entropy	5.0	3.5	6.0
moment-preserving	2.5	3.5	5.0
transition-matrix	2.0	1.25	5.0

Edge detection is sensitive to the threshold value because the threshold determines whether or not a pixel is accepted as an edge pixel. Threshold values are affected by camera resolution, color channels of different color spaces, and image characteristics. Since the image features are usually not known a priori, the thresholds were chosen by a tradeoff among seven image features (loose knot, tight knot, pitch pocket, pitch streak, wane, clear wood, and blue stain) so that they perform well are accepted for all but not necessarily the best for any particular one. A 0.2 difference in these values would cause significant

6.1.3 A Method Combining Splitting-and-Merging with Region Growing

A 12 pixels square tile size was used for finding tri-parameter clear wood features. The algorithm is not sensitive to tile size because sizes of 10 to 16 pixels square would not cause more than a 2% difference in clear wood or defect accuracy). However, a size of 12 was the best. The parameters for testing the uniformity of a region were set such that the ratio of standard deviations was 1.25, the lower bound of the ratio of minimum values was 0.7 and the upper bound was 1.2. The algorithm is sensitive to these three critical values (a 0.2 change would cause significant

Table 6.2. Thresholds for edge detection algorithms.

	5-bit RGB	5-bit gray	8-bit RGB	8-bit gray	8-bit Lab
compass	8	6	50	35	30
entropy	3	3	15	8	2
Laplace (3 x 3)	3	3	20	10	10
Laplace (5 x 5)	3	2	20	15	15

differences), and these values vary with different image features. The weighting factors for determination of whether a pixel was clear wood or a defect pixel were 6.25 for the 5-bit camera resolution with both gray-scale and RGB color spaces, 55 for the 8-bit camera resolution with both gray-scale and RGB color spaces, and 50 for the 8-bit resolution and Lab color space. The algorithm is not sensitive to the values of the weighting factors at the 8-bit resolution but it is sensitive at the 5-bit camera resolution. This is probably due to the fact that the number of levels for 5-bit resolution is much less than that for 8-bit resolution.

6.1.4 A Method Combining Clustering with Region Growing

The tile size for finding clear wood features was 12 pixels square for both 5-bit and 8-bit camera resolutions as well as gray-scale, RGB and Lab color spaces. The algorithm is not sensitive to the tile size. The weighting factors for testing whether a pixel was clear wood or a defect were 6.7 for the 5-bit camera resolution in both the gray-scale and RGB color spaces, 400 for the 8-bit resolution in both the gray-scale and RGB color spaces, and 180 for the 8-bit resolution in the Lab color space. The algorithm is not sensitive to the value of the weighting factors for the 8-bit camera resolution but it is sensitive at the 5-bit camera resolution.

6.2 STATISTICAL ANALYSIS OF PERFORMANCE MEASURES

The statistical analysis procedure was already discussed in Chapter 5. Additionally, one idea was used in the test procedure to make the data analysis easier. This idea was to try to remove one factor from the four interacting factors by breaking down the data set into separate features. For instance, if the data set had no four or three factor interaction but did have two factor interactions such as A&R, A&C, R&C, R&F, C&F, and A&F (A&R means algorithm and resolution are interacted, etc.), then a one-way ANOVA could be utilized. However, if features were separated into two subsets by trial and error (for instance, clear wood, loose knot, pitch pocket, pitch streak, tight knot, and wane as one set and blue stain as another set), then one factor can be removed from the paired interactions. For example, for the first subset the interacted pairs become AR, RF, AF, and CF so that in the test about resolution, the color factor is not interacted with resolution. Using knowledge obtained from previous research and during the course of this study, it was known that defect detection of some features was not affected significantly by color. For instance, the defect detection of tight knots, loose knots, and pitch pockets would not be affected significantly by color, but the defect detection of blue stain and pitch streaks would be significantly affected. Therefore, the break-down of the data was not difficult.

6.3 EXPERIMENTAL RESULTS

Before discussing any results and conclusions, the following points need to be mentioned:

1. All statistical tests used the 5% significance level;
2. The abbreviations of veneer features are:

   BS (blue stain);

   CL (clear wood);

   LK (loose knot);

PP (pitch pocket);

PS (pitch streak);

TK (tight knot);

WA (wane);

3. The abbreviation of algorithms are:

CO: Compass edge detection;

ET: Entropy edge detection;

L3: Nonlinear Laplace edge detection with a window size of 3;

L5: Nonlinear Laplace edge detection with a window size of 5;

RC: Algorithm combining clustering with region growing;

SM: Algorithm combining splitting-and-merging with region growing;

CET: Global entropy thresholding;

CMP: Global moment-preserving thresholding;

COT: Global Otsu thresholding;

CTR: Global transition-matrix thresholding;

LET: Local entropy thresholding;

LMP: Local moment-preserving thresholding;

LOT: Local Otsu thresholding;

LTR: Local transition-matrix thresholding;

4. The seven performance measures are:

M_1: the count percentage (the closer to 100%, the better, except for the feature clear wood where smaller percentages are better);

M_2: paired percentage (the closer to 100%, the better);

M_3: wrong percentage (the smaller, the better);

M_4: missed percentage (the smaller, the better);

M_5: mean centroid difference (the smaller, the better);

M_9: clear wood accuracy (the larger, the better);

M_{10}: defect accuracy (the larger, the better); and

5. Rules-of-thumb:

During the course of this study, it became clear that some measures would not show significant differences visually even though the statistical tests might show significant differences at the 5% level. By experience, the following rules-of-thumb were made to indicate no visual and practical difference even though the statistical test might indicate a significant difference:

(1) less than 10% difference in M_2, M_3 and M_4,

(2) less than 5% difference in M_9 and M_{10},

(3) less than 30% difference in M_1,

(4) less than 0.5 pixel difference in M_5.

The significant differences in this data analysis will not only use the statistical test results but also these rules-of-thumb. This means that a significant difference will be claimed only if the statistical test shows a significant difference at the 5% level and the difference meets the appropriate rule-of-thumb.

Table 6.3 is an example of the SAS output of the wrong percentage (M_3) for the four thresholding algorithms; the corresponding residual plots are shown in Figure 6.1. After trying different data transformation methods, the logarithm was used even though the residual plot was still not very satisfactory. The normal probability plot as well as histogram and box-and-whisker plots showed that the assumption of normality was met but the plot of residuals versus fitted values showed that the assumption of similar variance was not met very well. Since no data transformations could do better and the violation was not too severe, the four-factor factorial ANOVA was still accepted. As shown in Table 6.3, the number of observations in data set equals 4480 but only 2698 observations can be used due to missing values. This missing data (for the measures wrong percentage (M_3), missed percentage (M_4), and mean centroid difference (M_5)) was because those measures were for paired defects if an algorithm did not find a defect touching the true one, these measures were not available and set as missing. The Type III hypothesis test was used because it is not a function of cell counts. When the data are balanced or each cell has the same

Table 6.3. The ANOVA table for the four-factor analysis of the wrong percentage (M_3) for thresholding algorithms.

General Linear Models Procedure
Class Level Information

Class	Levels	Values
ALGO	8	CET CMP COT CTR LET LMP LOT LTR
RESO	2	5-BIT 8-BIT
COLOR	4	B G R Y
FEAT	7	BS CL LK PP PS TK WA

Number of observations in data set = 4480

NOTE: Due to missing values, only 2698 observations can be used in this analysis.

General Linear Models Procedure

Dependent Variable: M3

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	380	4489.9781376	11.8157319	4.55	0.0001
Error	2317	6017.6791206	2.5971856		
Corrected Total	2697	10507.6572582			

R-Square	C.V.	Root MSE	M3 Mean
0.427305	69.27945	1.6115786	2.3262001

Source	DF	Type III SS	Mean Square	F Value	Pr > F
ALGO	7	1074.0243299	153.4320471	59.08	0.0001
RESO	1	259.2352496	259.2352496	99.81	0.0001
ALGO*RESO	7	185.5327077	26.5046725	10.21	0.0001
COLOR	3	47.5126643	15.8375548	6.10	0.0004
ALGO*COLOR	21	30.0758872	1.4321851	0.55	0.9498
RESO*COLOR	3	7.0983380	2.3661127	0.91	0.4348
ALGO*RESO*COLOR	21	23.9177658	1.1389412	0.44	0.9873
FEAT	5	1245.9336377	249.1867275	95.94	0.0001
ALGO*FEAT	35	460.8181010	13.1662315	5.07	0.0001
RESO*FEAT	5	47.6575954	9.5315191	3.67	0.0026
ALGO*RESO*FEAT	35	95.0627176	2.7160776	1.05	0.3956
COLOR*FEAT	15	40.8294965	2.7219664	1.05	0.4015
ALGO*COLOR*FEAT	105	91.0822571	0.8674501	0.33	1.0000
RESO*COLOR*FEAT	15	15.6003456	1.0400230	0.40	0.9795
ALGO*RESO*COLOR*FEAT	102	92.1398153	0.9033315	0.35	1.0000

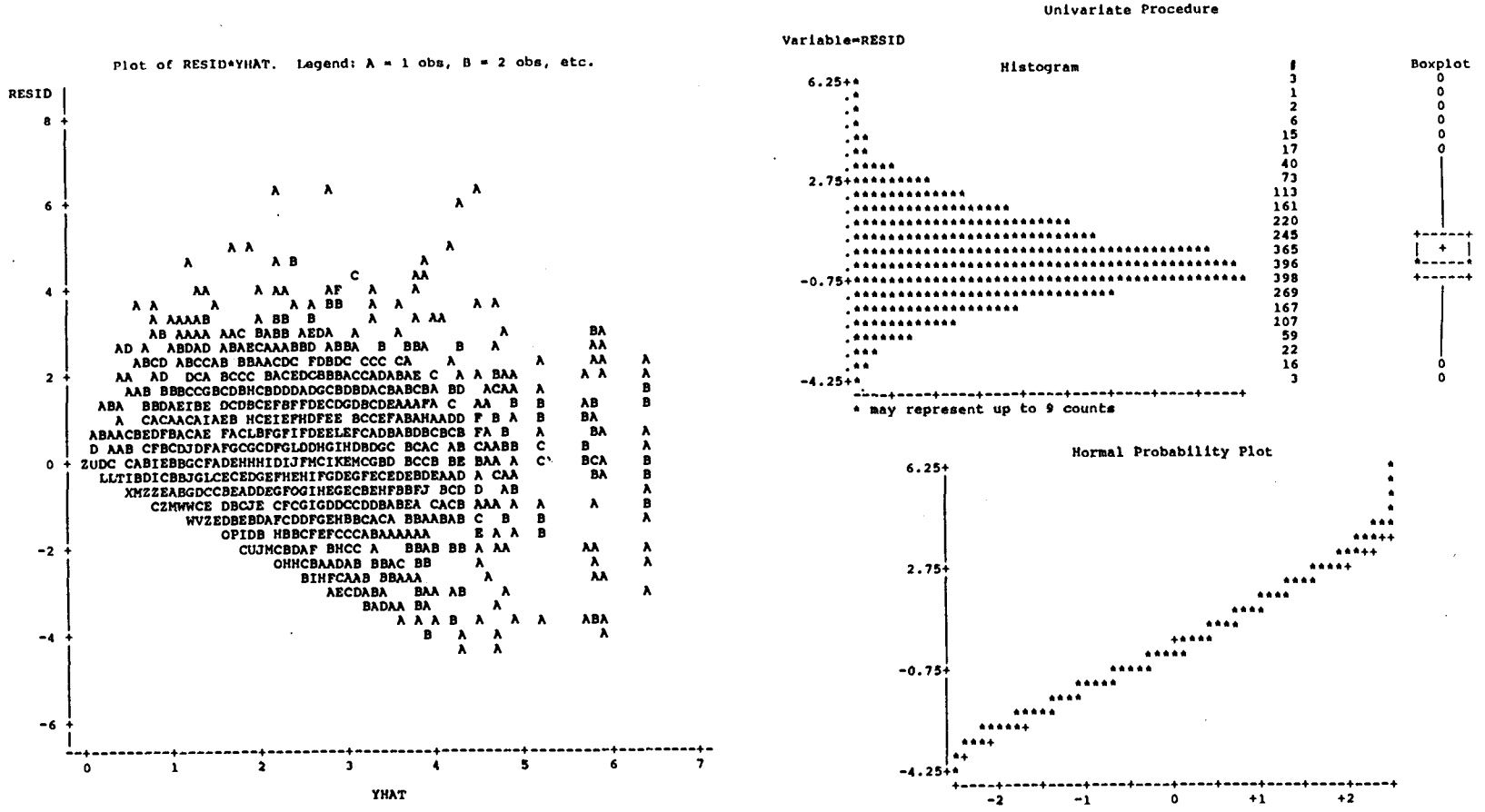

Figure 6.1. Plots of residuals.

numbers of data (no missing values), the Type III test gives the same results as the Type I test, which is the most commonly used type of test for balanced data. (Montgomery, 1988). From Table 6.3, we can conclude that the four-factor interaction and all three-factor interactions are not significant and only some two-factor interactions [algo*reso (algo stands for algorithm and reso stands for resolution), algo*feat (feat stands for feature), and reso*feat] are significant (their F values are greater than mathematically determined F value (4.55) at the 5% significance level or their pr-values are less than or equal to 0.05). In such cases, we need to go to step (4) mentioned previously for further statistical testing.

6.3.1 Results for Veneer Surface Features

No matter which algorithm was applied, two of the six veneer surface defects, blue stain and pitch streak (BS and PS), were always hard to detect compared to the other four features. The defect accuracies of blue stain and pitch streak did vary by algorithm applied, but they were always the two lowest accuracies among all features.

Some results are reported in two-way tables because there are interactions between the algorithm and feature categories. Figures 6.2 to 6.4 show the results of the paired percentage (M_2), missed percentage (M_4), and defect accuracy (M_{10}) for four edge detection and two region extraction algorithms; these are directly related to the ability to detect defects. All figures show that blue stain is the hardest one to detect (it had the lowest M_2 and M_{10}, as well as the highest M_4), and that pitch streak is the second hardest one to detect (it had the second lowest M_2 and M_{10}, as well as the second highest M_4). The detection of other types of defects varied by algorithm but all were significantly better than blue stain and pitch streak. However, the region growing/clustering combination and splitting-and-merging/region growing combination did show significant improvements in pitch streak detection. In contrast, the measures corresponding to clear wood accuracy for blue stain and pitch streak (the wrong percentage (M_3) and clear wood accuracy (M_9)) were either better or not significantly

different from the other features (Figure 6.5 (M_9) and Table 6.5 (M_3)). Table 6.4 shows the results for count percentage (M_1); blue stain results were still poor except for the compass edge detector, and pitch streak was almost always the worst (for some very big pitch streaks, the algorithms found them as a number of smaller pitch streaks so that the count number increased drastically). Using the mean centroid difference (M_5), the detected centroids of blue stain images were very far from the actual ones and much worse than any other feature. The mean centroids for pitch streaks always had the second or third worst performances (Table 6.6). For the other four features, wane was usually the hardest to detect with lower defect accuracy than the other three while the loose knot category was almost always the best except in wrong percentage (M_3) which means the defect portion was overestimated. The order of the defect types from the easiest to the hardest to detect was usually: loose knot, pitch pocket, tight knot, wane, pitch streak, and blue stain.

In the eight thresholding algorithms (four algorithms each having global and local versions), the ability of the local moment-preserving thresholding algorithm (LMP) to detect defects was very low compared to the other seven ones (defect accuracies: blue stain 6.86%, loose knot 24.51%, pitch pocket 22.02%, pitch streak 12.23%, tight knot 39.20%, and wane 21.96%). Therefore, this algorithm was not considered in the statistical analysis discussion. The global moment-preserving algorithm (CMP) really only found the loose knots and tight knots (defect accuracies: blue stain 10.75%, loose knot 83.15%, pitch pocket 60.24%, pitch streak 43.12%, tight knot 89.08%, and wane 48.49%). Compared against the better algorithms, it is the second worst performance following the local moment-preserving thresholding (LMP). Therefore, it also will not be discussed further in this chapter.

For the veneer surface features, results were similar to those found when using edge detection and region extraction algorithms. Results are reported in the two-way table format because the variables "algorithm" and "feature" interacted. Figure 6.6 to Figure 6.8 show the results for the paired percentage (M_2), missed percentage (M_4), and defect accuracy (M_{10}) for six thresholding algorithms. All figures show that blue stain is the hardest one to detect and pitch streak is the second

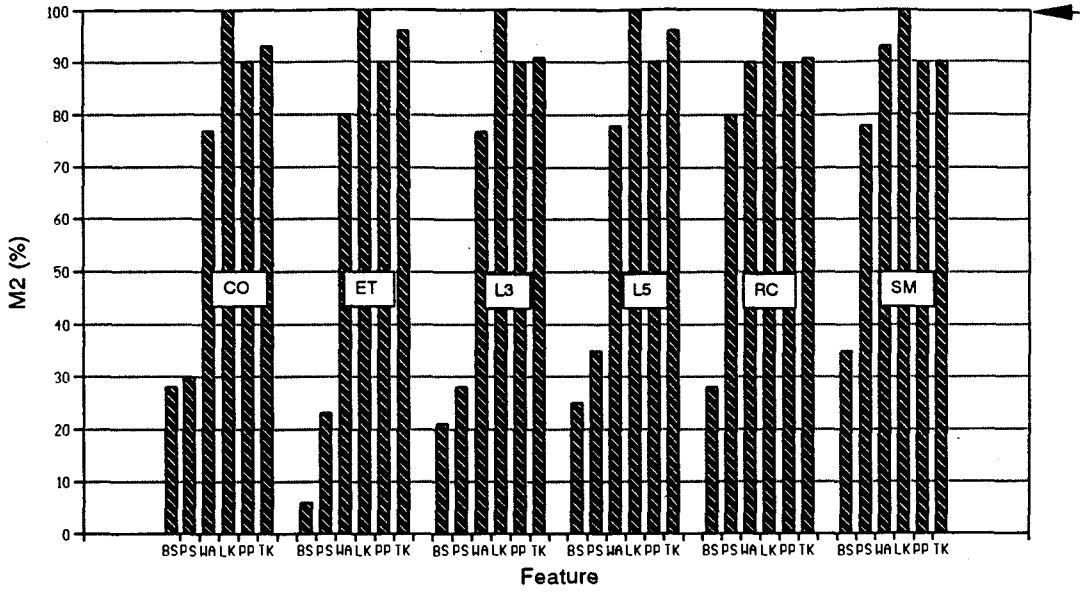

Figure 6.2. Performance results based on the paired percentage (M_2) for edge and region algorithms (arrow points to the ideal value; see text for a discussion of statistical differences).

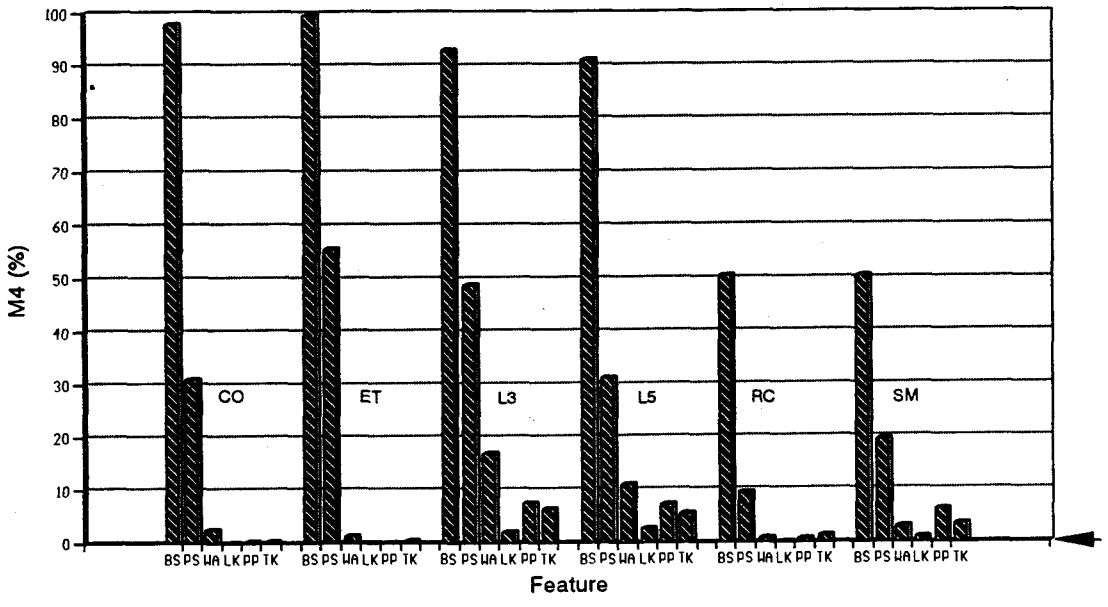

Figure 6.3. Performance results based on the missed percentage (M_4) for edge and region algorithms (arrow points to the ideal value; see text for a discussion of statistical differences).

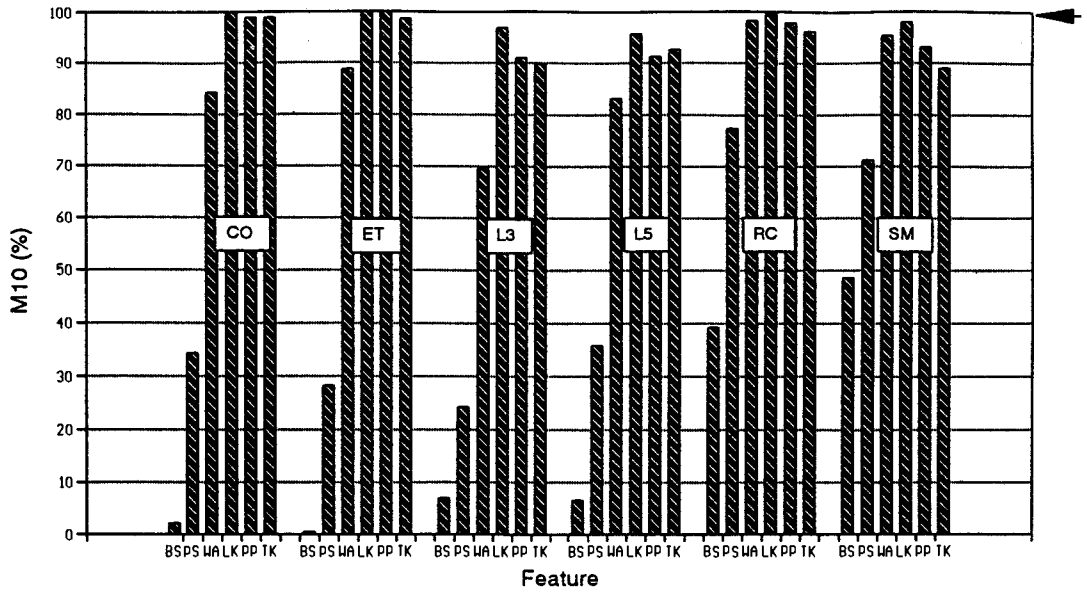

Figure 6.4. Performance results based on the defect accuracy (M_{10}) for edge and region algorithms (arrow points to the ideal value; see text for a discussion of statistical differences).

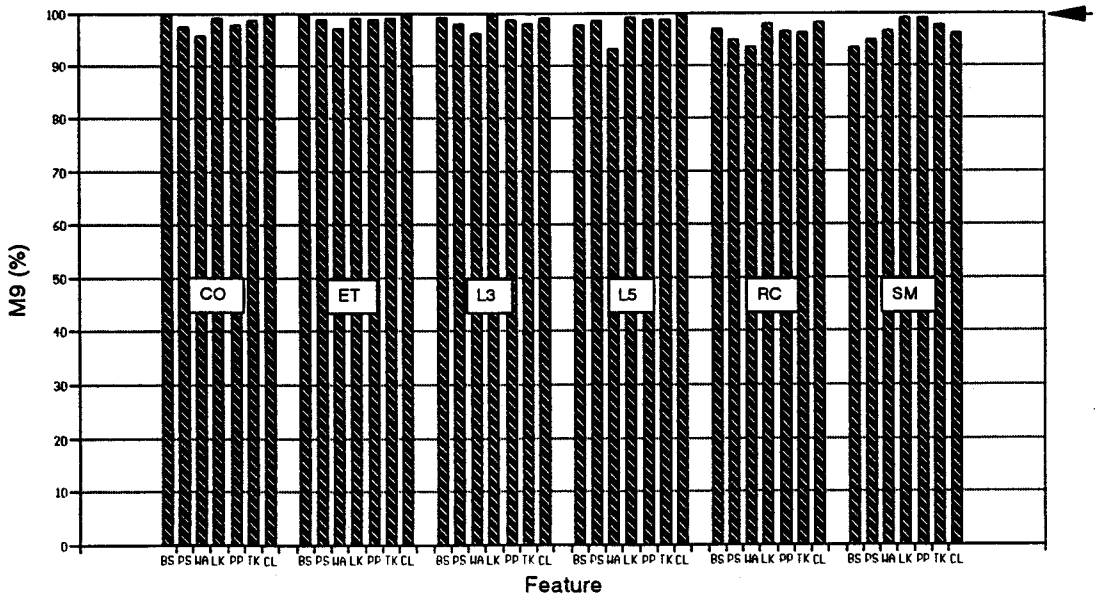

Figure 6.5. Performance results based on the clear wood accuracy (M_9) for edge and region algorithms (arrow points to the ideal value; see text for a discussion of statistical differences).

Table 6.4. Performance results based on the count percentage (M_1) for edge and region algorithms.

	CO	ET	L3	L5	RC	SM
BS	182.71	51.25	321.25	298.33	409.79	499.4
PS	620.0	385.0	515.0	455.0	357.5	300.0
WA	327.5	137.5	440.0	550.0	175.0	141.7
LK	125.0	117.5	120.0	187.5	110.0	107.5
PP	216.25	160.0	247.5	191.25	122.5	98.75
TK	255.0	145.0	198.75	176.25	280.0	192.5
CL	92.5	22.5	165.0	32.5	325.0	350.0

(the ideal value is 100%; see text for a discussion of statistical differences)

Table 6.5. Performance results based on the wrong percentage (M_3) for edge and region algorithms.

	CO	ET	L3	L5	RC	SM
BS	2.7	0.5	4.79	9.47	4.81	14.3
PS	44.86	29.07	14.79	27.39	55.17	20.65
WA	49.13	30.09	21.71	32.65	59.7	23.21
LK	116.36	119.53	56.08	61.57	169.61	69.67
PP	60.6	50.26	13.67	21.09	95.05	14.22
TK	96.14	74.19	40.54	57.12	138.3	49.85

(the ideal value is 0%; see text for a discussion of statistical differences)

Table 6.6. Performance results based on the mean centroid difference (M_5) for edge and region algorithms.

	CO	ET	L3	L5	RC	SM
BS	12.71	50.05	16.85	8.07	6.77	7.08
PS	1.63	2.35	2.3	2.06	2.58	3.26
WA	2.0	1.18	2.32	3.03	2.85	1.91
LK	0.74	0.71	1.31	1.26	1.55	0.67
PP	1.15	0.66	1.09	1.16	1.64	0.6
TK	1.1	1.08	1.36	1.59	1.73	1.33

(the ideal value is 0, see text for a discussion of statistical differences)

hardest one to detect. The measures corresponding to clear wood accuracy for blue stain and pitch streak (the wrong percentage (M_3), and clear wood accuracy (M_9)) were better or the same, as those shown in Figure 6.9 (M_9) and Table 6.8 (M_3). As shown in the Table 6.7, for the count percentage (M_1), blue stain and pitch streak are neither the worst nor the best. In M_5, the detected centroids of blue stain were very far from the actual ones and extremely worse than those of other features, and the results for pitch streak were the second worst (Table 6.9) Among the other four features, tight knots were the worst with the lowest defect accuracy and the other three were not consistent. The differences between loose knot, pitch pocket, tight knot and wane were much smaller compared to blue stain and pitch streak, and they were detected by those six thresholding algorithms with different accuracies.

The overall conclusions regarding detecting veneer surface defects are that blue stain can not be detected with any better than 62% defect accuracy by any of the algorithms just discussed; pitch streaks are also hard to detect, but some algorithms can reach 90% defect accuracy; and the other four defects can be detected with up to 95% defect accuracy by a number of the algorithms.

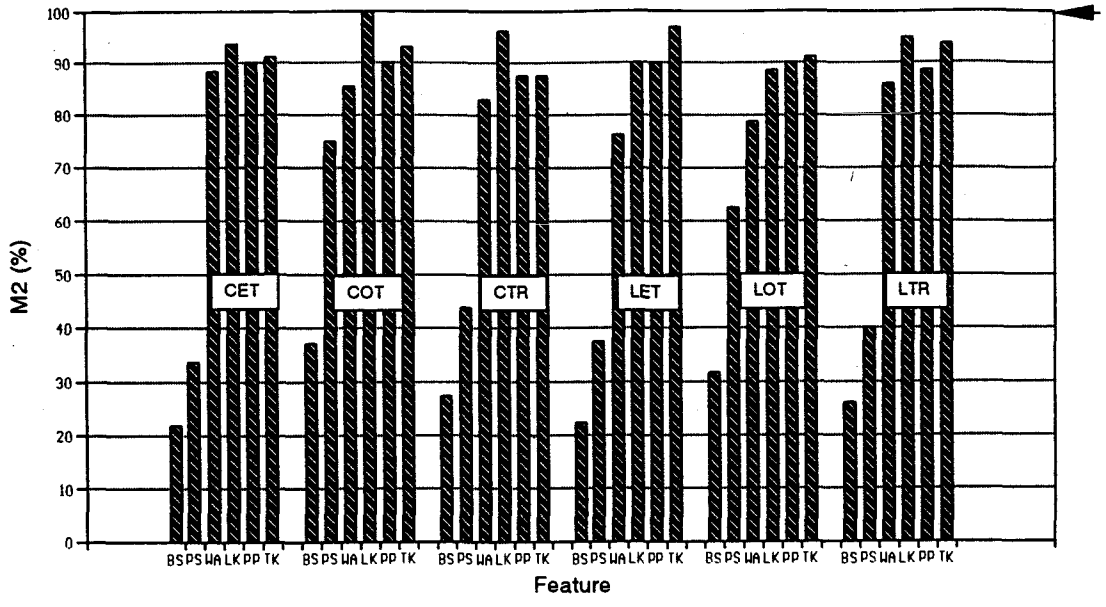

Figure 6.6. Performance results based on the paired percentage (M_2) for thresholding algorithms (the arrow points to the ideal value, see text for a discussion of statistical differences).

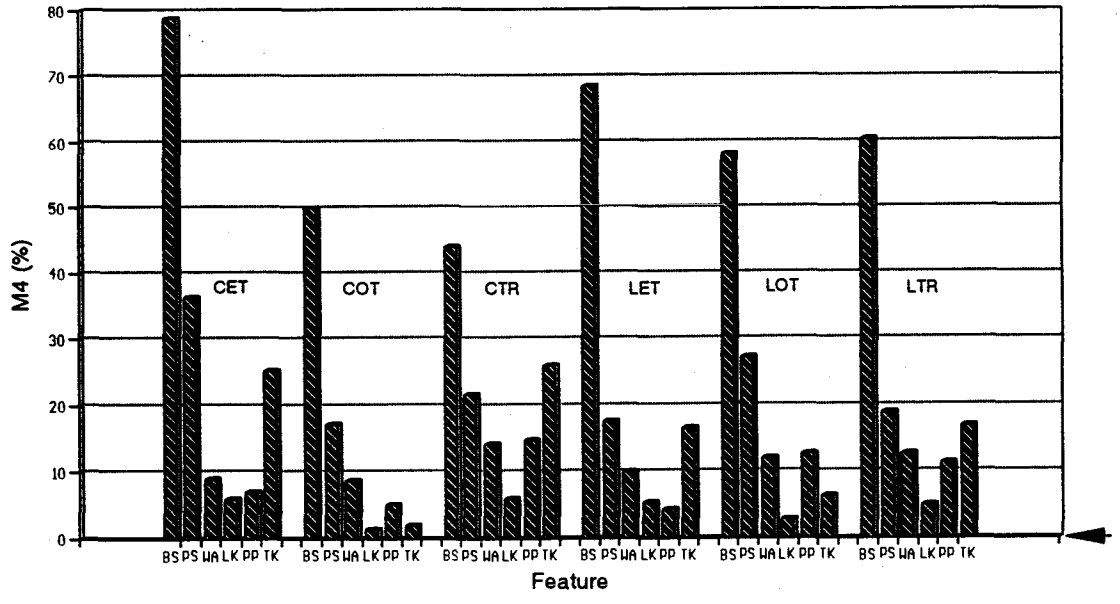

Figure 6.7. Performance results based on the missed percentage (M_4) for thresholding algorithms (the arrow points to the ideal value, see text for a discussion of statistical differences).

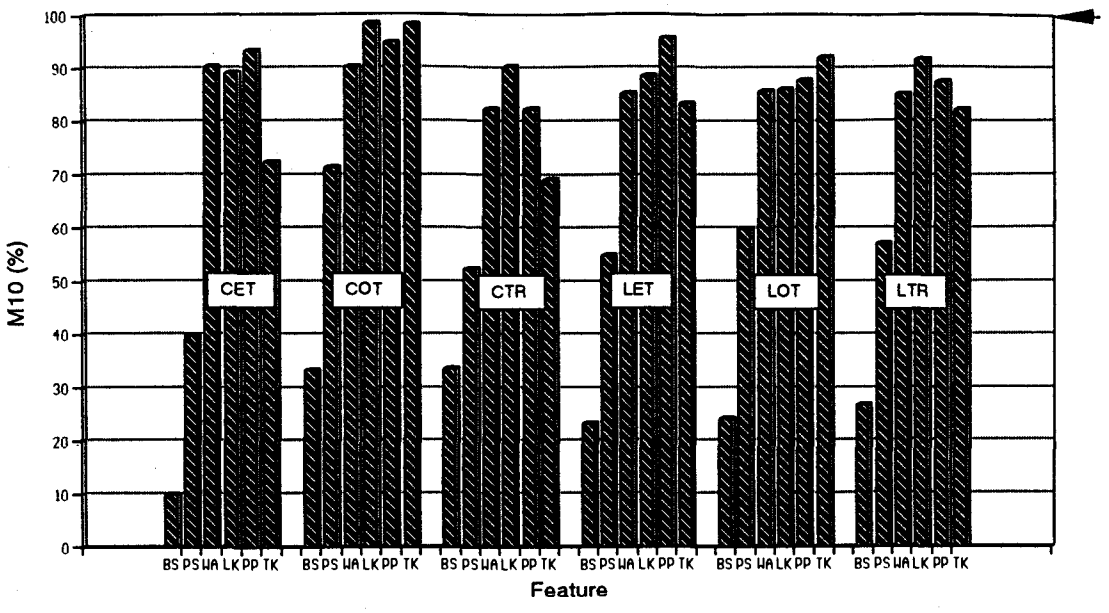

Figure 6.8. Performance results based on the defect accuracy (M_{10}) for thresholding algorithms (the arrow points to the ideal value, see text for a discussion of statistical differences).

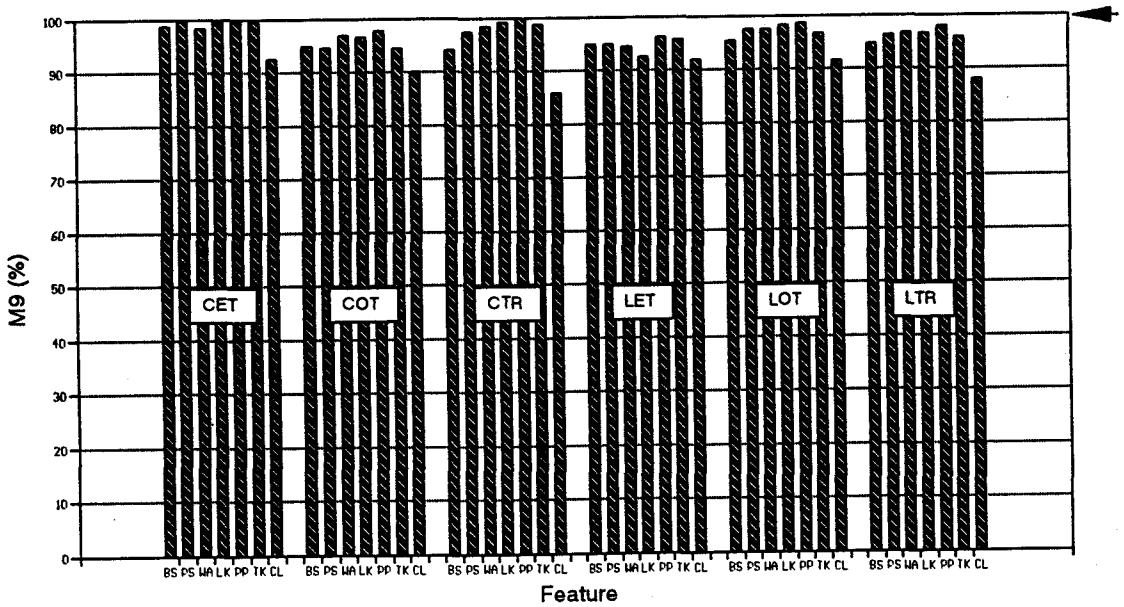

Figure 6.9. Performance results based on clear wood accuracy (M_9) for thresholding algorithms (the arrow points to the ideal value, see text for a discussion of statistical differences).

These conclusions coincide with previous research in the Department of Forest Products and were expected in this study. The reasons can be explained by the histograms of different defects and their backgrounds (clear wood). The top half of Figures 6.10 to 6.12 show the histograms for samples graded as medium blue stain, large pitch streak, and a large loose knot. The bottom half of the Figures are for their backgrounds (clear wood). Obviously, the histograms for blue stain and its background in each of the R, G, and B color components are all overlapped, even

Table 6.7. Performance results based on the wrong percentage (M_1) for thresholding algorithms.

	CET	COT	CTR	LET	LOT	LTR
BS	229.17	211.25	525.83	623.13	275.63	746.7
PS	410.0	248.75	397.5	848.75	282.5	676.3
WA	128.75	122.08	103.33	524.17	182.92	300.8
LK	123.75	181.25	113.75	677.5	202.5	406.3
PP	115.63	167.5	91.88	640.0	189.38	323.8
TK	126.25	272.5	273.75	722.5	301.88	539.4
CL	220.0	383.75	1142.5	902.5	511.25	1209.

(the ideal value is 100%; see text for a discussion of statistical differences)

for the best component R (Figure 6.10). The phenomenon also happened in pitch streaks (Figure 6.11). However, the histograms for the loose knot shifted so that the defect stands out from with only minimal overlapping of pixels (Figure 6.12).

Table 6.8. Performance results based on the wrong percentage (M_3) for thresholding algorithms.

Table 6.8. Performance results based on the wrong percentage (M_3) for thresholding algorithms.

	CET	COT	CTR	LET	LOT	LTR
BS	48.17	40.6	42.1	79.21	78.28	96.31
PS	6.6	104.92	27.53	12.55	18.89	9.6
WA	17.31	139.13	8.9	27.97	30.76	26.39
LK	32.93	974.3	46.9	1210	153.65	64.35
PP	6.74	198.23	3.17	35.81	21.88	44.19
TK	20.81	443.73	217.82	48.92	118.52	188.5

(the ideal value is 100%; see text for a discussion of statistical differences)

Table 6.9. Performance results based on mean centroid difference (M_5) for thresholding algorithms.

	CET	COT	CTR	LET	LOT	LTR
BS	30.76	17.43	12.64	25.2	19.21	20.15
PS	8.73	8.87	7.77	4.73	9.34	5.87
WA	1.84	5.1	1.82	2.22	3.16	3.04
LK	0.97	3.08	1.02	3.88	2.57	1.68
PP	0.52	2.58	0.57	2.38	0.99	1.89
TK	1.57	5.05	2.02	1.77	3.46	2.9

(the ideal value is 0; see text for a discussion of statistical differences)

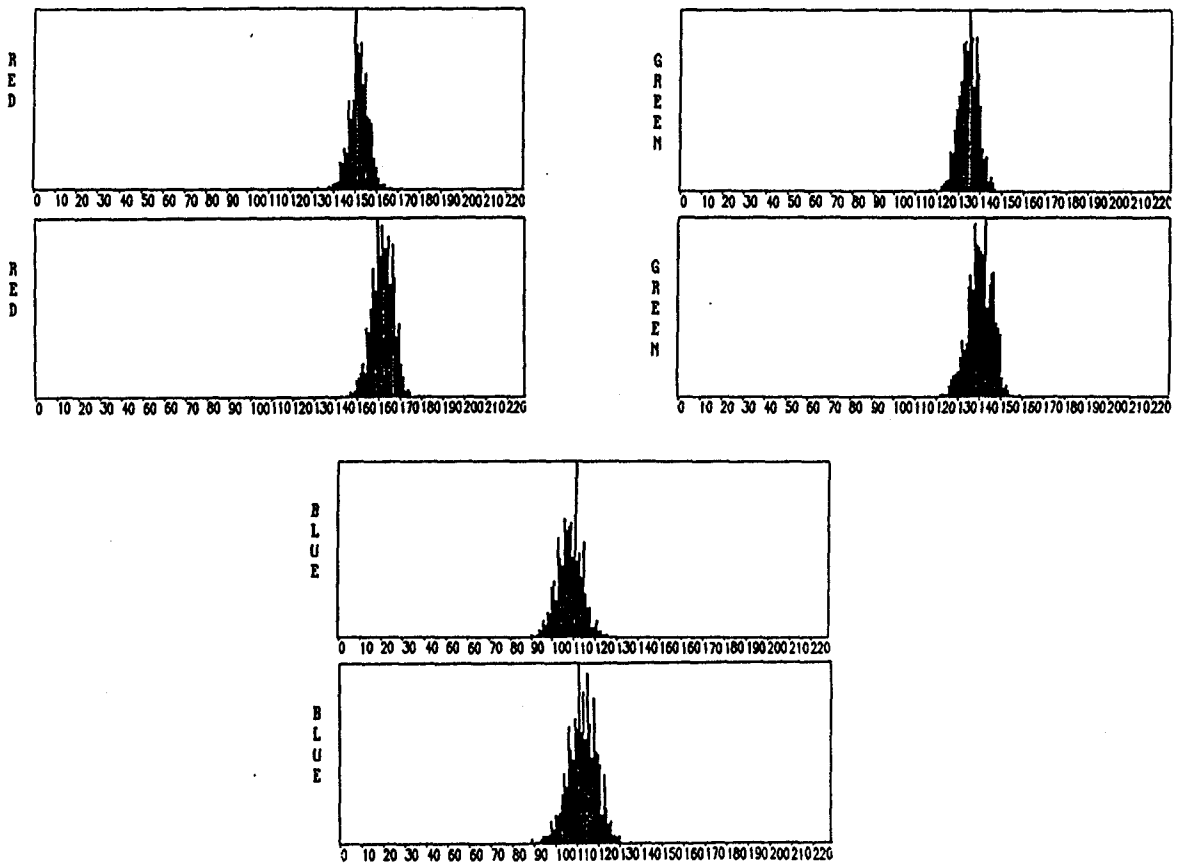

Figure 6.10. Histograms for a sample of blue stain (upper plot) and its background (clear wood, lower plot) in (a) red, (b) green, and (c) blue components of the RGB color space.

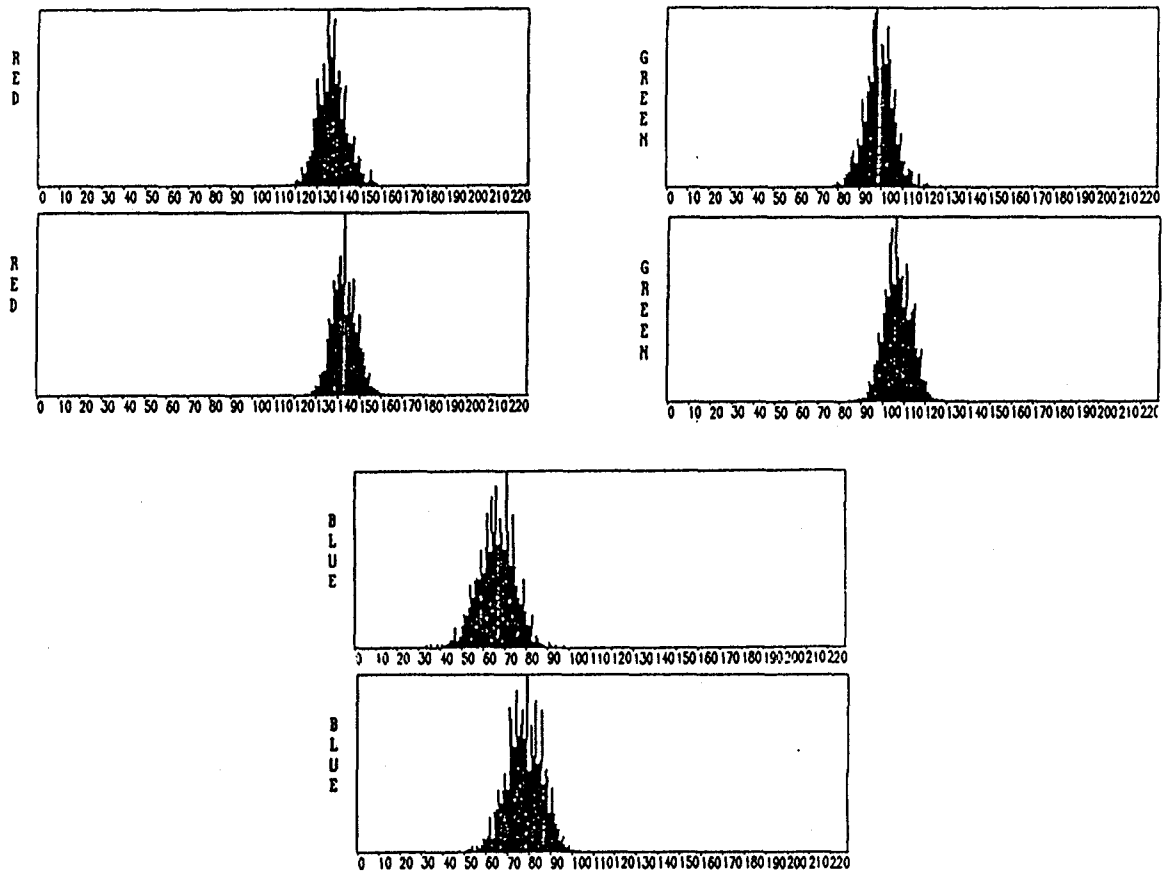

Figure 6.11. Histograms for a sample of pitch streak (upper plot) and its background (clear wood, lower plot) in (a) red, (b) green, and (c) blue components of the RGB color space.

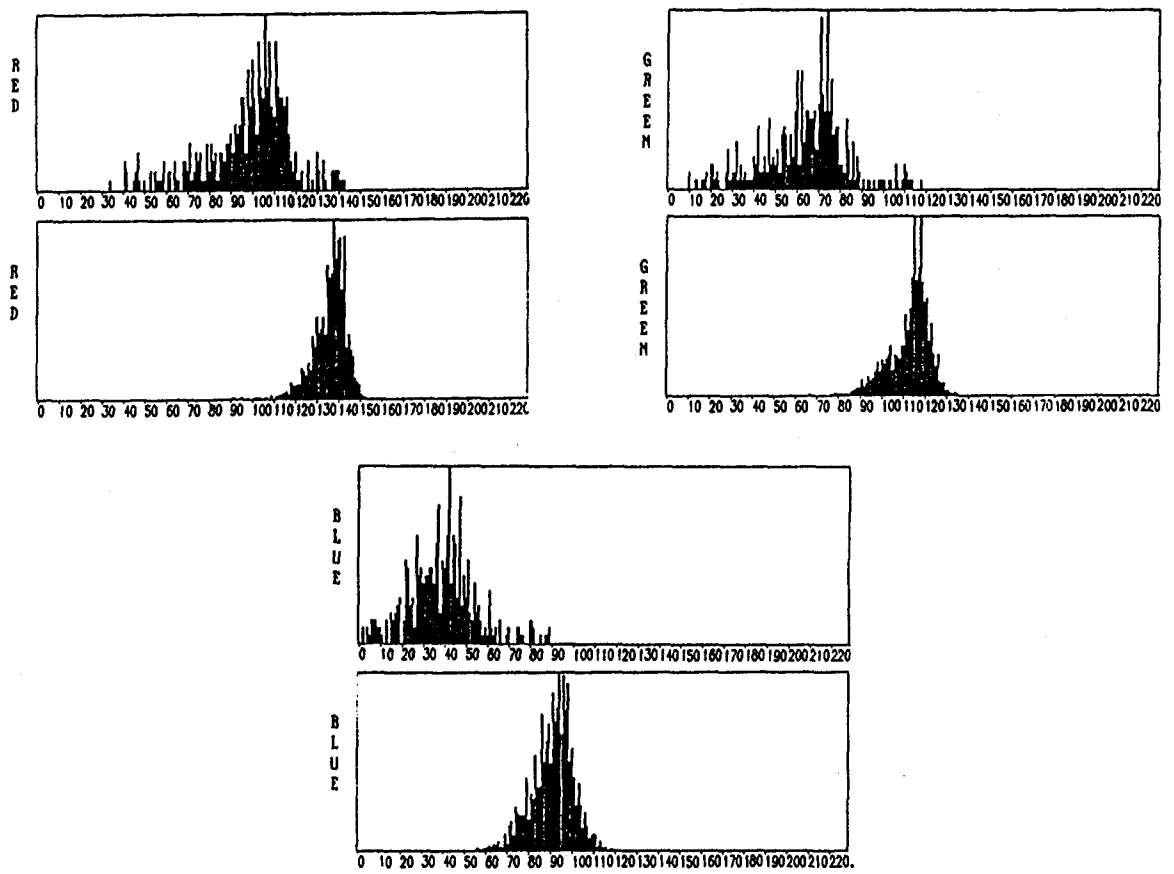

Figure 6.12. Histograms for a sample of loose knot (upper plot) and its background (clear wood, lower plot)) in (a) red, (b) green, and (c) blue components of the RGB color space.

As mentioned in Chapter 1, no matter which algorithm is used to perform segmentation, the information about differences in certain image characteristics must be used, either the differences between pixels (edge detection) or the differences between groups of pixels (region extraction and thresholding). This is why the loose knot could be detected with relatively high defect accuracy while blue stain and pitch streak could not.

6.3.2 Camera Resolution Effects

6.3.2.1 Edge and region algorithms camera resolution effects

For the edge detection and region extraction algorithms, the two camera resolutions did not make any statistically significant difference for all combinations of algorithms, colors, and veneer surface features using any of the seven measures. This is illustrated in the Table 6.10 where the means and standard deviations are listed for all algorithms combined. Considering execution speed, the 5-bit camera resolution is therefore preferable for all edge detection and region extraction algorithms.

This result for camera resolution is related to the mechanisms used by three of the four edge detection algorithms and two region extraction algorithms. Camera resolution corresponds to how many bits in a digital computer are used to represent the range of each color component of a tri-parameter color space (RGB or Lab) or a single parameter gray-scale space (256 for an 8-bit camera resolution and 32 for a 5-bit). The parameters that control defect detection in the two region algorithms (the weighting factors for the method combining the splitting-and-merging with region growing and the method combining clustering and region growing) were determined heuristically from the sample images for each color space. Three of the four edge detection algorithms used a 3 by 3 window. Only the nonlinear Laplace edge detector was different because it utilized two sizes of windows at once. The compass edge detector finds the gradient around the central pixel in a window. The gradient is not

Table 6.10. Results of measures under two camera resolutions for all edge detection and region extraction algorithms combined.

Measure	Resolution	Mean	Std. Dev.
M_1	5-bit	250.19%	319.12
	8-bit	235.79%	282.13
M_2	5-bit	71.42%	43.78
	8-bit	73.52%	42.71
M_3	5-bit	80.64%	101.59
	8-bit	83.71%	109.57
M_4	5-bit	13.17%	25.52
	8-bit	13.04%	25.56
M_5	5-bit	5.10	11.28
	8-bit	4.87	10.73
M_9	5-bit	97.76%	3.79
	8-bit	97.88%	3.58
M_{10}	5-bit	72.59%	37.22
	8-bit	73.08%	37.50

(no significant differences)

in terms of absolute values of the pixels in the window. Rather, it is a measure of relative change. The entropy edge detector finds the entropy of the center pixel in the window but that value is weighted by the average entropies of all other pixels in the window. The nonlinear Laplace edge detector simply finds the maximum difference between pixels inside the window as the edge strength of the central pixel. Therefore, changes in the absolute values in the same window of an image would only change the absolute edge strength of that central pixel in the window but would not significantly affect its position in the histogram of the resulting edge strength map. If the edge strength is large in a 5-bit image, it will still be large in an 8-bit image. The final

binary edge map was obtained by determining which image pixels had an edge strength over a threshold value obtained heuristically from sample images.

6.3.2.2 Thresholding algorithms camera resolution effects

The results for camera resolution effects in thresholding algorithms were more complicated than for the edge detection and region extraction algorithms. None of the camera resolutions was consistent for all measures. For instance, the resolution that could help improve defect accuracy would reduce clear wood accuracy in most experimental treatments.

The results are reported in several two-way tables because the algorithm and feature factors had interactions. The next four figures (Figures 6.13 to 6.16) and three tables (Tables 6.10 to 6.13) show the results. The top value in each cell of tables is the one corresponding to the 5-bit camera resolution and the bottom one corresponds to the 8-bit resolution. An asterisk in the top-left of a cell in a table or underneath the two abbreviated feature letters in a figure indicates that the results for that combination of factors are significantly different by both the statistical test and the rules-of-thumb criteria. Because there were interactions between algorithms and features for all measures, all results are reported in two-way tables.

When looking at these figures, it might seem confusing that some performance measure pairs are not significantly different even though they have even larger differences in means than those claimed as being significantly different. From a statistical point of view, this phenomenon is caused by variability of sample values used to calculate the mean value. For instance, Figure 6.17 shows the box-and-whisker plots for M_{10} under 5-bit and 8-bit camera resolutions using the global Otsu thresholding algorithm (COT) on pitch streaks. In a box-and-whisker plot, the central]box covers the middle 50% of the data values, the whiskers extend out to the minimum and maximum values, and the central line is at the median. The ranges of the two plots in Figure 6.17 are nearly overlapped for both boxes and whiskers.

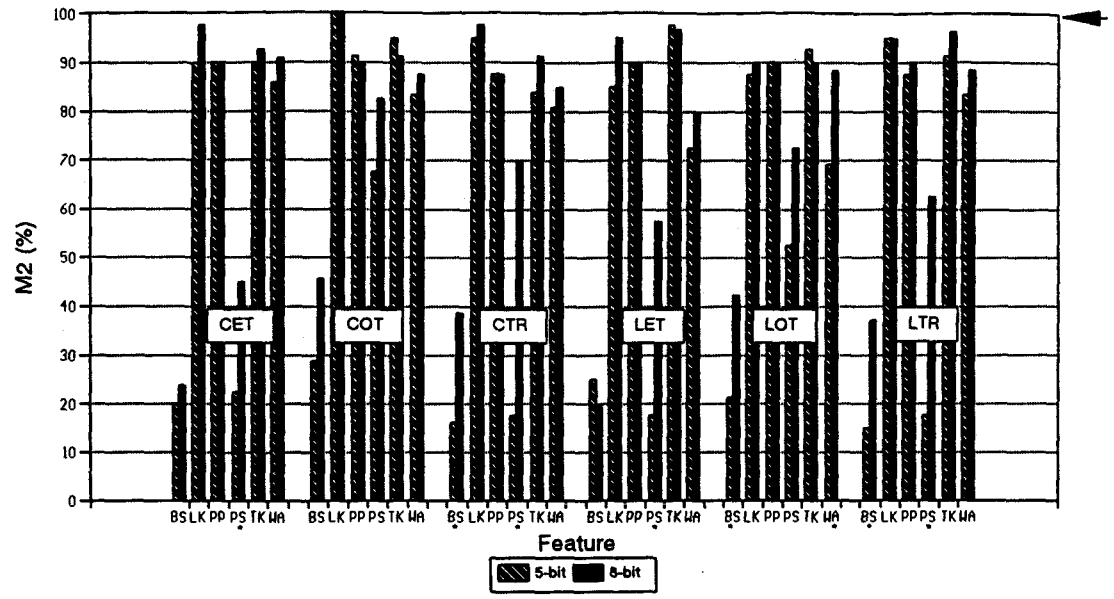

Figure 6.13. Performance results based on the paired percentage (M_2) for thresholding algorithms (arrow points to the ideal value, and asterisk indicates a statistically significant difference at the 5% level and the difference also meets the rules-of-thumb).

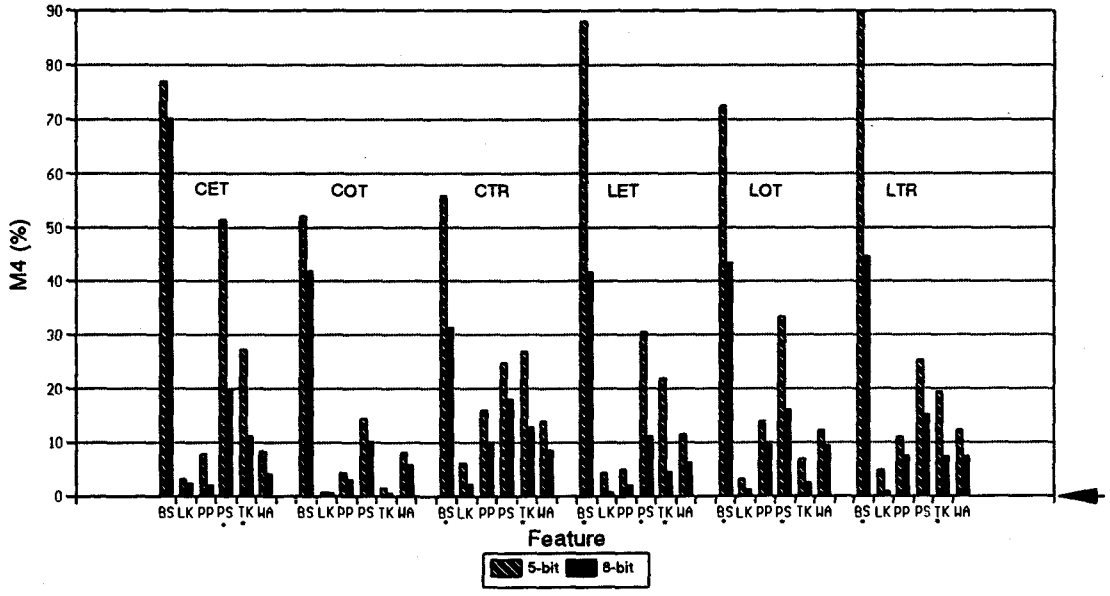

Figure 6.14. Performance results based on the missed percentage (M_4) for thresholding algorithms (arrow points to the ideal value, and asterisk indicates a statistically significant difference at the 5% level and the difference also meets the rules-of-thumb).

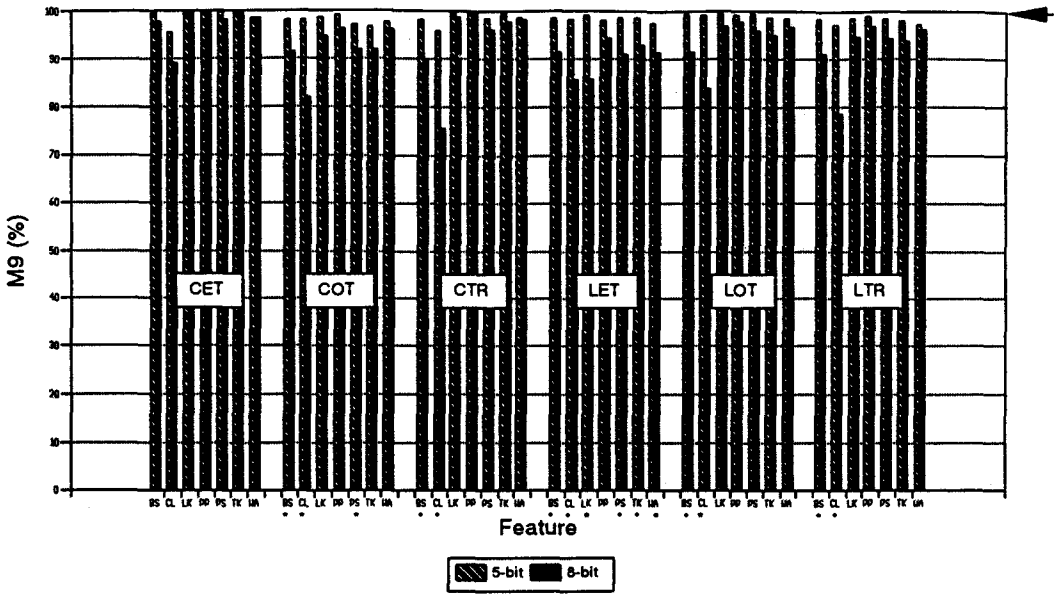

Figure 6.15. Performance results based on the clear wood accuracy (M_9) for thresholding algorithms (arrow points to the ideal value, and asterisk indicates a statistically significant difference at the 5% level and the difference also meets the rules-of-thumb).

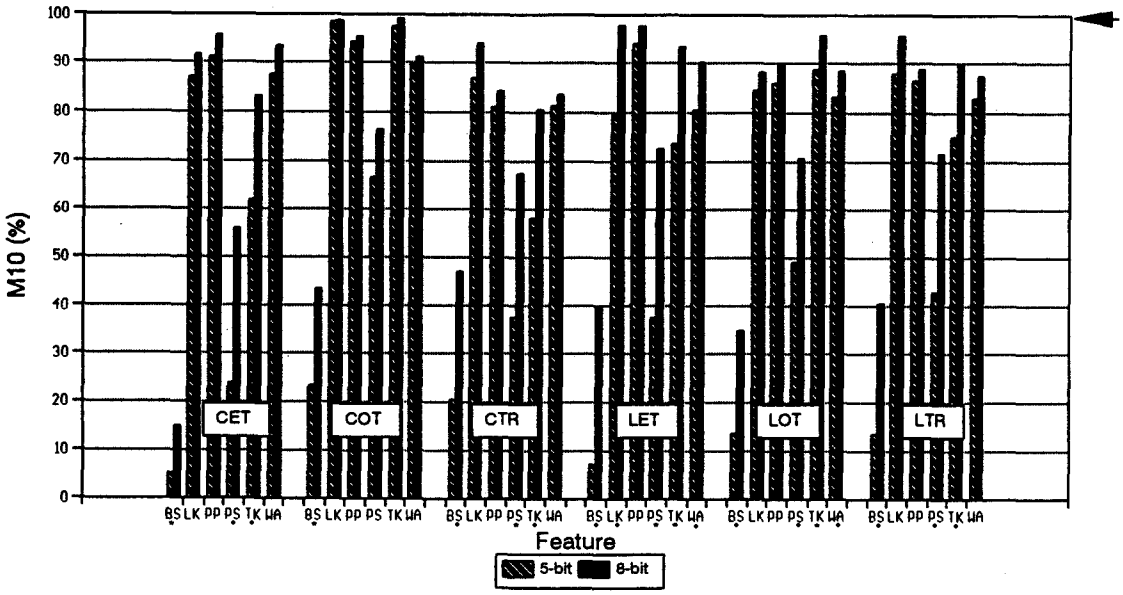

Figure 6.16. Performance results based on the defect accuracy (M_{10}) for thresholding algorithms (arrow points to the ideal value, and asterisk indicates a statistically significant difference at the 5% level and the difference also meets the rules-of-thumb).

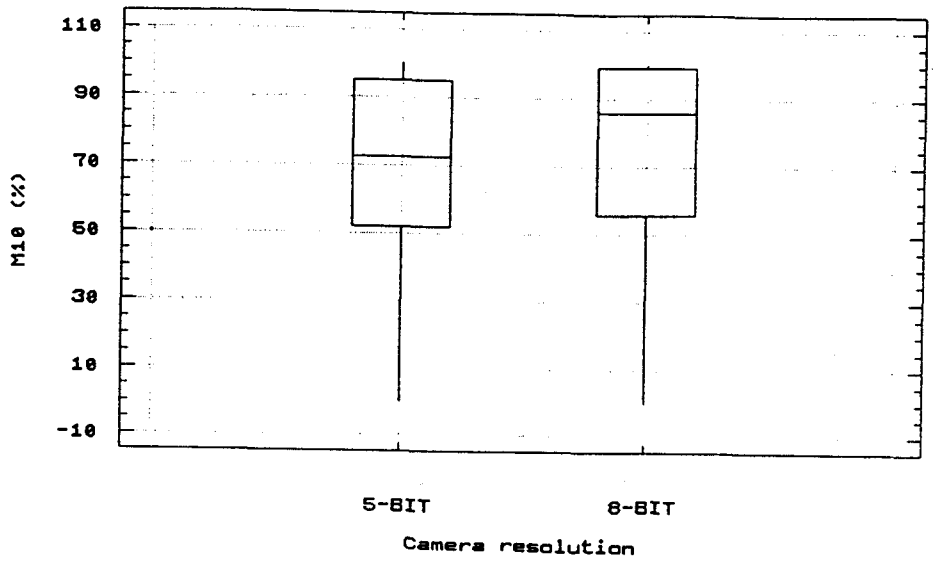

Figure 6.17. A box-and-whisker plot for defect accuracy (M_{10}) under 5-bit and 8-bit camera resolutions for pitch streak samples using the global Otsu thresholding algorithm.

Therefore, they are not considered to be significantly different at the 5% level even though the median values are quite different.

For the global entropy algorithm (CET), the 8-bit camera resolution was in general better than the 5-bit resolution. However, the 8-bit resolution was significantly better in count percentage (M_1) for blue stain, paired percentage (M_2) for pitch streaks, as well as missed percentage (M_4) and defect accuracy (M_{10}) for pitch streaks and tight knots.

For the global Otsu thresholding algorithm (COT), the 5-bit resolution results were in general better than the 8-bit resolution results. However, the results of the 5-bit resolution were better than the 8-bit in count percentage (M_1) for clear wood; wrong percentage (M_3) for blue stain, loose knots and tight knots; mean centroid difference (M_5) for loose knots and tight knots; and clear wood accuracy (M_9) for blue stain, clear wood and pitch streaks.

Table 6.11. Performance results based on the count percentage (M_1) under two camera resolutions.

	CET	COT	CTR	LET	LOT	LTR
BS	* 26.9 270.6	 143.6 211.0	 484.0 285.9	* 171.9 892.6	* 176.8 284.1	* 812.8 427.5
CL	* 5.0 160.9	* 119.7 536.8	 720.9 888.8	* 30.3 1608.0	* 103.2 809.7	 872.9 1006.
LK	 126.9 109.9	 138.7 191.7	 102.9 105.2	* 208.7 1019.2	 130.8 223.6	 344.4 320.9
PP	 109.1 117.3	 158.1 138.3	 89.4 86.1	* 245.9 881.3	 154.8 179.3	 294.82- 04.8
PS	 238.4 307.8	 232.4 202.8	 408.0 201.2	* 513.3 934.3	 229.9 264.6	* 808.9 361.2
TK	 115.3 125.5	 216.4 259.4	 158.3 184.1	* 270.0 1024.5	 237.7 284.8	 497.9 401.8
WA	 113.9 129.9	 119.3 107.0	* 113.6 84.3	 376.9 452.9	* 197.1 136.6	* 302.3 180.9

(The top value in each cell is for the 5-bit resolution and the bottom one is for the 8-bit; an asterisk indicates a statistically significant difference at the 5% level and the difference also meets the rules-of-thumb.)

For the global transition-matrix algorithm (CTR), the 5-bit resolution was better than the 8-bit in wrong percentage (M_3) for blue stain and loose knot; mean centroid difference (M_5) for blue stain; and clear wood accuracy (M_9) for blue stain and clear wood. The 8-bit resolution results were better than the 5-bit results in

Table 6.12. Performance results based on the wrong percentage (M_3) under two camera resolutions.

	CET	COT	CTR	LET	LOT	LTR
BS	1.73 0.61	* 2.09 19.18	1.19 19.38	1.34 17.53	* 1.62 37.16	* 0.74 20.74
LK	* 13.22 30.78	* 75.99 304.09	* 8.91 24.51	* 15.11 78.58	* 20.43 51.17	* 12.84 46.94
PP	* 2.1 5.26	5.97 19.95	0.73 1.32	10.88 15.11	1.35 2.54	1.95 5.75
PS	0.62 4.62	18.69 43.53	9.01 10.08	1.99 11.06	2.62 15.37	1.89 8.86
TK	3.0 10.86	* 68.67 312.68	2.82 9.62	* 3.26 27.9	* 15.22 67.51	* 8.91 38.53
WA	5.23 9.51	9.97 18.46	2.56 3.06	2.23 8.28	* 3.17 16.01	4.8 13.66

(The top value in each cell is for the 5-bit resolution and the bottom one is for the 8-bit; an asterisk indicates a statistically significant difference at the 5% level and the difference also meets the rules-of-thumb.)

paired percentage (M_2) for blue stain and pitch streaks; missed percentage (M_4) for blue stain and tight knots, and defect accuracy (M_{10}) for blue stain, pitch streaks, and tight knots. From a defect detection point-of-view, it might be appropriate to reduce clear wood accuracy by 20% in the clear wood feature in order to gain over 25% in defect accuracy for blue stain, pitch streaks, and tight knots.

For the local entropy thresholding algorithm (LET), the 5-bit resolution results were better than the 8-bit in count percentage (M_1) for all features except wane; in

Table 6.13. Performance results based on the mean centroid difference (M_5) under two camera resolutions.

	CET	COT	CTR	LET	LOT	LTR
BS	11.59 8.5	8.49 8.18	* 3.4 7.45	17.42 11.06	7.98 5.85	13.648. 81
LK	0.48 0.71	* 0.86 2.12	0.58 0.64	0.67 1.03	* 0.83 1.47	0.72 0.98
PP	0.42 0.46	0.48 0.95	0.49 0.37	0.9 0.56	0.52 0.5	0.52 0.71
PS	* 7.26 2.46	4.45 4.77	7.61 3.29	2.85 2.12	3.69 5.26	3.76 2.91
TK	0.99 1.0	* 1.48 3.61	1.0 1.09	0.96 1.13	* 1.08 1.8	1.33 1.47
WA	1.12 1.29	* 1.74 2.07	0.89 1.11	0.93 1.13	* 1.1 2.2	1.37 1.75

(The top value in each cell is for the 5-bit resolution and the bottom one is for the 8-bit; an asterisk indicates a statistically significant difference at the 5% level and the difference also meets the rules-of-thumb.)

wrong percentage (M_3) for blue stain, loose knots, and tight knots; and in clear wood accuracy (M_9) for all features except pitch pockets. The 8-bit results were better than the 5-bit in paired percentage (M_2) for pitch streaks; missed percentage (M_4) for blue stain, pitch streaks, and tight knots; and in defect accuracy (M_{10}) for all features except pitch pockets. If the count percentage is very important, then 5-bit resolution should be used for the local entropy thresholding algorithm (LET). If defect accuracy is of greater concern than the count percentage, then the 8-bit resolution should be

used for the local entropy thresholding (LET). Because there was no clear tendency, the resolution decision for this algorithm really depends on the application.

For the local Otsu thresholding algorithm (LOT), the 5-bit resolution results were better than the 8-bit results in count percentage (M_1) for blue stain and clear wood; wrong percentage (M_3) for all features except pitch pockets; mean centroid difference (M_5) in loose knots, tight knots, and wane; and in clear wood accuracy (M_9) for blue stain and clear wood. The 8-bit resolution results were better than the 5-bit results in count percentage (M_1) for wane; paired percentage (M_2) for blue stain and wane; missed percentage (M_4) for blue stain and pitch streaks; and defect accuracy (M_{10}) for blue stain, pitch streaks, tight knots, and wane. Overall, the 8-bit resolution results were a little better than the 5-bit results. Therefore, the 8-bit camera resolution is suggested for the local Otsu thresholding algorithm (LOT).

For the local transition-matrix thresholding algorithm (LTR), the 5-bit resolution results were better than the 8-bit results in wrong percentage (M_3) for blue stain, loose knots, and tight knots; and in clear wood accuracy (M_9) for clear wood and tight knots. The 8-bit results were better than the 5-bit in count percentage (M_1) for blue stain, pitch streaks, and wane; paired percentage (M_2) for blue stain and pitch streaks; missed percentage (M_4) for blue stain and tight knots; and defect accuracy (M_{10}) for blue stain, pitch streaks and tight knots. When comparing the amount gained to the amount lost using an 8-bit camera resolution, the 8-bit resolution seems preferable.

The conclusions about camera resolutions for the six thresholding algorithms are:

(1) for the global entropy thresholding (CET), global transition-matrix thresholding (CTR), local Otsu thresholding (LOT), and local transition-matrix thresholding (LTR) algorithms, use an 8-bit camera resolution;

(2) for the local entropy thresholding (LET) algorithm, if the count percentage is a key measure, then use a 5-bit camera resolution; if the

defect accuracy is of greater concern, then use an 8-bit camera resolution; and

(3) for the global Otsu thresholding (COT) algorithm, use a 5-bit camera resolution.

These conclusions are not surprising given that the 8-bit camera resolution gives more detail for an image than the 5-bit resolution. Therefore, more precise threshold levels can be selected. For instance, it is a common situation in images with only 32 levels that choosing 10 as a threshold value would under-estimate the defect but 11 would over-estimate the defect. For the same case, if that image is grabbed with an 8-bit camera resolution and thus has 256 levels, the 10 in the 5-bit image then corresponds to about 82 (random noise would affect the value somewhat) in an 8-bit image while 11 corresponds to 90. Thus, an ideal threshold value can be set as 86. However, the ideal situation does not always hold true in the automatic threshold selection algorithms. In these cases, an algorithm's capability of selecting a good threshold is restricted by certain statistical or mathematical methods.

Overall, the 8-bit camera resolution is best suited for all thresholding algorithms except the global Otsu thresholding algorithm (COT). For the edge detection and region extraction algorithms, camera resolution does not make any significant difference. In order to make all further discussions about data analysis easier and more consistent, the 8-bit camera resolution results will be used from this point on in this thesis.

6.3.3 Color Space Effects

6.3.3.1 Comparison of the RGB and Lab color spaces

As discussed in Chapter 5, tests were performed just to determine performance differences between the RGB color space and Lab color space. For the edge detection and region extraction algorithms, the RGB and Lab color spaces are to be compared,

while for the thresholding algorithms, the individual components of the RGB color space (R, G, and B) and L of the Lab color space are to be compared. All color space effects are in terms of the 8-bit camera resolution.

6.3.3.1.1. Comparison of the RGB and Lab color spaces for edge detection and region extraction algorithms

Table 6.14 lists the results of count percentage (M_1) under the RGB and Lab color spaces. The top value in each table cell corresponds to the M_1 under the RGB color space and the bottom one corresponds to the M_1 under the Lab color space. The asterisk indicates that a significant difference existed. In most of cases, the RGB color space algorithms identified a greater number of false defects so that it was significantly worse for the following feature and algorithm combinations:

(1) clear wood for compass edge detection (CO) and entropy edge detection (ET);

(2) tight knots for compass edge detection;

(3) wane for compass edge detection and nonlinear Laplace edge detection using window sizes of 3 and 5; and

(4) pitch pockets for entropy edge detection.

In contrast, the RGB color space was significantly better for:

(1) blue stain for the compass edge detector and nonlinear Laplace edge detection with a window size of 3; and

(2) pitch streak for nonlinear Laplace edge detector with a window size of 3.

The RGB and Lab color spaces were not significantly different for the measures paired percentage (M_2) and mean centroid difference (M_5). For the wrong percentage (M_3) and missed percentage (M_4), only algorithms and colors showed interactions. Figures 6.18 and 6.19 show the results of M_3 and M_4 under the RGB and Lab color spaces, respectively. An asterisk underneath the abbreviated algorithms names indicates that a significant difference existed. The only significant differences for both M_3 and M_4 were the nonlinear Laplace edge detector (with both window sizes

of 3 and 5), where the RGB color space was significantly better than Lab with about 45% and 110% differences in the wrong percentage (M_3), while RGB was significantly worse than Lab with about 30% and 35% differences in the missed percentage (M_4).

For the measures clear wood accuracy (M_9) and defect accuracy (M_{10}), only algorithms and colors showed interactions. The RGB or Lab color spaces were not significantly different for any of the algorithms as shown in Figures 6.20 and 6.21.

Combining all of the results, the significant differences occurred in count percentage, wrong percentage, and missed percentage (M_1, M_3, and M_4). These results indicate that the RGB color space has better defect detection capabilities than the Lab color space, especially, for the defects blue stain and pitch streaks. If the concern of finding all defects is greater than the concern of finding some false defects, then the RGB color space is better than the Lab color space. Otherwise, Lab can be used.

6.3.3.1.2. Comparison of the RGB and Lab color spaces for thresholding algorithms

Table 6.15 lists the results of count percentage (M_1) for the color components L, R, G, and B in order from top to bottom in each cell. The results indicate that L and Lab color spaces for edge detection and region extraction algorithms (arrow points to the ideal value).

For the paired percentage measure (M_2), only algorithms and colors showed interactions. Figure 6.22 shows the results of M_2 under the RGB and Lab color spaces, where L was not really any better than R, G, and B for the features blue stain, pitch streaks and wane.

For the wrong percentage measure (M_3), there was no interaction at all. The result was that L was not significantly different from R, G, and B (10.95% for R, 10.97% for L, 13.3% for gray, 14.93% for G, and 18.31% for B). The missed

Table 6.14. Performance results based on the count percentage (M_1) under the RGB and Lab color spaces for edge and region algorithms.

	CO	ET	L3	L5	RC	SM
BS	* 163.2 24.6	 19.63 4.7	* 260.9 7.2	 258.1 137.2	 330.9 330.4	 286.1 339.2
CL	* 64.4 14.9	* 14.87 143.1	 39.5 0.0	 14.9 7.2	 231.7 191.5	 268.0 272.4
LK	 123.2 114.4	 108.3 116.9	 108.3 108.3	 182.2 116.9	 100.0 108.3	 100.0 100.0
PP	 159.5 108.3	* 137.7 99.99	 94.0 94.3	 135.2 131.0	 140.5 94.3	 100.0 100.0
PS	 418.9 270.3	 247.8 308.4	* 176.3 298.4	 299.6 366.1	 289.2 215.0	 265.1 268.3
TK	* 245.7 115.5	 131.0 116.9	 106.9 120.3	 142.1 123.2	 270.3 238.0	 187.7 187.6
WA	* 255.5 132.5	 129.4 125.9	* 190.8 105.8	* 372.9 159.5	 182.4 130.7	 126.4 108.8

(The top value in each cell is for the RGB color space and the bottom one is for the Lab; an asterisk indicates a statistically significant difference at the 5% level and the difference also meets the rules-of-thumb.)

percentage (M_4) between the color components R, G, B and L were not statistically significant different.

For the mean centroid different measure (M_5), only algorithms and colors showed interactions. The Table 6.16 lists the results of M_5 for L, R, G and B from

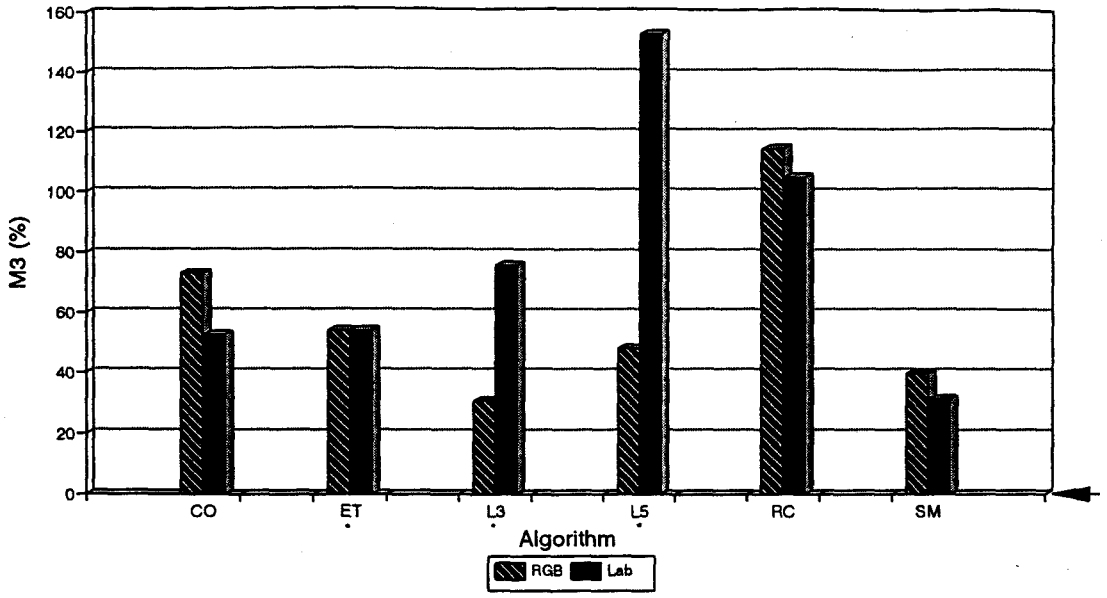

Figure 6.18. Performance results based on the wrong percentage (M_3) under the RGB and Lab color spaces for edge detection and region extraction algorithms (arrow points to the ideal value).

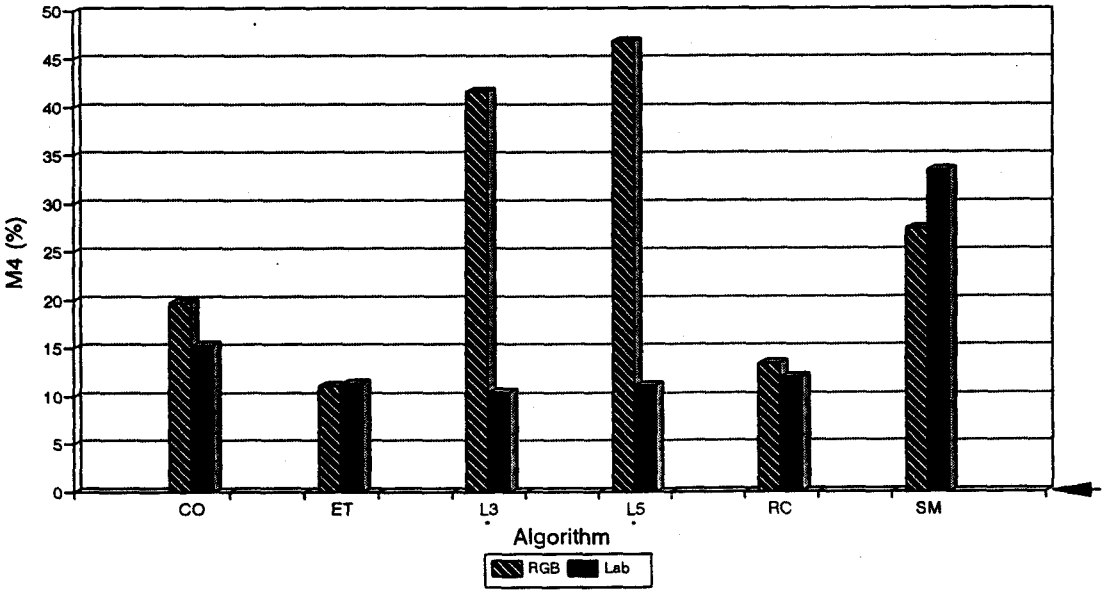

Figure 6.19. Performance results based on the missed percentage (M_4) under the RGB and Lab color spaces for edge detection and region extraction algorithms (arrow points to the ideal value).

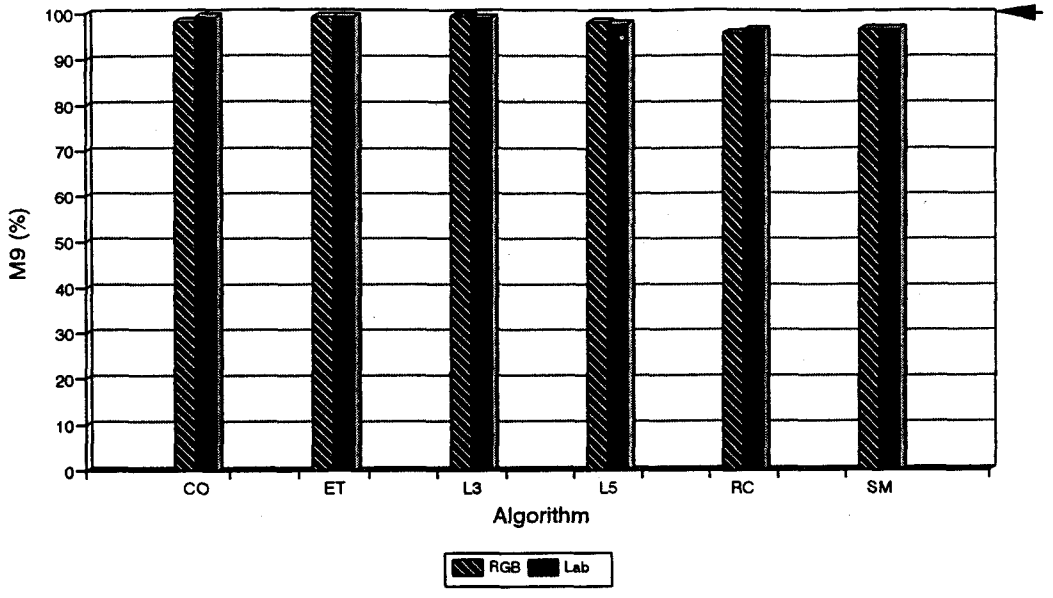

Figure 6.20. Performance results based on the clear wood accuracy (M_9) under the RGB and Lab color spaces for edge detection and region extraction algorithms (arrow points to the ideal value).

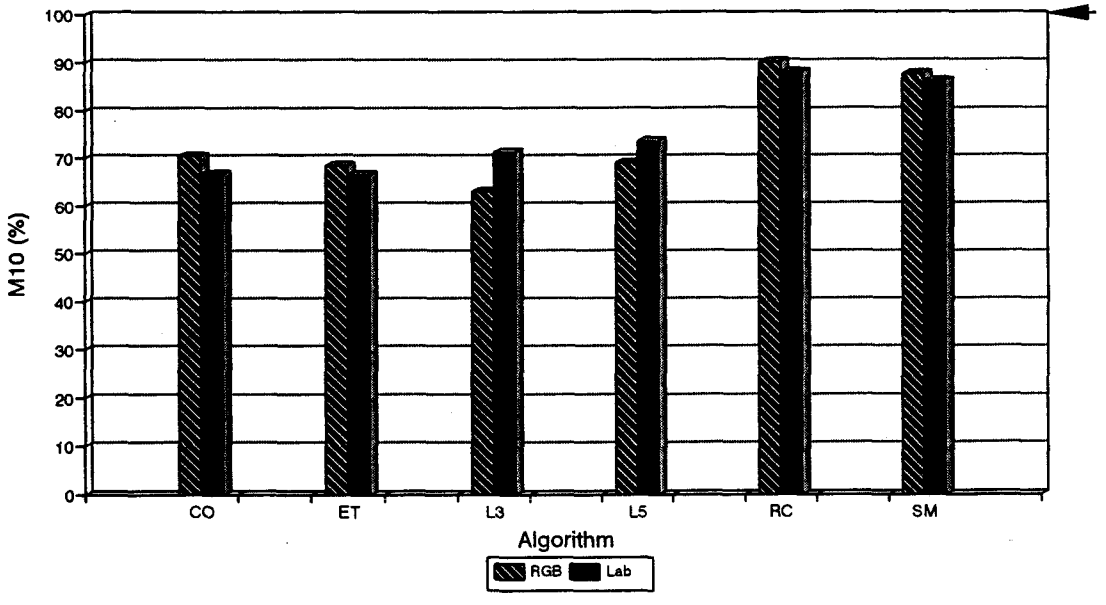

Figure 6.21. Performance results based on the defect accuracy (M_{10}) under the RGB and Lab color spaces for edge detection and region extraction algorithms (arrow points to the ideal value)..

Table 6.15. Performance results based on the count percentage (M_1) using the RGB and Lab color spaces for thresholding algorithms.

	CET	COT	CTR	LET	LOT	LTR
BS	* 358.93 77.7 278.82 78.1	* 328.5 165.9 227.0 239.5	 392.2 211.4 326.0 313.3	* 1792. 763.3 910.0 1060.	* 655.3 218.4 328.6 277.0	* 742.3 287.1 531.1 430.9
CL	* 313.7 195.0 124.3 277.7	* 1039. 525.2 562.2 507.5	 1164. 973.0 1013. 823.3	* 3580. 1526. 1684. 1817.	* 1426. 848.8 717.2 793.3	* 1493 1088 972.5 1104
LK	* 134.4 108.7 100.0 133.3	 266.7 242.5 222.7 125.2	 115.9 143.1 100.0 100.0	* 2423. 1100. 997.8 1066.	 298.9 259.6 254.2 152.5	* 927.5 395.3 314.6 277.1
PP	* 158.0 108.7 127.0 117.7	 182.8 139.7 150.8 143.9	 94.4 94.4 94.4 94.4	* 2335. 748.6 1114. 913.5	 261.2 189.8 168.3 176.1	* 1004 160.1 198.1 316.2
PS	* 504.6 162.9 315.1 388.0	 268.7 232.6 210.1 188.2	 310.4 191.9 237.2 227.8	* 2609. 958.7 937.9 1029.	 322.1 349.0 262.0 207.4	* 445.6 324.4 361.3 900.9
TK	* 189.7 127.0 127.0 122.7	 285.4 278.4 277.4 233.5	 186.3 292.7 156.2 162.4	* 2015. 937.9 1058. 1083.	 321.5 288.2 256.4 295.4	* 961.3 524.4 367.5 354.4
WA	* 269.4 121.1 118.4 107.6	 109.3 118.4 100.6 113.7	 76.7 69.1 92.1 92.1	* 969.3 370.4 430.0 638.8	 167.5 153.2 133.9 143.2	* 408.9 181.1 184.2 170.0

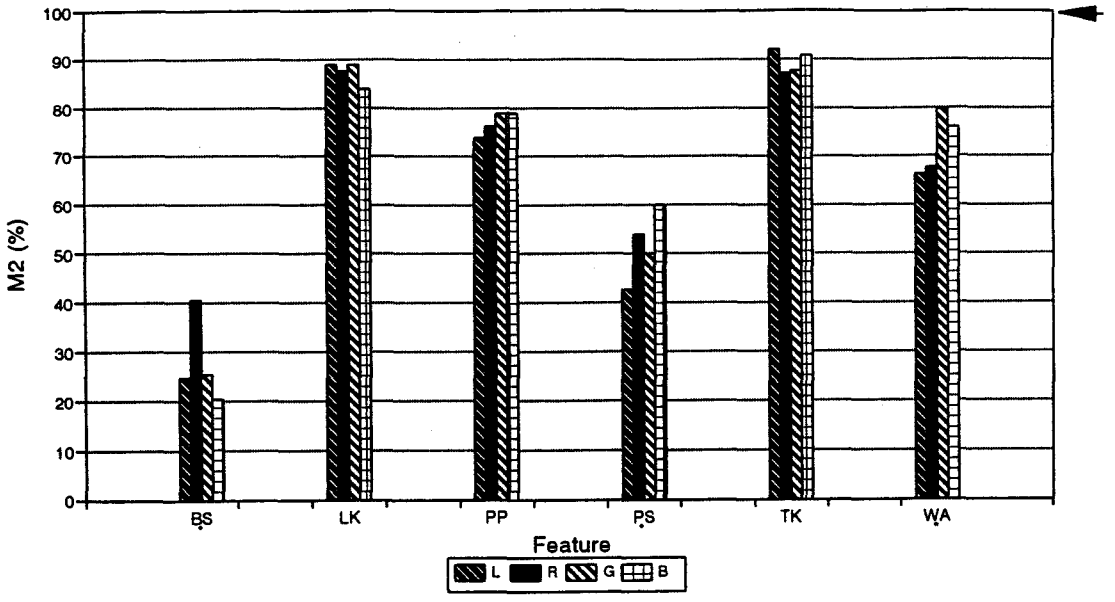

Figure 6.22. Performance results based on the paired percentage (M_2) under the L, R, G, and B color components for thresholding algorithms (arrow points to the ideal value).

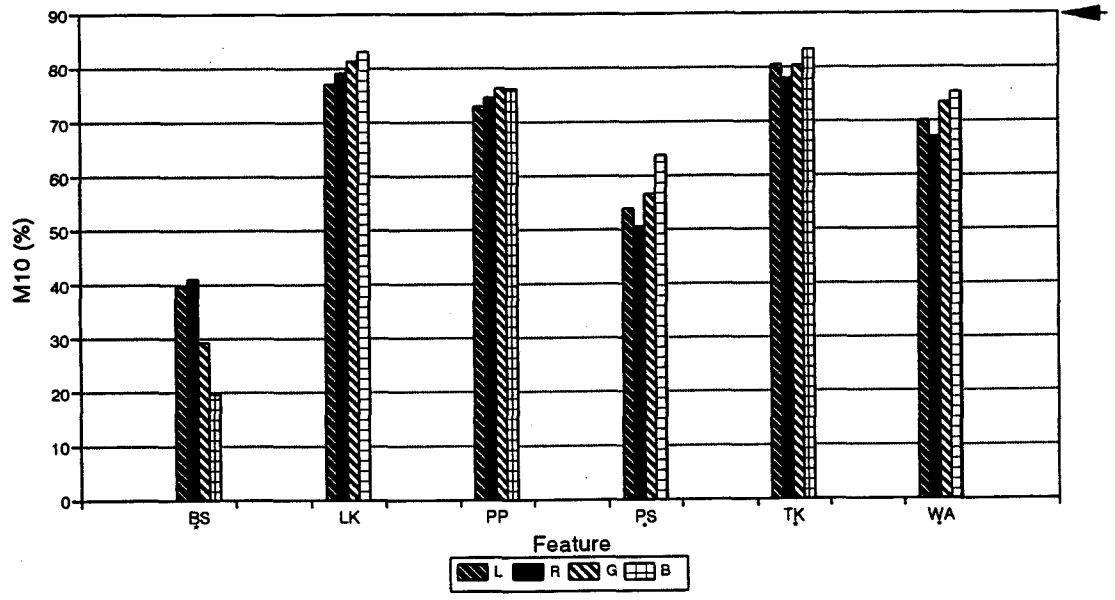

Figure 6.23. Performance results based on the defect accuracy (M_{10}) under the L, R, G, and B color components for thresholding algorithms (arrow points to the ideal value).

the top to the bottom in each cell. Only the feature blue stain for the three global thresholding algorithms was significantly different. Therefore,

(1) L and G were not significantly different for global entropy thresholding, but they were significantly better than R and B; and

(2) L, R, and G were not significantly different for the global Otsu thresholding and transition-matrix thresholding algorithms, but they were significantly better than B;

For the clear wood accuracy measure (M_9), only algorithms and colors showed interactions. Figure 6.24 shows the results for clear wood accuracy (M_9).

Conclusions are:

(1) color components R, G, and B were not significantly different from L for all features for global and local entropy thresholding;

(2) L was significantly worse than R, G or B for the global and local transition-matrix thresholding;

(3) L and R were never significantly different for the global Otsu thresholding (for blue stain, clear wood, and tight knots, L and R were significantly better than G and B); and

(4) L was not significantly different from R for the local Otsu thresholding (they both were significantly better than B for the features blue stain and clear wood).

For the defect accuracy measure (M_{10}), only algorithms and colors showed interactions. Figure 6.23 shows the results for defect accuracy. The color components L and R were not significantly different for pitch streaks, tight knots, and wane, but they were worse than G and B. L was not significantly different from R for blue stain but was significantly better than G and B for blue stain.

Combining all the results leads to the conclusion that the color component L of the Lab color space is not significantly better than R, G, or B of the RGB color space. Therefore, the color space RGB is suggested for the thresholding algorithms if it is native to the camera.

Table 6.16. Performance results based on the mean centroid difference (M_5) using the RGB and Lab color spaces for thresholding algorithms.

	CET	COT	CTR	LET	LOT	LTR
BS	* 3.25 15.28 4.84 59.81	* 9.54 8.79 11.65 23.26	* 15.28 10.20 5.21 29.32	7.52 13.02 30.88 24.98	4.2 11.59 14.3 16.06	19.71 5.2 17.8 22.9
LK	0.84 1.06 1.66 0.94	2.67 3.75 5.41 3.69	1.28 1.18 0.85 1.13	0.76 5.87 7.53 3.48	2.82 1.70 5.35 5.0	7.41 1.14 3.55 2.11
PP	0.53 0.47 0.68 0.64	0.88 4.31 4.29 5.56	0.50 0.33 0.37 0.88	1.82 2.57 2.04 1.87	0.59 0.71 1.97 2.09	2.15 3.94 2.22 2.13
PS	2.66 12.04 3.97 5.25	7.92 9.76 10.81 7.97	11.19 10.94 7.77 3.43	4.28 5.41 6.27 2.83	6.02 9.65 11.12 7.55	3.75 7.73 4.33 6.10
TK	1.60 1.45 1.60 1.81	6.60 5.07 8.31 7.92	1.73 1.91 1.97 3.49	1.17 3.45 2.13 1.45	2.24 3.48 4.84 6.15	4.21 4.35 4.97 1.17
WA	1.75 1.96 2.04 2.25	5.43 5.53 5.57 4.95	1.66 2.33 1.30 1.33	4.09 3.15 1.27 2.92	3.76 4.81 4.57 3.94	9.19 2.11 3.54 5.25

(The values for the color components L, R, G, and B are in order from top to bottom in each cell; an asterisk indicates a statistically significant difference at the 5% level and the difference also meets the rules-of-thumb.)

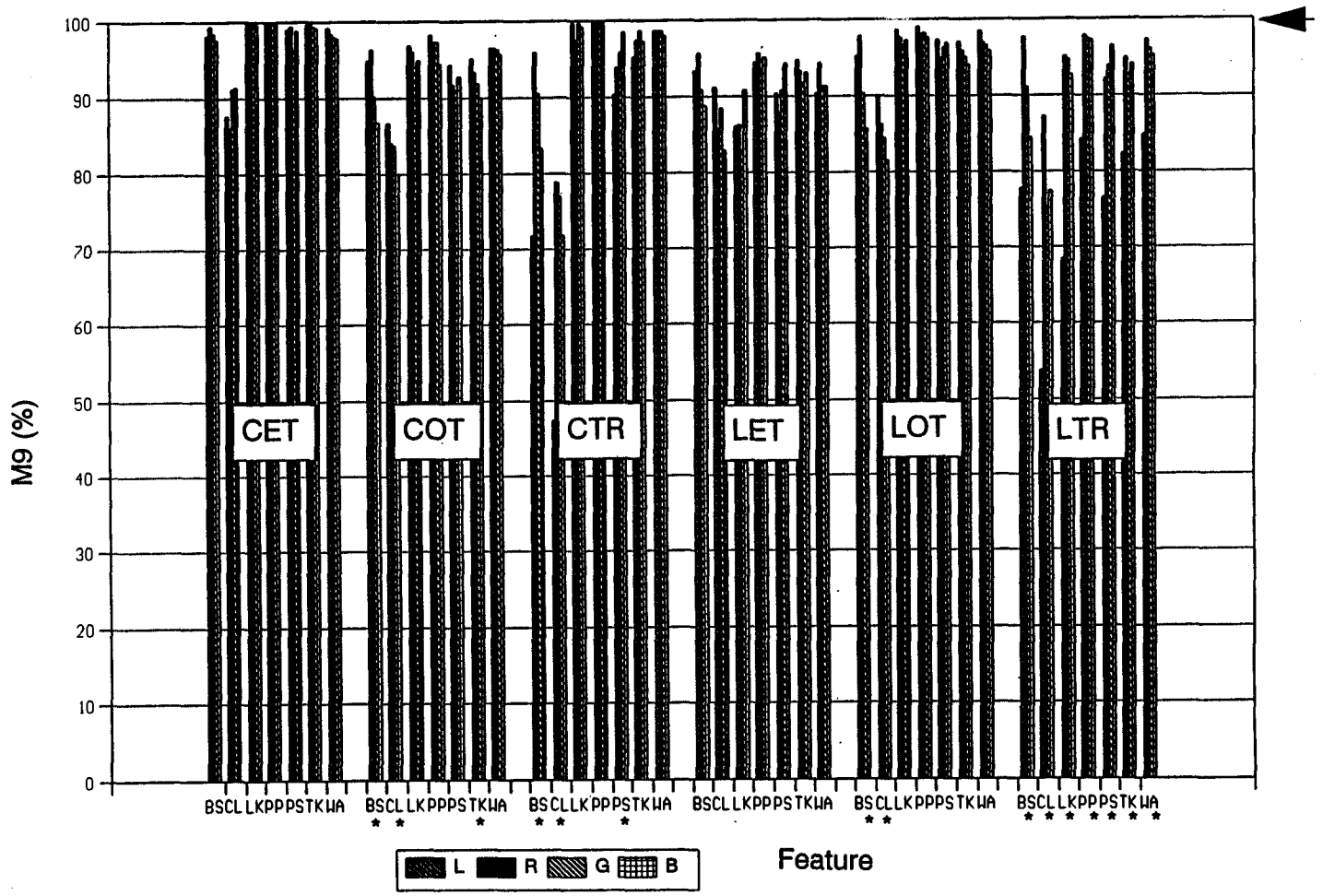

Figure 6.24. Performance results based on the clear wood accuracy (M_9) under the L, R, G, and B color components for thresholding algorithms (arrow points to the ideal value).

6.3.3.2 Comparison of the RGB color space to gray-scale

After comparing the RGB color space to the Lab color space, an analysis between the RGB color space and gray-scale was made for both 5-bit and 8-bit camera resolutions.

6.3.3.2.1 Comparison of the RGB color space to gray-scale for edge detection and region extraction algorithms

For the count percentage (M_1), color showed interactions with both algorithm and feature for the data set of clear wood, loose knots, pitch pockets, tight knots, and wane (CL, LK, PP, TK, and WA). The results are listed in Table 6.17. The significant differences that occurred were:

(1) for the nonlinear Laplace edge detector with a window size of 3, the RGB color space was significantly better than the gray-scale in the features pitch pockets, tight knots, and wane;

(2) for the nonlinear Laplace edge detector with a window size of 5, gray-scale was significantly better than the RGB color space for the feature loose knots; and

(3) for the method combining splitting-and-merging with region growing (SM), gray-scale was significantly better than the RGB color space for the feature clear wood.

In the data set of blue stain and pitch streaks (BS and PS), color showed an interaction with camera resolution, but for both camera resolutions, the results for the RGB color space and gray-scale were not significantly different (for 5-bit: 326% for RGB, 259% for gray; for 8-bit camera resolution: 260% for RGB, 301% for gray).

For the paired percentage measure (M_2) and mean centroid difference (M_5), the colors RGB and gray-scale were not statistically different (for count percentage (M_2), the 74.2% value for RGB and 70.8% value for gray-scale are statistically significantly different but not significantly different under the rules-of-thumb).

Table 6.17. Performance results based on the count percentage (M_1) using the RGB color space and gray scale for edge detection and region extraction algorithms.

	CO	ET	L3	L5	RC	SM
CL	26.9 86.41	1.76 13.17	75.13 60.95	9.77 2.71	282.39 7.02	* 364.6 133.6
LK	126.11 15.9	108.7 120.5	113.2 121.4	* 214.4 126.6	100.0 115.9	100.0 112.3
PP	173.4 214.7	140.3 160.1	* 103.8 344.9	181.8 151.8	127.8 106.4	97.2 99.5
TK	241.7 200.1	127.4 145.2	* 107.6 238.4	188.2 149.9	284.9 188.0	204.4 136.9
WA	306.3 289.8	124.4 139.9	* 214.2 609.8	482.3 503.8	182.7 133.5	127.2 135.5

(The top value in each cell is the one corresponding to the RGB color space and the bottom for the gray-scale, while an asterisk indicates the values in the cell are significantly different.)

For the wrong percentage (M_3), color was interacted with algorithm for the data set of loose knots, pitch pockets, tight knots, and wane (LK, PP, TK, and WA). The results are shown in Figure 6.25. Gray-scale was significantly better than the RGB color space for the algorithms nonlinear Laplace edge detector with a window size of 5 (L5) and the method combining clustering with region growing (RC). For blue stain, gray-scale was significantly better than the RGB color space (BS: 349% for RGB, 94% for gray), while for pitch streaks, the two color spaces were not significantly different (PS: 182% for RGB, 140% for gray-scale).

For the missed percentage (M_4), color showed an interaction with the algorithm factor for the data sets loose knots, pitch pockets, tight knots, and wane (LK, PP, TK, and WA). The results are shown in Figure 6.26. The RGB color space is significantly better than the gray-scale only for the nonlinear Laplace edge detector (both window sizes of 3 and 5). For blue stain, the RGB color space was significantly better than the gray-scale for the two region extraction algorithms and the nonlinear Laplace edge detector with a window size of 5 (Figure 6.27). For pitch streaks, there was no significant difference between the two spaces (Figure 6.28).

For the clear wood accuracy (M_9), color showed interactions with both the algorithm and camera resolution factors for the data sets clear wood, loose knots, pitch pockets, pitch streaks, tight knots, and wane (CL, LK, PP, PS, TK, and WA). The results under the 8-bit camera resolution showed no significant difference between the two spaces (Figure 6.29). As shown in Figure 6.30, the same is true for blue stain.

For defect accuracy (M_{10}), color showed no interactions with other factors for the data sets loose knots, pitch pockets, tight knots, and wane (LK, PP, TK, and WA). The two color spaces were not significantly different (94.6% for the RGB color space and 92.64% for gray-scale; color was significantly different statistically but not significantly different by the rules-of-thumb). In the data set for blue stain, color showed interactions with the algorithm factor, and the results are shown in Figure 6.31. Though the defect accuracies are all low, the RGB color space is significantly better than gray-scale for the two region extraction algorithms and the nonlinear Laplace edge detection algorithm (both window sizes of 3 and 5) but not for the compass or entropy edge detection algorithms. In the data set of pitch streaks, color was not interacted with other factors and the RGB color space was significantly better than gray-scale (49.9% for RGB color space and 40.59% for gray-scale).

Combining all results, the RGB color space was significantly better than gray-scale for defect detection. The only sacrifices were in M_1 and M_3 for clear wood. This coincides with previous research. The color information for some defect types (e.g., blue stain and pitch streaks) is very important for defect accuracy (M_{10}) as

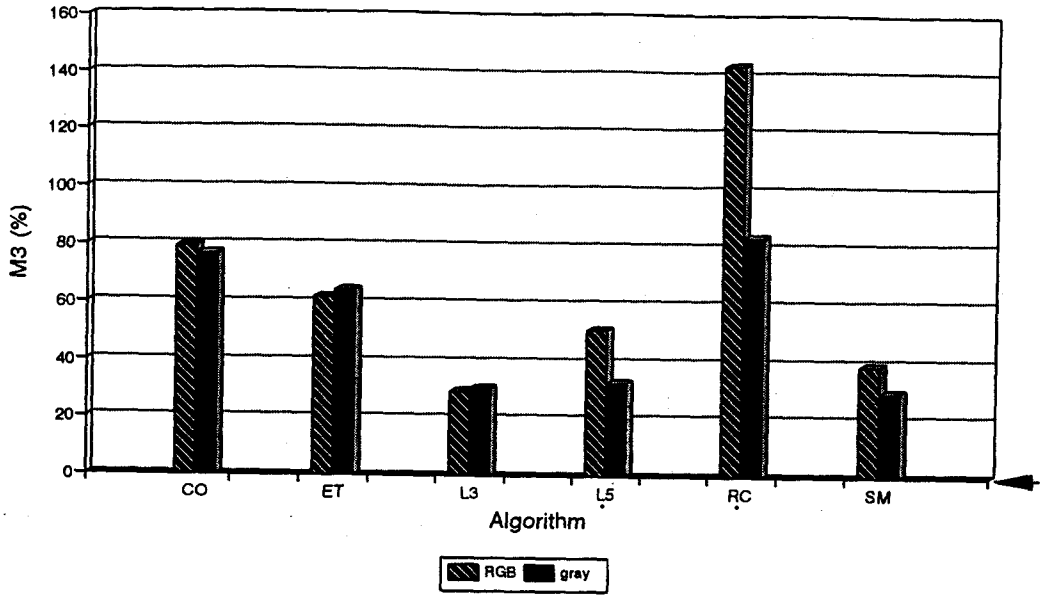

Figure 6.25. Performance results based on the wrong percentage (M_3) using the RGB color space and gray-scale for edge detection and region extraction algorithms (arrow points to the ideal value).

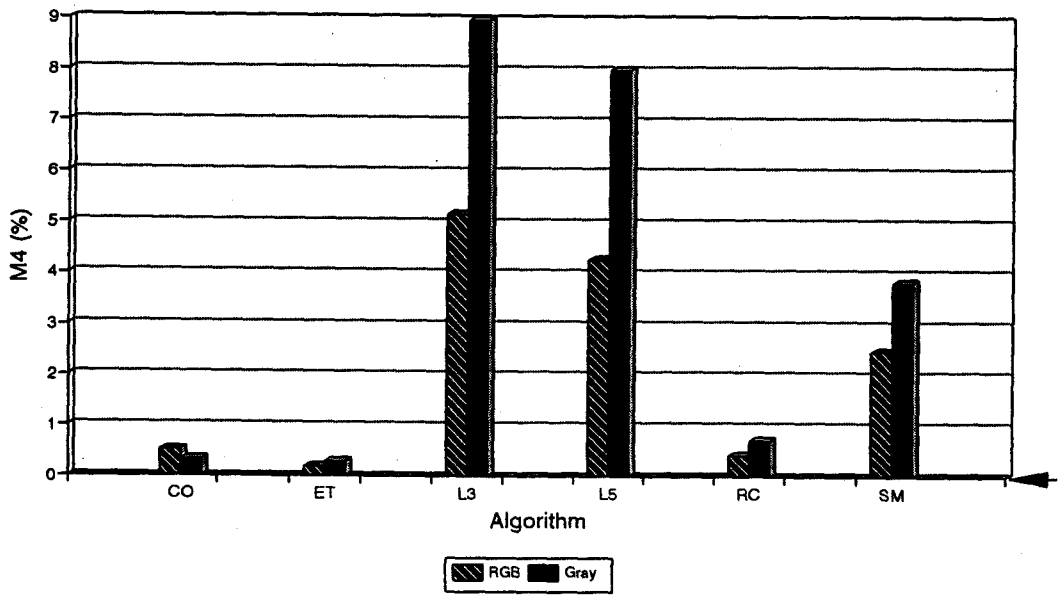

Figure 6.26. Performance results based on the missed percentage (M_4) for the data set of loose knots, pitch pockets, tight knots, and wane using the RGB color space and gray-scale for edge detection and region extraction algorithms (arrow points to the ideal value).

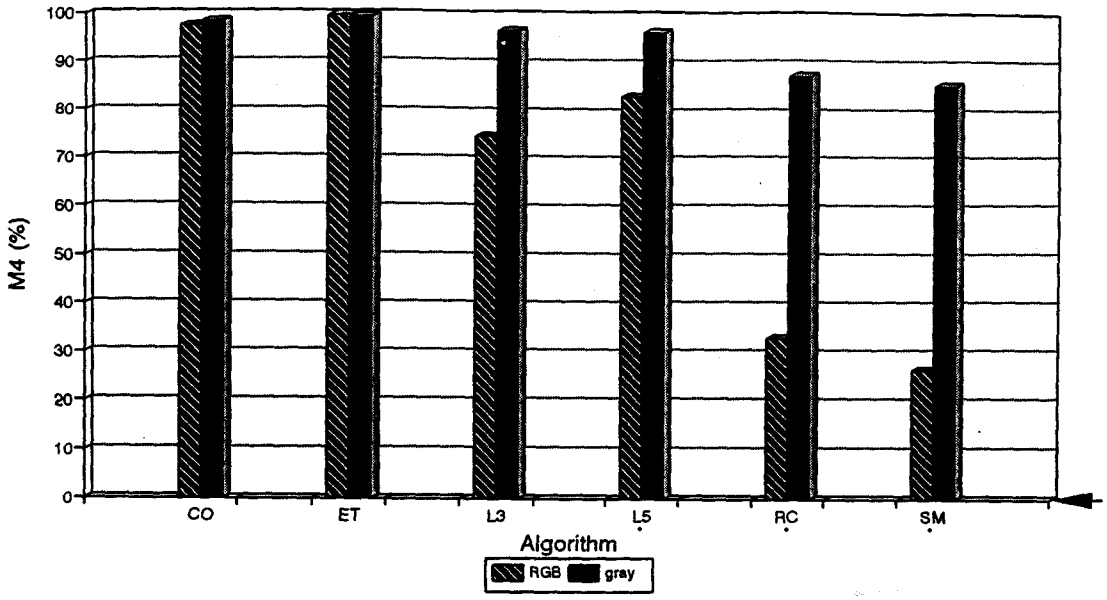

Figure 6.27. Performance results based on the missed percentage (M_4) for blue stain using the RGB color space and gray-scale for edge detection and region extraction algorithms (arrow points to the ideal value).

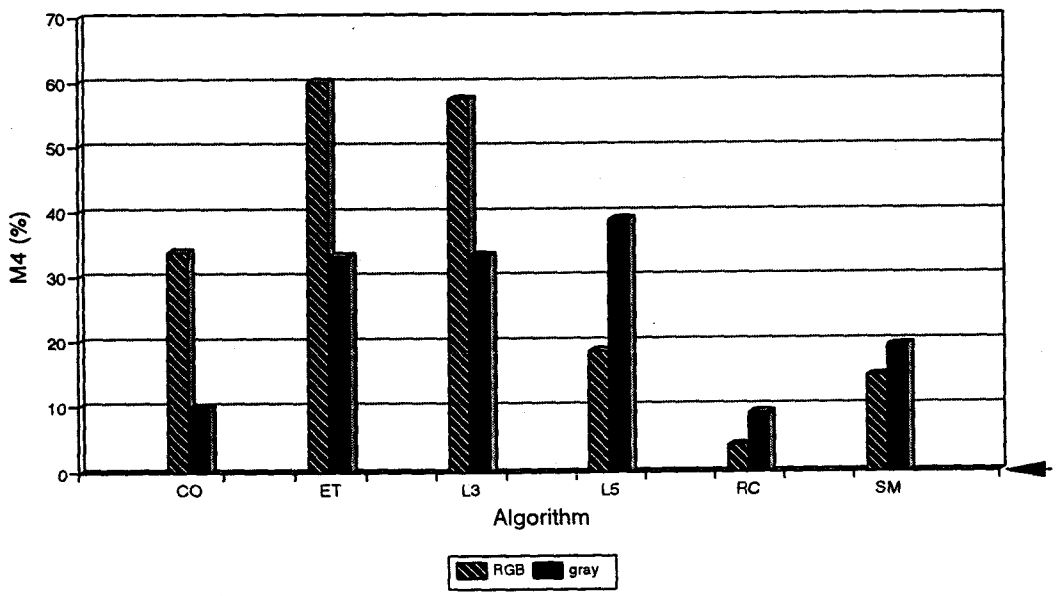

Figure 6.28. Performance results of the missed percentage (M_4) for pitch streaks using the RGB color space and gray-scale for edge detection and region extraction algorithms (arrow points to the ideal value).

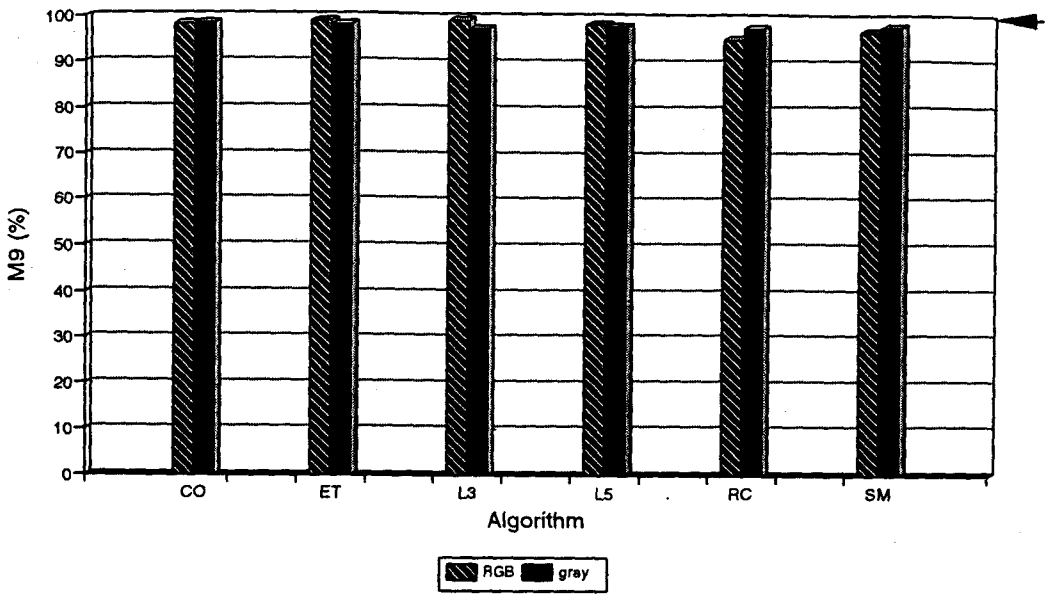

Figure 6.29. Performance results based on the clear wood accuracy (M_9) for the data set of clear wood, loose knots, pitch pockets, pitch streaks, tight knots, and wane using the RGB color space and gray-scale for 8-bit camera resolution for edge detection and region extraction algorithms (arrow points to the ideal value).

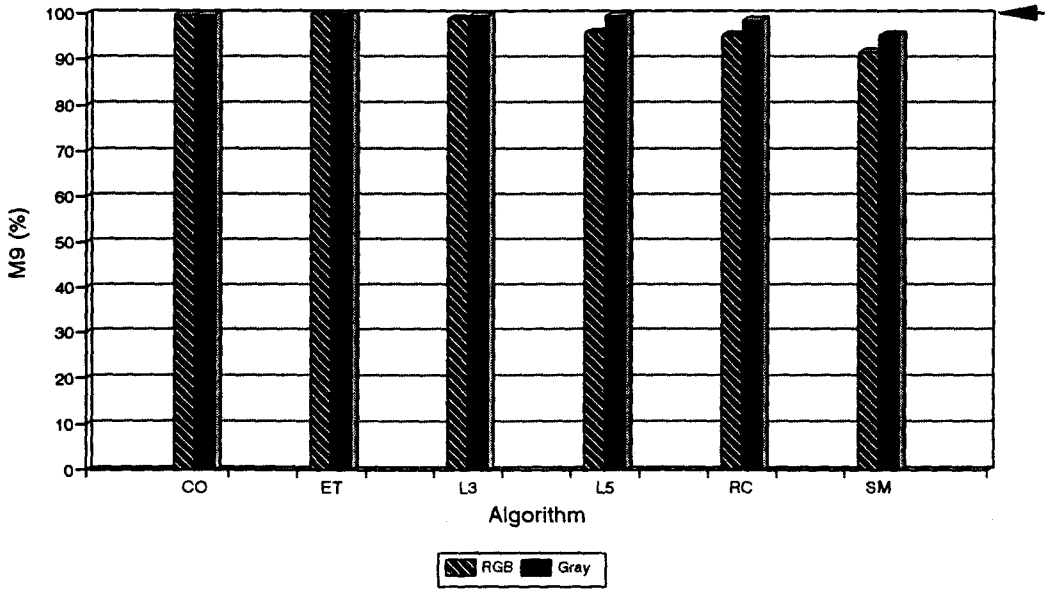

Figure 6.30. Performance results based on the clear wood accuracy (M_9) for blue stain using the RGB color space and gray-scale for edge detection and region extraction algorithms (arrow points to the ideal value).

shown in Figure 6.31 where the RGB color space had larger defect accuracies in blue stain and pitch streaks.

6.3.3.2.2 Comparison of the RGB color space to gray-scale for thresholding algorithms

For the paired percentage measure (M_1), the color components R, G, and B and gray-scale were not significantly different. For the count percentage measure (M_2), the color components R, G, and B and gray-scale were not significantly different for the data sets loose knots, pitch pockets, pitch streaks, tight knots, and wane (LK, PP, PS, TK, and WA). For blue stain (BS), color was significantly different, where the results were 34.12% for R (the best), 15.57% for B (the worst), 24.22% for G, and 24.17% for gray-scale.

For the wrong percentage measure (M_3), the color showed no interactions with other factors and the color components R, G, and B and gray-scale were not significantly different (6.74% for R, 9.14% for G, 9.22% for gray, and 12.73% for B; they were significantly different statistically but did not meet the rules-of-thumb).

For the missed percentage measure (M_4), the color components R, G, and B and gray-scale were not significantly different for the data sets blue stain, pitch pockets, pitch streaks, tight knots, and wane (BS, PP, PS, TK, and WA). For loose knots (LK), color was not significantly different, where the resulting values were 2.98% for B, 3.48% for G, 3.52% for gray-scale, and 4.92% for R (they were significantly different statistically but did not meet the rules-of-thumb).

For the mean centroid difference measure (M_5), the color components R, G, and B and gray-scale were not significantly different for the data sets loose knots, pitch pockets, pitch streaks, tight knots, and wane (LK, PP, PS, TK, and WA). For blue stain (BS), color was significantly different, where R (6.98) and G (7.06) were the best, gray-scale (9.36) was in the middle, and B (12.8) was the worst.

For the clear wood accuracy measure (M_9), the color components R, G, and B and gray-scale were not significantly difference for the data set of clear wood, loose

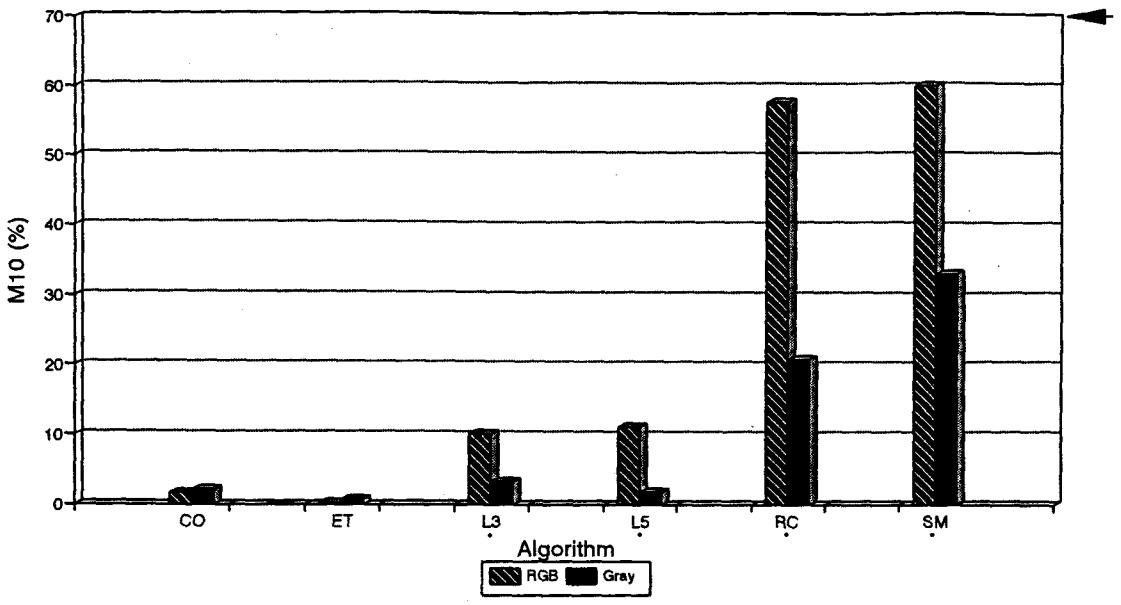

Figure 6.31. Performance results based on the defect accuracy (M_{10}) for blue stain using the RGB color space and gray-scale for edge detection and region extraction algorithms (arrow points to the ideal value).

knots, pitch pockets, pitch streaks, tight knots, and wane (CL, LK, PP, PS, TK, and WA). For blue stain (BS), color had interactions with both algorithm and resolution. The blue stain results are shown in Figure 6.32. Considering only the 8-bit camera resolution, the color components were significantly different for the Otsu thresholding and transition-matrix thresholding algorithms in both their global and local versions. R was the best, B was the worst, and G and gray-scale were similar.

For the defect accuracy measure (M_{10}), the color components R, G, and B and gray-scale were significantly different for the data sets loose knots, pitch pockets, pitch streaks, tight knots, and wane (LK, PP, PS, TK, and WA), where B (75.28%) was the best, gray-scale (72.89%) and G (71.76%) were similar, and R (68.42%) was the worst. For blue stain (BS), color showed no interactions with other factors and color components were significantly different, where R (30.48%) was the best, gray-scale (21.49%) and G (20.08%) were similar, and B (12.75%) was the worst.

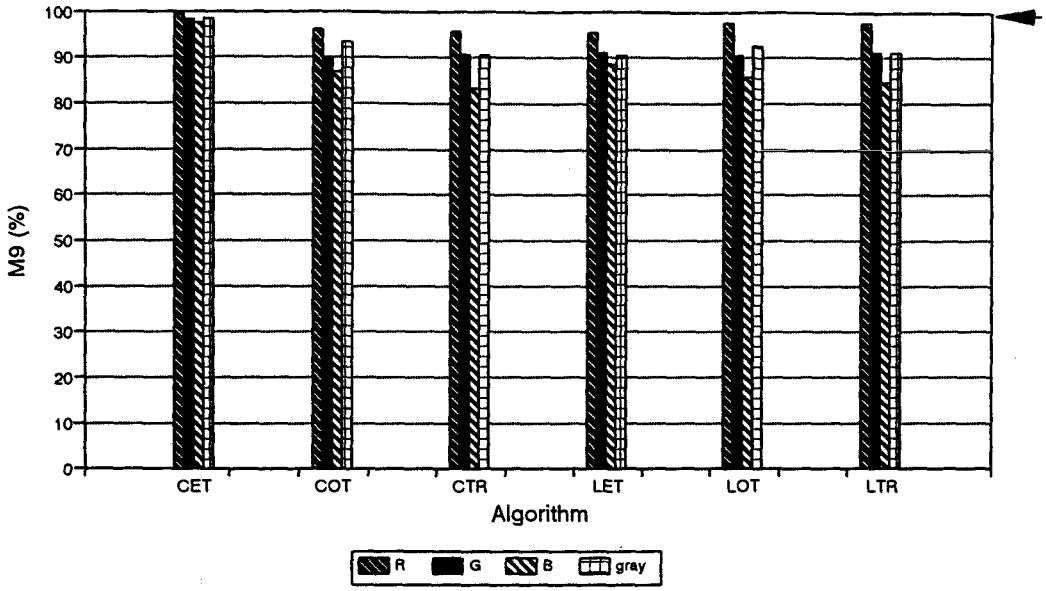

Figure 6.32. Performance results based on the clear wood accuracy (M_9) for blue stain using the R, G, B and gray-scale with 8-bit camera resolution for thresholding algorithms (arrow points to the ideal value).

Considering the results, the RGB color space is suggested for the thresholding algorithms in defect detection. The color information is very important for some defect types, for instance blue stain. The best single color component of the RGB color space for the defect detection is R while the worst single color component for defect detection is B. For some defect types, such as tight knots, loose knots, pitch pockets, and wane, the performances for these six thresholding algorithms were not significantly different under both the RGB color space and gray-scale because the intensity alone can offer enough information for defect detection. One thing that needs to be mentioned involves pitch streak results. For these thresholding algorithms, the results did not show significant difference for finding pitch streaks using the RGB color space or by using gray-scale . The color component B of the RGB color space is the most significant single color component for the detection of some light pitch streaks on veneer surfaces. The reason this did not appear in the experiment with the thresholding algorithms was because either the pitch streak

samples were not representative enough or the information supplied by the primary color component B was not picked up by these six thresholding algorithms.

For all defect types except blue stain, the performances of each color component (R, G, or B) did not show significant differences. However, for blue stain, the R component was significantly better than the other two. Therefore, the R component of the RGB color space is suggested for the thresholding algorithms.

6.3.4 Algorithms Effects

6.3.4.1 Comparison of edge detection algorithms

In this section, the performances of all three edge detection algorithms (compass edge detection, entropy edge detection, nonlinear Laplace edge detection with a window size of 3, and nonlinear Laplace edge detection with a window size of 5) are compared. All comparisons will be made using the 8-bit camera resolution and RGB color space.

For the count percentage measure (M_1), the factor algorithm showed an interaction with features. The results are listed in Table 6.18. The nonlinear Laplace edge detector with a window size of 3 (L3) and the entropy edge detector (ET) were better than the nonlinear Laplace edge detector with a window size of 5 (L5) and the compass edge detector (CO) for all features except clear wood. For clear wood, the performances by algorithms were not significantly different.

For the paired percentage measure (M_2), the algorithms showed no interactions with the features and the algorithms were not significantly different. For the wrong percentage measure (M_3), the algorithm showed no interactions with the features, but the algorithms were significantly different. The nonlinear Laplace edge detector with a window size of 3 (L3) was the best (21.85%), nonlinear Laplace edge detector with a window size of 5 (L5) and the entropy edge detector (ET) were similar (34.38% and 44.58%), and the compass edge detector (CO) was the worst (51.5%).

For the missed percentage measure (M_4), the algorithm showed no interactions with the features and the algorithms were not significantly different. They were significantly different statistically but did not meet the rules-of-thumb (1.04% for the entropy edge detector (ET), 1.94% for the compass edge detector (CO), 5.79% for the nonlinear Laplace edge detector with a window size of 3 (L3), and 6.67% for the nonlinear Laplace edge detector with a window size of 5 (L5)).

For the mean centroid difference measure (M_5), the factor algorithm showed no interactions with the factor feature and the algorithms were not significantly different.

For the clear wood accuracy measure (M_9), the algorithm factor was interacted with the features. The results are listed in Table 6.19 and they were not significantly different (they were significantly different statistically but did not meet the rules-of-thumb). For the defect accuracy measure (M_{10}), the algorithm factor showed no interactions with the features factor and the algorithms were not significantly different.

Combining all results, the three edge detectors performed similarly in finding defects, but the entropy edge detector and nonlinear Laplace edge detector with a window size of 3 are better than the compass edge detector and the nonlinear Laplace edge detector with a window size of 5 because they found fewer false defects.

6.3.4.2 Global thresholding versus local thresholding

In this section, global thresholding algorithms are compared to their corresponding local thresholding versions. All comparisons were made using the 8-bit camera resolution and the color component R of the RGB color space.

For the entropy thresholding algorithms, the algorithm factor showed interactions with the features factor for the count percentage measure (M_1) (Figure 6.33). The asterisk indicates the significantly different pairs are clear wood, loose knots, pitch streaks, and tight knots. The global entropy thresholding algorithm (CET) was significantly better than the local version (LET) in all significantly different pairs. For the paired percentage (M_2), missed percentage (M_4), and clear

Table 6.18. Performance results based on the count percentage (M_1) for edge detection algorithms.

BS	ET	(CO	L3	L5)
	140.72	(846.34	1244.92	1263.77)
CL	(L5	ET	L3	CO)
	(112.11	114.57	165.47	352.98)
LK	(ET	L3)	CO	L5
	(568.73	568.73)	636.02	906.39
PP	L3	(L5	ET)	CO
	509.78	(682.8	696.15)	796.1
PS	L3	(L5	CO)	ET
	870.22	(1231.03	1436.67)	2066.99
TK	(L3	ET	L5)	CO
	(564.22	664.49	714.86)	1180.73
WA	ET	L3	CO	L5
	657.89	920.26	1207.43	1779.22

(The order of the algorithms and corresponding performance values is from the best to the worst and values in the brackets indicate that they are not significantly different.)

Table 6.19. Performance results based on the clear wood accuracy (M_9) for edge detection algorithms.

BS	(ET	CO	L3	L5)
	(99.98	99.62	99.57	96.79)
CL	(ET	L5	CO	L3)
	(99.97	99.84	99.75	98.86)
LK	(L3	ET	L5	CO)
	(99.58	99.21	99.17	99.15)
PP	(L3	L5	ET	CO)
	(99.56	99.02	98.44	97.93)
PS	(L3	ET	L5	CO)
	(99.65	99.15	98.66	97.88)
TK	(L3	ET	L5	CO)
	(99.53	99.18	98.68	98.59)
WA	(L3	ET	CO	L5)
	(98.06	97.49	95.89	94.94)

(The order of the algorithms and corresponding performance values is from the best to the worst and the values in the brackets indicate that they are not significantly different.)

wood accuracy (M_9), the algorithm (global version or local version) showed no interactions with features, and the two algorithms were not significantly different. For the wrong percentage (M_3), the factor algorithm showed interactions with the factor features (Figure 6.34). An asterisk indicates the significantly different pairs (blue stain and pitch streaks). The global entropy thresholding (CET) was significantly better than the local version (LET) in all significantly different pairs. For the mean centroid difference (M_5), the algorithm did not interact with features, and the global version (CET) was significantly better than the local version (LET) (1.23 for CET and 1.77 for LET). For the defect accuracy (M_{10}), the algorithm factor did interact with features (Figure 6.35). An asterisk indicates the significantly different pair for the feature pitch pockets. The global entropy thresholding (CET) was significantly better than the local one (LET) for pitch pockets.

Combining the results, the conclusion can be made that global entropy thresholding is better than local entropy thresholding because it finds fewer false defects and is more capable at finding defects.

For the Otsu thresholding algorithms, the algorithm factor did not interact with the factor features for the count percentage measure (M_1). The global version was significantly better than the local one (192.76% for COT and 770.76% for LET). The algorithms were also not significantly different for the measures paired percentage, missed percentage, mean centroid difference, and defect accuracy (M_2, M_4, M_5, and M_{10}). For the measure wrong percentage (M_3), the algorithm did not interact with the factor features and the global version was significantly better than the local one (8.06% for COT and 19.66% for LET). For the clear wood accuracy measure (M_9), the algorithm did not interact with features and two versions were not significantly different (94.52% for COT and 19.66% for LET) (they were significantly different statistically but did not meet the rules-of-thumb).

Combining all results together, the conclusion can be made that the global Otsu thresholding algorithm is better than the local entropy thresholding algorithm because it finds fewer false defects with similar defect detection ability.

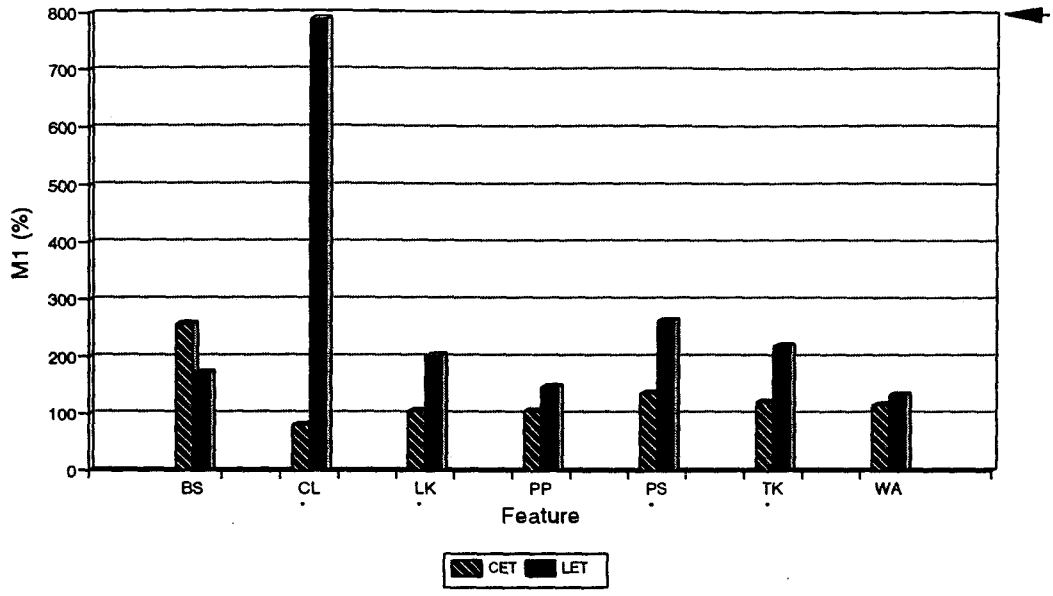

Figure 6.33. Performance results based on the count percentage (M_1) for global and local entropy thresholding algorithms (arrow points to the ideal value).

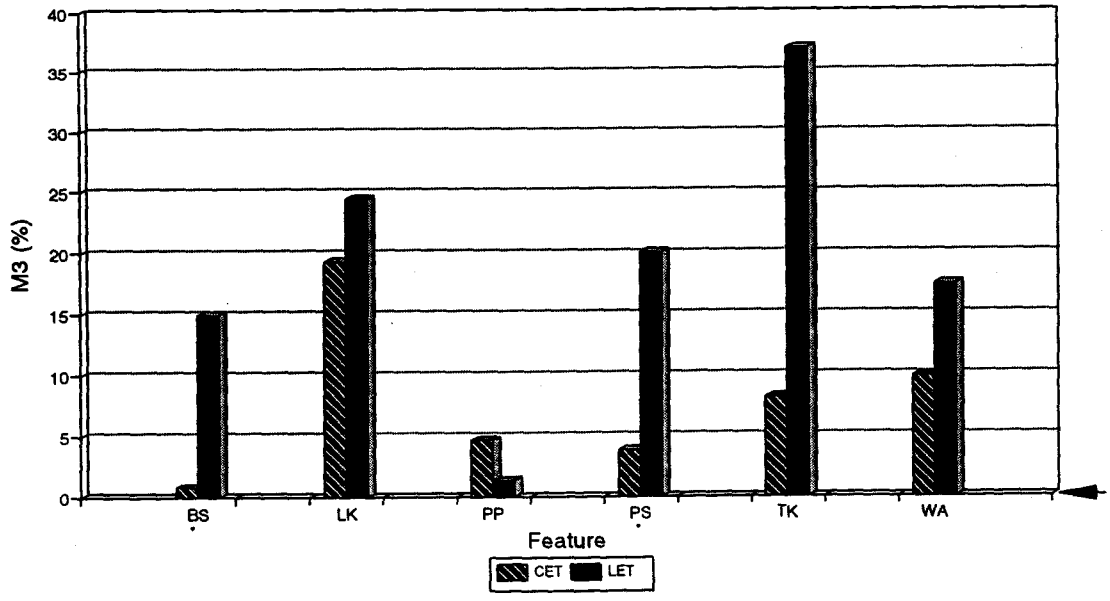

Figure 6.34. Performance results based on wrong percentage (M_3) for global and local entropy thresholding algorithms (arrow points to the ideal value).

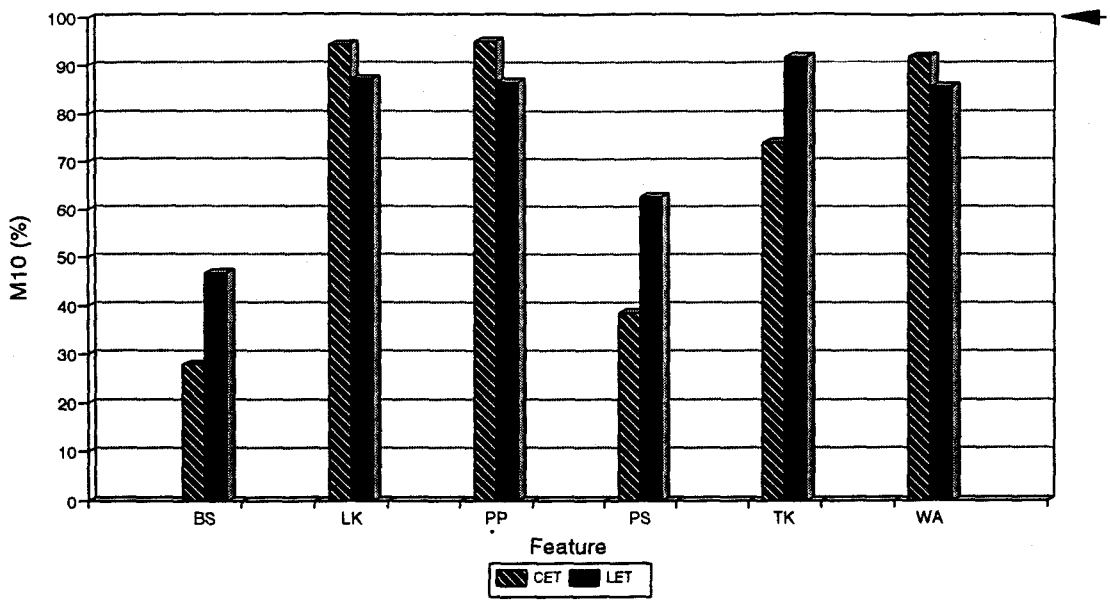

Figure 6.35. Performance results based on the defect accuracy (M_{10}) for global and local entropy thresholding algorithms (arrow points to the ideal value).

For the transition-matrix thresholding algorithms, the algorithms factor did not interact with the features factor for the count percentage measure (M_1), and the local version was significantly better than the global one (402.88% for LTR and 885.19% for CTR). The algorithms factor did not interact with the features factor and the two algorithms were not significantly different for the measures paired percentage, wrong percentage, missed percentage, mean centroid difference, and defect accuracy (M_2, M_3, M_4, M_5, and M_{10}). For the clear wood accuracy measure (M_9), the algorithms factor did not interact with the features factor, and the local version was significantly better than the global one (94.66% for LTR and 91.19% for CTR).

Combining all results, the conclusion can be made that the local transition-matrix thresholding is better than the global transition-matrix thresholding because it finds fewer false defects and has greater capability for defect detection.

6.3.4.3 Comparison of global entropy thresholding, global Otsu thresholding and local transition-matrix thresholding algorithms

The better thresholding algorithms found in the last section, namely global entropy thresholding, global Otsu thresholding, and local transition-matrix thresholding will be compared in this section. The 8-bit camera resolution and R color component of the RGB color space are still used.

For the count percentage measure (M_1), the algorithms factor did not interact with the features factor (Table 6.20). Global entropy thresholding was the best for all features, local transition-matrix thresholding was the worst for all features except wane, and the global Otsu thresholding was in the middle for all features except wane. For the measures M_2, M_3, M_4, M_5, and M_9, the algorithms factor did not interact with the features factor and the three algorithms were not significantly different. For the defect accuracy measure (M_{10}), the algorithms factor did not interact with the features factor and the results are shown in Figure 6.36. The only significant differences were in the features blue stain and wane. For blue stain, the global Otsu and the local transition-matrix thresholding algorithms were similar and significantly better than the global entropy thresholding algorithm. For wane, the global entropy was the best and the global Otsu was the worst (91.9% for CET, 75.35% for LTR, and 65.12 for COT).

Combining all results, the conclusion can be made that global entropy thresholding is the best because it found the fewest number of false defects and detected wane with the highest defect accuracy. Although the global entropy thresholding algorithm was the worst in finding blue stain, the other two algorithms only found blue stain with less than 62% defect accuracy. The local transition-matrix thresholding algorithm found too many false defects (with the highest count percentage for almost all features) but resulted in a 10% higher defect accuracy in wane than the global Otsu thresholding algorithm. The comparison between these two algorithms becomes application dependent. If the defect wane is an important concern, then local transition-matrix thresholding may be a better choice. However, if the number of

Table 6.20. Performance results based on the count percentage (M_1) for global entropy, global Otsu, and local transition-matrix thresholding algorithms.

BS	(COT	LTR	CET)
	(174.35	214.49	258.28)
CL	CET	(COT	LTR)
	81.26	(406.33	1027.4)
LK	(CET	COT)	LTR
	(106.9	116.9)	275.45
PP	(CET	COT	LTR)
	(106.9	93.55	128.57)
PS	(COT	CET)	LTR
	(127.25	138.1)	304.63
TK	(CET	COT)	LTR
	(122.22	152.84)	337.1
WA	(CET	LTR)	COT
	(117.62	160.06)	66.67

(The order of the algorithms and corresponding performance values is from the best to the worst and the values in the brackets indicate that they are not significantly different.)

false defects detected is of great concern, then the global Otsu thresholding algorithm is suggested.

6.3.4.4 Comparison of some prechosen algorithms

In this section, some prechosen algorithms will be compared. These algorithms are:

(1) the entropy edge detection,

(2) nonlinear Laplace edge detection with a window size of 3,

(3) global entropy thresholding,

(4) global Otsu thresholding,

(5) local transition-matrix thresholding,

(6) a method combining splitting-and-merging with region growing, and

(7) a method combining clustering with region growing.

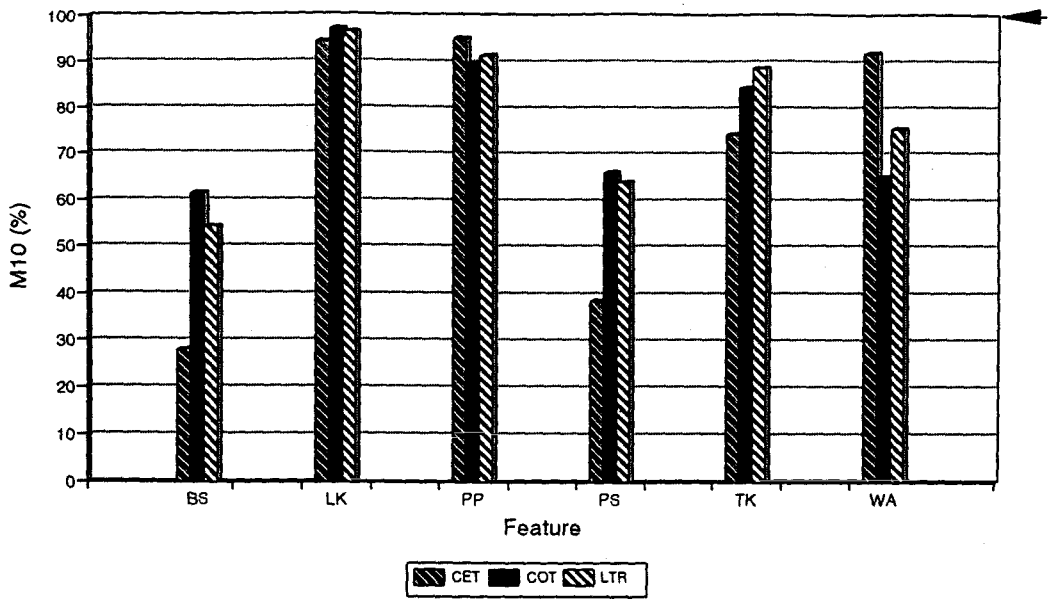

Figure 6.36. Performance results based on the defect accuracy (M_{10}) for global entropy, global Otsu and local transition-matrix thresholding algorithms (arrow points to the ideal value).

These seven algorithms will be compared together by fixing camera resolution at a 8-bits and the color space as RGB for the edge detection and region extraction algorithms and the single R color component of the RGB color space for the thresholding algorithms. The results of all measures (M_1, M_2, M_3, M_4, M_5, M_9, and M_{10}) are listed in the Tables 6.21 to 6.27, respectively, and the average values of all features for the seven performance measures are listed in Table 6.28.

The entropy edge detection algorithm had a low ability to detect defects, which resulted in low values for the paired percentage (M_2), missed percentage (M_4), mean centroid difference (M_5), and defect accuracy (M_{10}) for the features blue stain and pitch streak. Conversely, it resulted in higher values of wrong percentage (M_3) and clear wood accuracy (M_9).

The nonlinear Laplace edge detection with a window size of 3 performed with low ability to detect defects, resulting in low values of the paired percentage (M_2), missed percentage (M_4), mean centroid difference (M_5), and defect accuracy (M_{10}) for

the features blue stain, pitch streak, and wane. It also resulted in higher values of wrong percentage (M_3) and clear wood accuracy (M_9). Its performance was similar to the entropy edge detector with the only difference coming in the performance for wane.

The ability to detect defects with the global entropy thresholding algorithm ranged in the middle of the seven algorithms but resulted in good values of wrong percentage (M_3) and clear wood accuracy (M_9). It performed the worst in the mean centroid difference (M_5) as compared to the other six algorithms.

The global Otsu thresholding algorithm showed good results for the measures paired percentage (M_2) and missed percentage (M_4). Its performance ranged in the middle among the seven algorithms for the wrong percentage (M_3) and the mean centroid difference (M_5). It did not perform well in count percentage (M_1) for clear wood and wane, and it had a low clear wood accuracy (M_9) for clear wood samples. In defect accuracy (M_{10}), it performed well on blue stain and loose knots but did poorly on pitch pockets and tight knots and was the worst on wane. The global Otsu thresholding has the tendency to find relatively large numbers of false defects, especially for tight knots and wane.

The performances of the local transition-matrix thresholding algorithm for all measures except count percentage (M_1) ranged in the middle among the seven algorithms. It was the worst in the count percentage (M_1) for the features clear wood, loose knots, pitch pockets, pitch streaks, and tight knots, especially for clear wood where it found extremely large numbers of false defects.

The method combining clustering with region growing had one of the highest defect detection capabilities. it performed very well for paired percentage (M_2), missed percentage (M_4), and defect accuracy (M_{10}). However, it was the worst in wrong percentage (M_3) and the worst in clear wood accuracy (M_9) for the features pitch streaks, tight knots, and wane. It also performed the worst in the mean centroid difference (M_5) for the features loose knots, pitch pockets, and wane. This algorithm has the tendency to overestimate the defect area (M_3) on the features loose knots, pitch pockets, tight knots, and wane. However, it is the only one that had an overall defect

accuracy over 90% in conjunction with an overall clear wood accuracy of 95.36% (Table 6.28).

The method combining splitting-and-merging with region growing had the second highest defect detection capability. it performed very well for paired percentage (M_2), missed percentage (M_4), and defect accuracy (M_{10}). It was in the middle for count percentage (M_1), wrong percentage (M_3), and mean centroid difference (M_5). It had the lowest clear wood accuracy (M_9) for the feature blue stain, but it ranged in the middle for the other features. Considering all performance measures, this method is the best one among all seven algorithms because it had the highest defect accuracy on blue stain and was second highest on pitch streaks; it had only one worst case in clear wood accuracy for blue stain (92.69%). It had an overall defect accuracy of 87.72% and clear wood accuracy of 96.33%.

The two edge detection algorithms could only find defects with distinct boundaries, such as loose knots, pitch pockets, and wane. It could not find defects whose boundaries had big transition zones, such as blue stain, pitch streaks, and some tight knots. The two region extraction algorithms were superior in finding defects among the three segmentation categories and their parameters can be easily changed to monitor their performances. The three thresholding algorithms are better than the edge detection algorithms since they can find some of the difficult defect types (blue stain and pitch streaks), but there is no simple way to improve their performances like the region extraction algorithms.

In conclusion, the overall order of performance of the seven algorithms is:

First: the method combining splitting-and-merging with region growing;

Second: the method combining clustering with region growing;

Third: global entropy thresholding, global Otsu thresholding, and local transition-matrix thresholding; and

Fourth: entropy edge detection and nonlinear Laplace edge detection with a window size of 3.

Table 6.21. Performance results based on the count percentage (M_1) for seven algorithms.

BS	(COT	LTR	L3	SM	CET	RC)	ET
	(198.11	259.27	260.85	286.13	304.67	330.85)	19.96
CL	(ET	L3)	CET	(RC	SM)	(COT	LTR)
	(14.87	39.54)	165.3	(231.71	268.44)	(797.64	1068.84)
LK	(SM	RC	CET	L3	ET	COT)	LTR
	(99.99	99.99	108.28	108.28	108.28	132.47)	352.18
PP	(SM	CET	ET	RC)	(COT	L3)	LTR
	(99.99	108.28	137.71	140.51)	(94.33	94.33)	149.15
PS	(CET	COT	L3)	(ET	SM	RC)	LTR
	(154.77	172.02	176.3)	(247.81	265.25	289.23)	349.31
TK	L3	(CET	ET	SM	COT)	RC	LTR
	106.93	(125.87	130.97	187.37	222.01)	270.25	460.92
WA	(CET	SM	ET)	(LTR	RC	L3)	COT
	(120.27	125.58	129.38)	(174.4	182.38	190.78)	74.11

(The order of the algorithms and corresponding performance values is from the best to the worst and the values in the brackets indicate that they are not significantly different.)

Table 6.22. Performance results based on the paired percentage (M_2) for seven algorithms.

BS	(COT	LTR	RC	SM)	CET	(ET	L3)
	(65.83	55	52.5	44.17)	39.17	(8.33	5.83)
LK	(CET	COT	ET	L3	SM	RC	LTR)
	(100	100	100	100	100	100	90.0)
PP	(CET	COT	ET	L3	LTR	RC	SM)
	(90	90	90	90	90	90	90)
PS	(SM	RC	COT)	(CET	LTR)	L3	ET
	(90	90	80)	(60	60)	30	20
TK	(LTR	COT	ET	L3	SM	RC	CET)
	(95	95	95	95	95	90	90)
WA	(CET	RC	SM	L3	ET	LTR	COT)
	(90	90	90	80	80	73.33	70)

(The order of the algorithms and corresponding performance values is from the best to the worst and the values in the brackets indicate that they are not significantly different.)

Table 6.23. Performance results based on the wrong percentage (M_3) for seven algorithms.

BS	(ET	CET	RC	SM	LTR	L3	COT)
	(0.2	0.77	2.42	3.53	6.65	9.45	17.09)
LK	(CET	COT	LTR)	(L3	SM)	(ET	RC)
	(19.34	22.15	32.96)	(61.7	78.42)	(114.28	232.71)
PP	(COT	CET	LTR)	(L3	SM)	ET	RC
	(1.07	4.69	8.35)	(13.77	15.02)	48.97	122.11
PS	(CET	ET	L3	LTR	SM	COT)	RC
	(3.87	6.04	6.31	8.48	10.77	12.96)	40.17
TK	(CET	COT)	(L3	LTR	SM	ET)	RC
	(8.21	8.34)	(33.04	47.99	64.15	72.66)	211.42
WA	(COT	LTR	CET	L3)	(SM	ET	RC)
	(3.16	6.4	9.88	10.99)	(19.5	26.99	66.76)

(The order of the algorithms and corresponding performance values is from the best to the worst and the values in the brackets indicate that they are not significantly different.)

Table 6.24. Performance results based on the missed percentage (M_4) for seven algorithms.

BS	(COT	SM	RC)	(LTR	CET)	(ET	L3)
	(17.29	18.71	27.95)	(41.09	47.78)	(99.78	101.95)
LK	(ET	RC	LTR	L3	SM	CET	COT)
	(0.0	0.12	0.52	0.6	1.34	1.39	1.41)
PP	(ET	RC	CET	L3	SM	LTR	COT)
	(0.03	0.62	3.23	4.84	5.27	5.96	7.6)
PS	RC	(SM	LTR	COT)	(CET	ET	L3)
	2.66	(12.01	14.82	18.91)	(26.81	57.33	64.52)
TK	(ET	RC	SM	L3	LTR	COT)	CET
	(0.49	1.05	1.69	4.79	6.54	8.12)	12.45
WA	(RC	ET	SM	CET	COT	LTR)	L3
	(0.68	1.58	2.77	3.44	8.85	9.74)	12.02

(The order of the algorithms and corresponding performance values is from the best to the worst and the values in the brackets indicate that they are not significantly different.)

Table 6.25. Performance results based on the mean centroid difference (M_5) for seven algorithms.

BS	(SM	RC	COT	CET	LTR	L3)	ET
	(7.17	7.25	8.93	11.07	12.27	16.32)	50.02
LK	(ET	CET	COT	SM	LTR	L3)	RC
	(0.76	0.8	0.88	0.88	1	1.07)	1.92
PP	(COT	CET	ET)	(SM	L3)	LTR	RC
	(0.32	0.45	0.56)	(0.73	0.94)	1.54	2.16
PS	RC	(L3	SM	LTR	COT	ET)	CET
	2.81	(4.32	4.95	5.35	7.01	8.44)	9.29
TK	(CET	ET	L3	COT	SM	RC)	LTR
	(1.23	1.29	1.33	1.38	1.59	1.98)	2.95
WA	(LTR	CET	L3	ET	COT	SM	RC)
	(1.58	1.72	1.76	1.76	1.87	2.93	3.87)

(The order of the algorithms and corresponding performance values is from the best to the worst and the values in the brackets indicate that they are not significantly different.)

Table 6.26. Performance results based on the clear wood accuracy (M_9) for seven algorithms.

BS	(ET	L3	CET	LTR	RC	COT)	SM
	(99.98	99.57	99.41	97.64	96.07	95.71)	92.69
CL	(ET	L3	RC	SM)	(LTR	CET)	COT
	(99.97	98.86	98.3	95.09)	(87.3	86.0)	78.77
LK	(CET	L3	ET	SM	COT	RC	LTR)
	(99.83	99.58	99.21	98.87	97.42	97.29	95.72)
PP	(COT	CET	L3	SM	ET	LTR	RC)
	(99.86	99.68	99.56	99.06	98.44	97.84	95.94)
PS	(L3	CET	ET	SM)	(COT	RC	LTR)
	(99.65	99.33	99.15	95.25)	(93.83	93.13	92.15)
TK	(CET	L3	ET	COT	SM)	(LTR	RC)
	(99.63	99.53	99.18	97.45	96.94)	(94.83	94.35)
WA	(COT	CET	L3	ET	LTR	SM)	RC
	(98.63	98.25	98.06	97.49	97.14	96.45)	92.42

(The order of the algorithms and corresponding performance values is from the best to the worst and the values in the brackets indicate that they are not significantly different.)

Table 6.27. Performance results based on the defect accuracy (M_{10}) for seven algorithms.

BS	(SM	COT	RC	LTR)	CET	(L3	ET)
	(61.91	61.52	56.28	54.28)	27.99	(6.23	0.22)
LK	(ET	RC	L3	SM	COT	LTR	CET)
	(100	99.79	99.11	97.81	97.02	96.36	94.32)
PP	(ET	RC	CET)	(SM	L3)	(LTR	COT)
	(99.97	98.53	94.99)	(93.28	92.63)	(91.39	86.64)
PS	RC	SM	(COT	LTR)	(CET	ET	L3)
	91.06	82.43	(66.15	63.92)	(38.59	24.49	17.29)
TK	(ET	RC)	(SM	L3	LTR)	COT	CET
	(98.99	96.9)	(95.41	94.5	88.7)	84.38	74.19
WA	RC	(SM	CET)	ET	(LTR	L3	COT)
	98.52	(95.45	91.9)	87.2	(75.35	68.33	65.12)

(The order of the algorithms and corresponding performance values is from the best to the worst and the values in the brackets indicate that they are not significantly different.)

Table 6.28. Performance results for seven measures for seven algorithms for all seven features.

M1	ET	L3	CET	SM	RC	COT	LTR
	128.81	166.79	195.03	255.6	283.57	353.21	487.14
M2	RC	SM	COT	CET	LTR	L3	ET
	85.42	84.86	83.47	78.19	77.22	66.81	65.56
M3	CET	L3	COT	SM	ET	LTR	RC
	13.73	42.64	61.55	63.96	84.75	153.45	155.87
M4	RC	ET	SM	LTR	COT	L3	CET
	7.59	9.26	10.21	14.23	15.65	17.86	18.91
M5	L3	SM	COT	RC	ET	LTR	CET
	2.98	4.18	4.22	4.35	4.49	4.87	4.96
M9	L3	ET	CET	SM	RC	LTR	COT
	99.26	99.06	97.45	96.33	95.36	94.66	94.52
M10	RC	SM	LTR	COT	CET	ET	L3
	90.18	87.72	78.33	77.3	70.33	68.48	63.01

(The order of the algorithms and corresponding performance values is from the best to the worst.)

CHAPTER 7 CONCLUSIONS

For defect detection on Douglas-fir (Pseudotsuga menziesii) veneer surfaces, three existing edge detection algorithms (compass edge detection, entropy edge detection, and nonlinear Laplace edge detection) were adapted and improved and four existing automatic thresholding algorithms (entropy, Otsu, moment-preserving, and transition-matrix) were adapted in both global and local thresholding versions. Two region extraction algorithms were developed (a method combining splitting-and-merging with region growing and a method combining clustering with region growing). The performances of these algorithms under the combinations of 5-bit and 8-bit camera resolutions and three different color spaces (the RGB color space, the Lab color space, and gray-scale) were compared by multi-factor analysis of variance (ANOVA). The best color space or color component (for thresholding) and best camera resolution for each algorithm were obtained and the best algorithms for defect detection on veneer surfaces were identified. As expected, the two region extraction algorithms were superior in defect detection (especially for difficult defect types such as blue stain and pitch streak) and also their parameters can be easily changed to monitor their performances. The edge detection algorithms could only find defects having large degrees of abrupt changes compared to the background, that is, loose knots, pitch pockets, wane, and some tight knots. The thresholding algorithms were better than the edge detection algorithms in that they could not only find knots, pitch pockets, and wane but also find some blue stain and pitch streaks.

7.1 CONCLUSIONS REGARDING THE COLOR SPACES

1. For the three edge detection and two region extraction algorithms, the RGB color space contributed to better defect detection capabilities than the Lab color space, especially, for blue stain and pitch streaks. However, the use of the RGB color space resulted in finding more false defects.

2. For the three edge detection and two region extraction algorithms, the RGB color space was significantly better than gray-scale for defect detection but it also resulted in a higher probability of finding more false defect.

3. For the three thresholding algorithms (entropy, Otsu, and transition-matrix thresholding with both global and local versions), the color component L of the Lab color space was not significantly better than R, G, or B of the RGB color space.

4. For all defect types except blue stain, the performances of each color component R, G, and B of the RGB color space did not show significant differences. For blue stain, the color component R was significantly better than the other two. Therefore, the R component of the RGB color space is suggested for the three thresholding algorithms (entropy, Otsu, and transition-matrix).

7.2 CONCLUSIONS REGARDING CAMERA RESOLUTIONS

1. For three edge detection and two region extraction algorithms, the two camera resolutions did not make any significant difference for all combinations of algorithms, colors, and veneer surface features. However, when considering computer speed, the 5-bit camera resolution would be more appropriate.

2. For the global entropy thresholding, global transition-matrix thresholding, local Otsu thresholding, and local transition-matrix thresholding algorithms, the 8-bit camera resolution was better than the 5-bit camera resolution. For the local entropy thresholding algorithm, if the count percentage is a key point, then a 5-bit camera resolution should be used. If the concern is more about defect accuracy, then the 8-bit camera resolution is more appropriate. For the global Otsu thresholding, a 5-bit camera resolution is significantly better.

7.3 CONCLUSIONS REGARDING FEATURES ON VENEER SURFACES

1. The most difficult defect to detect is blue stain, which could not be detected with over 62% defect accuracy by any of the algorithms discussed.
2. Pitch streaks are also hard to detect but one of the algorithm did reach 91% defect accuracy.
3. Pitch pockets, tight knots, loose knots, and wane can be detected with up to 95% defect accuracy by certain algorithms.

7.4 CONCLUSIONS REGARDING THE ALGORITHMS

1. The three edge detectors performed similarly in finding defects but the entropy edge detector and nonlinear Laplace edge detector with a window size of 3 were better at finding fewer false defects.
2. The moment-preserving algorithm (both global and local versions) performed so poorly that the global moment-preserving thresholding algorithm could only find tight knots and loose knots with a defect accuracy over 83% but less than 90% and the local moment-preserving thresholding could not find defects with a defect accuracy over 40%. Therefore, the moment-preserving algorithm was excluded from the experimental analysis and also was not suggested at all for defect detection on veneer surfaces.
3. In the global versus local version of the thresholding algorithms, global entropy thresholding is better than local entropy thresholding because fewer false defects were found and it has a better capability to detect defects. The global Otsu thresholding algorithm is better than the local Otsu thresholding algorithm because it finds fewer false defects while maintaining similar defect detection capability. The local transition-matrix thresholding algorithm is better than the global transition-matrix thresholding algorithm because it finds fewer false defects and has better a greater capability to detect defects. Therefore, there is no tendency regarding adequacy in

global thresholding and local thresholding for defect detection on veneer surfaces since performance depends on the algorithm itself.

4. Among the three prechosen thresholding algorithms (global entropy, global Otsu, and local transition-matrix thresholding), the global entropy thresholding algorithm is the best because it found the fewest false defects and found wane with the highest defect accuracy, though it was the worst at finding blue stain. However, the other two algorithms only found blue stain with less than 62% defect accuracy. The local transition-matrix thresholding found too many false defects but resulted in a higher defect accuracy in wane than the global Otsu thresholding algorithm. Therefore, any comparison between these two algorithms becomes application dependent. If wane is important, then the local transition-matrix thresholding algorithm is a better choice. If the number of false defects detected is of concern, then the global Otsu thresholding is suggested.

5. Among the seven prechosen algorithms (entropy edge detector, nonlinear Laplace edge detector with a window size of 3, a method combining splitting-and-merging with region growing, a method combining clustering with region growing, global entropy thresholding, global Otsu thresholding, and local transition-matrix thresholding), the two region extraction algorithms are superior in finding defects among the three segmentation categories and also their parameters can be easily changed to monitor their performances to meet application requirements. The method combining clustering with region growing does not perform as well as the method combining splitting-and-merging with region growing because it finds more false defects. Two of the edge detectors can only find defects with clear boundaries, such as loose knots, pitch pockets, and wane but not defects whose boundaries have large transition zones, such as blue stain, pitch streaks, and some tight knots. The three thresholding algorithms are better than the edge detection algorithms since they can find some of the more difficult defect types (blue stain and pitch streaks) but they found much more false defects than edge detection algorithms.

7.5 AREAS FOR FUTURE RESEARCH

Areas for future research include:

1. Use images taken using a camera that actually has a 5-bit camera resolution and then with an 8-bit resolution camera to compare the camera resolutions for each algorithm. This is simply a verification test.
2. Use other color spaces to see if they are better than the RGB color space; for instance, the Luv color space, the H-V-S color space, and Ohta's O_1-O_2-O_3 color space. Also, try the native Lab color space instead of the transformation to compare to the RGB color space.
3. For each algorithm under a proper camera resolution and color space, find the best parameters for each defect type so that the best performance for each algorithm for each defect type can be found. Then, the best algorithm for each defect type can be found by statistical analysis.
4. Group the images for each feature according to the grading rules or their sizes and intensities and analyze the performance of algorithms by those groups.
5. Find a proper region extraction method instead of an edge detection method for the clear wood test for all thresholding algorithms to improve their performances.
6. Search for better light sources to enhance the desired features of the veneer surfaces; for instance, use different camera filters.

BIBLIOGRAPHY

Abutaleb, A.S. 1989. Automatic thresholding of gray-level picture using two-dimensional entropy. *Computer Vision, Graphics and Image Processing* 47:22-32.

Aggarwal, P.K., and J.W. Bacus. 1977. A multi-spectral approach for scene analysis of cervical cytology smears. *Journal of Histochemistry and Cytochemistry* 25:668-680.

Ahuja, A., and A. Rosenfeld. 1978. A note on the use of second-order gray-level statistics for threshold selection. *IEEE Transaction on Systems and Man Cybernetics* SMC-8:895-899.

Amadasun, M., and R.A. King. 1988. Low-level segmentation of multispectral images via agglometative clustering of uniform neighborhoods. *Pattern Recognition* 21(3):261-268.

Asano, T., and N. Yokoya. 1981. Image segmentation scheme for low-level computer vision. *Pattern Recognition* 14(1-6):267-279.

Bajcsy, R., S.W. Lee, and A. Leonardis. 1990. Color image segmentation and color constancy. *SPIE Perceiving, Measuring, and Using Color* (1250):245-255.

Ballard, D.H. 1981. Generalizing the Hough transformation to detect arbitrary shapes. *Pattern Recognition* 13(2):111-122.

Bartz, M.R. 1969. Optimizing a video processor for OCR. *Procedings of International Joint Conference on Artificial Intelligence* 70-90. Mitre Corporation, Bedford, Massutruss.

Basu, S., and K.S. Fu. 1987. Image segmentation syntactic method. *Pattern Recognition* 20(1):33-44.

Beckers, A.L.D. 1986. *Metingen van parameters voor Viet-Linearire Objectgrootte-filters in beelden.* Ingenieur's thesis, Departement of Applied Physics, Delft University of Technology,,Dutch.

Bernsen, J. 1986. Dynamic Thresholding of Gray-level Images. *International Conference on Pattern Recognition* 1251-1255.

Bhanu, B., and B.A. Parvin. 1987. Segmentation of natural scenes, *Pattern Recognition* 20(5):487-496.

Bhanu, B., and O. Faugeras. 1982. Segmentation of images having unimodal distributions. *IEEE Transaction on Pattern Analysis, Machine Intelligence* PAMI-4(4):408-419.

Billmeyer, F.W., and M. Saltzman. 1981. *Principles of Color Technology*. New York: John Wiley and Sons.

Boukharouba, S., J.M. Rebordao, and L. Wendel. 1985. An amplitude segmentation method based on the distribution function of an image. *Computer Vision, Graphics and Image Processing* 29:47-59.

Bovik, A.C., T.S. Huang, and D.C. Munson. 1986. Nonparametric tests for edge detection in noise. *Pattern Recognition* 19(3):209-219.

Bribiesca, R., and A. Guzman. 1980. How to describe the form and how to measure differences in shapes using shape numbers. *Pattern Recognition* 12:101-112.

Brice, C., and C. Fennema. 1970. Scene analysis using regions. *Artificial Intelligence* 1:205-226.

Brink, A.D. 1992. Thresholding of digital images using two-dimensional entropies. *Pattern Recognition* 25(80):803-808.

Browning, J.D., and S.L. Grey. 1982. Segmentation of pictures into regions with a tile by tile method. *Pattern Recognition* 15:1-10.

Brunner, C.C., G.B. Show, D.A. Butler, and J.W. Funck. 1989. Using color in machine vision systems for wood processing. *Wood and Fiber Science* 22(4):413-428.

Brunner, C.C., A.G., Maristany, A.B. Butler, D. VanLeeuwen, and J.W. Funck. 1992. An evaluation of color spaces for detecting defects in Douglas-fir veneer. *Industrial Metrology* 2:169-184.

Brunner, C.C., A.G., Maristany, A.B. Butler, and J.W. Funck. 1993. Enhanceing color-image datq for wood-surface feature identification. *Conference Proceedings of the 5th International Conference on Scanning Technology & Process Control for the Wood Products Industry*, Pages Brunner 1-6. Atlanta, GA. U.S.A., October 25-27.

Bryant A. and J. Bryant. 1989. Recognizing shapes in planar binary images. *Pattern Recognition* 22(2):155-164.

Bullock, B.L. 1976. Finding structure in outdoor scenes. *Pattern Recognition and Artificial Intelligence*. ed. C. H. Chen, New York: Academic Press.

Burt, P.J., T.H. Hong, and A. Rosenfeld. 1981. Segmentation and estimation of image region properties through cooperative hierarchical computation. *IEEE Transaction on System, Man Cybernetics* SMC-11(12):802-809.

Cahn, R.L., R.S. Poulsen, and G. Toussaint. 1977. Segmentatión of cervical cell images. *Journal of Hitochemistry and Cytochemistry* 25:681-688.

Cannon, R.L., J.V. Dave, and J.C. Bezdek. 1986. Efficient implementation of the fuzzy c-means clustering algorithms. *IEEE Transaction on Pattern Analysis and Machine Intelligence* PAMI-8(2):248-255.

Canny, J.F. 1986. A computational approach to edge detection. *IEEE Transaction on Pattern Analysis and Machine Intelligence* 8:679-698.

Carlton, S.G., and O.R. Mitchell. 1977. Image segmentation using texture and grey level. *Proceedings of IEEE Conference on Pattern Recognition and Image Processing*, 387-391. Troy, New York.

Celenk, M. 1988. A recursive clustering technique for color picture. *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 437-444.

Celenk, M. 1990. A color clustering technique for image segmentation. *Computer Vision, Graphics, and Image Processing* 52:145-170.

Celenk, M., and S.H. Smith. 1986. Color image segmentation by clustering and parametric-histograming technique. *Proceedings of International Conference on Pattern Recognition*, 883-885.

Chassery, J.M., and C. Garbay. 1984. An iterative segmentation method based on a contextual color and shape criterion. *IEEE Transaction on Pattern Analysis and Machine Intelligence* PAMI-6(6):794-800.

Cheevasuvit, F., H. Maitre, and D. Vidal-Madjar. 1986. A robust method for picture segmentation based on a split-and-merge procedure. *Computer Graphics and Image Processing* 34:268-281.

Chen, L.H. and W.H. Tsai. 1988. Moment-preserving curve detection. *IEEE Transaction on System and Man Cybernetics* SMC-18(1):148-158.

Chen, P.C. and T. Pavlidis. 1980. Image segmentation as an estimation problem. *Computer Graphics and Image Processing* 12:153-172.

Chen, S.Y., W.C. Lin, and C.T. Chen. 1989. An expert vision system for medical image segmentation. *Proceedings of SPIE* Volume 1092: Medical Imaging III:Image Processing, 162-172.

Chen, S.Y., W.C. Lin, and C.T. Chen. 1991. Splitting-and-merging image segmentation based on localized feature analysis and statistical tests. *Graphic Models and Image processing* 53(5):457-475.

Chow. C.K., and T. Kaneko. 1972. Automatic boundary detection of left ventricle from cineangiogram. *Computer Biomedical Research* 5:338-410.

Cohen, F.S., and D.B. Cooper. 1983. Real-time textured image segmentation based on non-causal Markovian random field models. *Proceedings of 3rd International Conference on Robotics Vision Sensor Control*, 17-27. Cambridge, Masstruses.

Coleman, G.B., and H.C. Andrews. 1979. Image segmentation by clustering. *Proceedings of IEEE* 67(5):773-785.

Commission Internationale de L'Eclairage. 1986. *Colorimetry*. 2nd ed. CIE Publication No 15.2. Central Bureau of the CIE. Vienna, Austria.

Conners, R.W., and C.A. Harlow. 1980. A theoretical comparison of texture algorithms. *IEEE Transaction on Pattern Analysis and Machine Intelligence* PAMI-2:204-222.

Conners, R.W, C.W. McMillin, K. Lin, and R.E. Vasquez-Espinosa. 1983. Identifying and locating surface detects in wood: Part of an Automated Lumber Processing System. *IEEE Transaction on Pattern Analysis and Machine Intelligence* PAMI-5(6):273-583.

Conners, R.W, C.W. McMillin, and C.N. Ng. 1985. The utility of color information in the location and identification of defects in surfaced hardwood lumber. *Proceedings of the 1st International Conference on Scanning Technique in Sawmilling*, XVIII-1 to XVIII-33. Hyatt on Union Square, San Francisco, California. U.S.A.

Conners, R.W., M.M. Trivedi, and C.A. Harlow. 1984. Segmentation of a high-resolution urban scene using texture operators. *Computer Graphics and Image Processing* 25:273-310.

Davis, L.S. 1975. A survey of edge detection techniques. *Computer Graphics and Image Processing* 4:248-270.

Davis, L.S., and A. Rosenfeld. 1978. Noise cleaning by iterated local averaging. *IEEE Transaction on System, and Man Cybernetics* SMC-8:705-710.

Deravi, F., and S.K. Pal. 1983. Grey level thresholding using second-order statistics. *Pattern Recognition Letters* 1:417-422.

Derin, H., and W. Cole. 1986. Segmentation of textured images using Gibbs random fields. *Computer Vision, Graphics, and Image Processing* 35:72-98.

Doherty, M.F., C.M. Bjorklund, and M.T. Noga. 1986. Split-merge: an enhanced segmentation capability. *Proc. International Conference on Pattern Recognition*, 325-330.

Doherty, M.F., M.T. Noga, and C.M. Bjorklund. 1985. Use of compound predicates in split-merge segmentation. *IEEE Conference on Computer Vision and Pattern Recognition,* 659-661. San Francisco.

Doyle, W. 1962. Operations useful for similarity-invariant pattern recognition. *Journal of Association of Computer* 9:259-267.

Draper, B.A., B.A. Collins, J. Brolio, and E.M. Riseman. 1989. The scheme system. *International Journal of Computer Vision* 2:209-250.

Duda, R.O., and P.E. Hart. 1972. Use of the Hough transformation to detect lines and curves in pictures. *Communs ACM* 15:11-15.

Duda, R.O., and P.E. Hart. 1973. *Pattern Classification and Scene Analysis*. New York: Wiely.

Eichel, P., and E. Delp. 1985. Sequential edge detection in correlated random fields. *Proceedings of International Conference on Computer Vision and Pattern Recognition*, 14-21. San Francisco.

Elliott, H., and L. Srinivasan. 1981. An application of dynamic programming to sequential boundary estimation. *Computer Graphics and Image Processing* 17:291-314.

Fekete, G., J.O. Eklundh, and A. Rosenfeld. 1981. Relaxation: Evaluation and applications. *IEEE Transaction on pattern Analysis and Machine Intelligence* PAMI-3:460-469.

Feldman, A., and Y. Yakimovsky. 1974. Decision theory and artificial intelligence. *I.A. Semantics based on Region Analyzer Artificial Intelligence* 5:349-371.

Forrer, J.B. 1987. *Image pre-processing algorithms for isolation of defects in douglas-fir veneer.* Master thesis in Department of Forest Products, Oregon State university.

Forrer, J., D.A. Butler, J.W. Funck, and C.C. Brunner. 1988. Image sweep-and-mark algorithms. Part 1. Basic algorithms. *Forest Products Journal* 38(11/12):75-79.

Forrer, J., D.A. Butler, C.C. Brunner, and J.W. Funck. 1989. Image sweep-and-mark algorithms. Part 2. Performance evaluations. *Forest Products Journal* 39(1):39-42.

Fram, J.R., and E.S. Deutsch. 1975. On the quantitative evaluation of edge detection schemes and their comparison with human performance. *IEEE Transactions on Computer* C-24:616-628.

Fu, K.S. 1982. *Syntactic patter recognition and applications*. Engle-wood Clitts NJ: Prentice-Hall.

Fukada, Y. 1980. Spatial clustering procedures for region analysis. *Pattern Recognition* 12:395-407.

Furst, M.A., and P.E. Caines. 1984. Edge detection for digital grey level images via dynamic programming. *Proceedings of International Conference on Pattern Recognition,* 55-58.

Gambotto, J.P. 1986. A hierarchical segmentation algorithm. *Proceedings of International Conference on Pattern Recognition*, 951-95.

Gambotto, J.P., and O. Monga. 1985. A parallel and hierarchical algorithm for region growing. *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 649-652. San Francisco.

Ghelli, F.C. 1988. A sequential learning method for boundary detection. *Pattern Recognition* 21(2):131-139.

Glasbey, C.A. 1993. An analysis of histogram-based thresholding algorithms. *Graphical Model and Image Processing* 55(6):532-537.

Goldberg, M., and S. Shlien. 1978. A cluster scheme for multispectral images. *IEEE Transaction on System and Man Cybernetics* SMC-8(2):86-92.

Gonzalez, R.C., and P. Wintz. 1987. *Digital Image Processing*. Addison-Wesley, Masstruses.

Griffith, A.K. 1973. Edge detection in simple scenes using a priori information. *IEEE Transaction on Computers* C-22:371-381.

Gupta, J.N., and P.A. Wintz. 1974. Computer processing algorithm for locating boundaries in digital pictures. *Proceedings of International Joint Conference on Pattern Recognition*, 155-156.

Haddon, J.F. 1988. Generalized threshold selection for edge detection. *Pattern Recognition* 21(3):195-203.

Haralick, R.M. 1980. Edge and region analysis for digital image data. *Computer Graphics and Image Processing* 12:60-70.

Haralick, M. 1984. Digital step edge from zero crossing of second directional derivatives. *IEEE Transaction on Pattern Analysis and Man Intelligence* 6(1):39-49.

Haralick, R.M., and I. Dinstein. 1975. A spatial clustering procedure for multi-image data. *IEEE Transaction on Circuits System* CAS-22(5):440-450.

Haralick, R.M., and J.S.J. Lee. 1990. Context dependent edge detection and evaluation. *Pattern Recognition* 23(1/2):1-19.

Haralick, R.M., K. Shanmugam, K., and I. Dinstein. 1973. Texture features for image classification. *IEEE Transaction on Systems and Man Cybernetics* SMC-3:610-621.

Haralick, R.M., and L.G. Shapiro. 1985. Survey: image segmentation techniques. *Computer Vision, Graphics and Image Processing* 29:100-132.

Hattori, K., and Y. Torii. 1993. Effective algorithms for the nearest neighbor method in the clustering problem. *Pattern Recognition* 26(50):741-746.

Hauser, R.F. 1984. A stochastic approach to edge detection. *Proceedings of International Conference on Pattern Recognition*, 52-54.

Healey, G., and T.O. Binford. 1987. Color algorithms for a general vision system. *International Journal of Computer Artificial Intelligence*, 759-762.

Horowitz, S.L., and T. Pavlidis. 1974. Picture segmentation by a directed split-and-merge procedure. *Proceedings of International Joint Conference on Pattern Recognition*, 424-433.

Horowitz, S.L., and T. Pavlidis. 1976. Picture segmentation by a tree traversal algorithm. *Journal of Association of computer and Machine*,368-388.

Huang, J., and D.H. Tseng. 1988. Statistical theory of edge detection. *Computer Vision, Graphics and Image Processing* 43:337-346.

Huang, C., T. Cheng, and C. Chen. 1992. Color images segmentation using scale space filter and Markov random field. *Pattern Recognition* 25(10):1217-1229.

Hueckel, M. 1971. An operator which locates edges in digital pictures. *Journal of Association of computer and Machine* 20:634-547.

Hueckel, M. 1973. A local visual operator which recognizes edges and lines. *Journal of Association of Computer and Machine* 20:634-647.

Hunter, R.S., and R.W. Harold. 1987. *The measurement of appearance*. New York: John Wiley & Sons.

Huntsberger, T.L., and M.F. Descalzi. 1985. Color edge detection. *Pattern Recognition Letters* 3:205-209.

Hus, C.C., and J.S. Huang. 1990. Partitioned Hough transform for ellipsoid detection. *Pattern Recognition* 23(3-4):275-282.

Illingworth, J., and J. Kittler. 1987. The adaptive Hough transform. *IEEE Transaction on Pattern Analysis and Machine Intelligence* PAMI-9(5).

Jarvis, R.A. 1977. Region based image segmentation using shared near neighbor clustering. *The 7th International Conference on Cybernetics and Society*, 19-21. Washington, D.C. Sept. 1977.

Johannsen, G., and J. Bille. 1982. A threshold selection method using information measures. *Proceedings of the 6th International Conference on Pattern Recognition*, 140-143, Munich, Germany.

Kapur, J.N., P.K. Sahoo, and A.K.C. Wong. 1985. A new method for grey-level picture thresholding using the entropy of the histogram. *Computer Vision, Graphics, and Image Processing* 29:273-285.

Kartz, Y.H. 1965. Pattern recognition of meteorological satellite cloud photography. *Proceedings of the 3rd Sympothum on Remote Sensing of Environment*, 173-214.

Kelly, M.D. 1971. Edge detection in pictures by computer using planning. *Machine Intelligence* 6:397-409. Edinburgh: Edinburgh University Press.

Khotanzad, A., and A. Bouarfa. 1990. Image segmentation by a parallel, non-parametric histogram based on clustering algorithm. *Pattern Recognition* 23(9):961-973.

Kimme, C., D.H. Ballard, and J. Sklansky. 1975. Finding curves by an array of accumulators. *Communs ACM* 18:120-122.

Kirby, R.L., and A. Rosenfeld. 1979. A note on the use of (grey level, local average grey level) space as an aid in thresholding selection. *IEEE Transaction on Systems and Man Cybernetics* SMC-9:860-864.

Kirsch, R. 1971. Computer determination of the constituent structure of biological images. *Computer Biomedical Research* 4:315-328.

Kitchen, L., and A. Rosenfeld. 1981. Edge evaluation using local edge coherence. *IEEE Transaction on System and Man Cybernetics* SMC-11(9):597-605.

Kittler, J., and J. Illingworth. 1985. Threshold selection based on a simple image statistic. *Computer Vision, Graphics, and Image Processing* 30:125-147.

Kittler, J., and J. Illingworth. 1986. Minimum error thresholding. *Pattern Recognition* 19(1):41-47.

Kohler, R. 1981. A segmentation system based on thresholding. *Computer Graphics and Image Processing* 15:319-338.

Koontz, W.L., P.M. Narendra, and K. Fudunaga. 1976. A graph theoretic approach to nonparametric cluster analysis. *IEEE Transaction on Computer* C-25:936-944.

Krishnapuram, R., H. Frigui, and O. Nasraoui. 1993. The fuzzy C quadratic shell clustering algorithm and the detection of second-degree curves. *Pattern Recognition Letters* 14:545-552.

Kullback, S. 1959. *Information Theory and Statistics*. New York: Wiley.

Kundu, M.K. 1990. Robust edge detection. *Pattern Recognition* 23(5): 423-440.

Kundu, M.K., and S.K. Pal. 1986. Thresholding for edge detection using human psychovisual phenomena. *Pattern Recognition letters* 4:433-441.

Kurita, T., N. Otsu, and N. Abdelmalek. 1992. Maximum likelihood thresholding based on population mixture models. *Pattern Recognition* 25(10):1231-1240.

Lam, L., S.W. Lee, and C.Y. Suen. 1992. Thinning methodologies - a comprehensible survey. *IEEE Transaction on Pattern Analysis and Machine Intelligence* PAMI-14(9):869-885.

Laprade, R.H. 1988. Split-and-merge segmentation of aerial photographs. *Computer Vision, Graphics, and Image Processing* 44:77-86.

Leavers, V.F. 1993. Which Hough Transform ? *Computer Vision, Graphics, and Image Processing: Image Understanding* 58(2):250-264.

Lee, C.H. 1986. Recursive region splitting at hierarchical scope views. *Computer Vision, Graphics, and Image Processing* 33:237-258.

Lee, D.L., G.W. Wasilkowski, and R. Mehrotra. 1993. A new zero-crossing-based discontinuity detector. *IEEE Transaction on Image Processing* 2(2):265-268.

Lee, J.S.J., R.M. Haralick, and L.G. Shapiro. 1986. Morphologic edge detection. *Proceedings of International Conference on Pattern Recognition*, 369-372.

Lee, S.U., and S.Y. Chung. 1990. A comparative performance study of several global thresholding techniques for segmentation. *Computer Vision, Graphics, and Image Processing* 52:171-190.

Lester, L.M., H.A. Williams, B.A. Weintraub, and J.F. Brenner. 1978. Two graph searching techniques for boundary finding in white blood cell images. *Computer Biological Medic* 8:293-308.

Levine, M.D. 1985. Rule-based image segmentation: A dynamic control strategy approach. *Computer Vision, Graphics, and Image Processing* 32:127-140.

Levine, M.D., and A.M. Nazif. 1985. Dynamic measurement of computer generated image segmentation. *IEEE Transaction on Pattern Analysis and Machine Intelligence* PMAI-7(2):155-164.

Levine, M.D., and S.I. Shaheen. 1981. A modular computer vision system for picture segmentation and interpretation. *IEEE Transaction on Pattern Analysis and Machine Intelligence* PAMI-3(5):540-556.

Levy, M. 1988. A new theoretical approach to relaxation, application to edge detection. *Proceedings of International Conference on Pattern Recognition*, 208-212.

Li, C.H., and C.K. Lee. 1993. Minimum cross entropy thresholding. *Pattern Recognition* 26(4):617-625.

Li, Z.N., B. Yao, and F. Tong. 1993. Linear generalized Hough transform. *Pattern Recognition Letters* 11(11):725-733.

Lim, Y.W., and S.U. Lee. 1990. On the color image segmentation algorithm based on the thresholding and the fuzzy c-means techniques. *Pattern Recognition* 23(9):935-952.

Lkatosh, B.K., 1966. *Defectoscopy of wood.* translated in 1970. Indian National Scientific Documentation Center, New Delhi.

Luijendijk, H. 1991. Automatic threshold selection using histograms based on the count of 4-connected regions. *Pattern Recognition Letters* 12:219-228.

Maristany, A.G., A.B. Butler, C.C. Bruuner, and J.W. Funck. 1991. Exploiting local color information for defect detection on Douglas-fir veneer. *Proceedings of the 4th International Conference on Scanning Technique in the Wood Industry*, Pages Maristany 1-7. Burlingame, San Francisco, California, U.S.A., October 28-29.

Marr, D. 1982. *Vision.* San Francisco: Freeman. Marr, D., and E. Hildreth. 1980. Theory of edge detection. *Proceedings of Royal Society*, London B207, 187-217.

Martelli, A. 1976. An application of heuristic search methods to edge and contour detection. *Communs ACM*-19:73-83.

Mason, D., I.J. Lauder, D. Rutoritz, and G. Powart. 1975. Measurement of C-Bands in human chromosomes. *Computer Biological Medic* 5:179-201.

Matsuyama, T. 1989. Expert systems for image processing: Knowledge based composition of image analysis processing. *Computer Vision, Graphics, and Image Processing* 48:22-49.

McLean, G.F., and M.E. Jernigan. 1988. Hierarchical edge detection. *Computer Vision, Graphics, and Image Processing* 44:350-366.

McMillin, C.W. 1982. Application of automatic image analysis to wood science. *Wood Science* 14(3):97-105.

McMillin, C.W. 1984. Evaluating wood failure in plywood shear by optical image analysis. *Forest Products Journal* 34(7/8):67-69.

Meer, P., and H. Wechsler. 1989. Edge detection by associative mapping. *Pattern Recognition* 22(5):491-503.

Milgram, D.L. 1979. Region extraction using convergent evidence. *Computer Graphics and Image Processing* 11:1-12.

Minor, L.G., and J. Sklansky. 1981. The detection and segmentation for blobs in infrared images. *IEEE Transaction on System and Man Cybernetics* SMC-11(3):194-201.

Montanari, U. 1971. On the optimal detection of curves in noisy pictures. *Communs ACM* 14:335-345.

Montgomery, D.C. 1991. Design and Analysis of Experiments. 3rd edition, John Wiley & Sons.

Moon, P., and D.E. Spencer. 1948. The color of unstained wood. *Journal of the Optical Society of America* 38(4):404-408.

Morrin, T.H. 1974. A black-white representation of a grey-scale picture. *IEEE Transaction on Computer* C-23:184-186.

Nagin, P., A. Hanson, and E. Riseman. 1978. Relaxation-based segmentation based on spatial context and feature space cluster labels. *Proceedings of IEEE Conference on Pattern Recognition and Image Processing*, 421. Chicago, Illinois.

Nalwa, V.S. 1987. Edge-detector resolution improvement by image interpolation. *IEEE Transaction on Pattern Analysis and Machine Intelligence* PAMI-9(3):446-451.

Narendra, P.M., and M. Goldberg. 1977. A non-parametric clustering scheme for LANSAT. *Pattern Recognition* 9:207-215.

Neuvo, Y., P. Heinonen, and I. Defee. 1986. New types of edge detectors based on linear and median operators. *Proceedings of International Conference on Pattern Recognition*, 702-704.

Nevatia, R. 1977. A color edge detector and its use in scene segmentation. *IEEE Transaction on System and Man Cybernetics* SMC-7(11):820-826.

Oe, S. 1993. Texture segmentation method by using two-dimensional AR model and Kullback information. *Pattern Recognition* 26(2):237-244.

Ohlander, R., K. Price, and D.R. Reddy. 1978. Picture segmentation using a recursive region splitting method. *Computer Graphics and Image Processing* 8:313-332.

Ohta, Y., T. Kanade, and T. Sakai. 1980. Color information for region segmentation. *Computer Graphics and Image Processing* 13:222-241.

Otsu, N. 1979. A threshold selection method from grey-level histogram. *IEEE transaction on Systems and Man Cybernetics* SMC-9:62-66.

Overheim, R.D., and D.L. Wagner. 1982. *Light and Color*, New York: Wiley.

Pal, N.R., and S.K. Pal. 1993. A review on images segmentation techniques. *Pattern Recognition* 26(9):1277-1294.

Pal, S.K., and R.A. King. 1983. On edge detection of X-ray images using fuzzy sets. *IEEE Transaction on Pattern Analysis and Machine Intelligence* PAMI-5(1):69-77.

Pal, S.K., and N.R. Pal. 1987. Segmentation based on measures of contrast, homogeneity and region size. *IEEE Transaction on System and Man Cybernetics* SMC-17(5):857-868.

Pal, S.K. and N.R. Pal. 1988. Object extraction from image using higher order entropy. *Proceedings of International Conference on Pattern Recognition*, 348-350.

Panda, D.P. 1977. Segmentation of FLIR image by pixel classification. *University of Maryland Computer Science Center TR 508.*

Parvin, B.A. 1984. A split and merge algorithm for segmentation of natural scenes. *The 7th International Conference on Pattern Recognition*, 294-295.

Pavlidis, T. 1972. Segmentation of pictures and maps through functional approximation. *Computer Graphics and Image Processing* 1:360-372.

Peleg, S. 1978. Iterative histogram modification. *IEEE Transaction on System and Man Cybernetics* SMC-8:555-556.

Peleg, S. 1980. A new probabilistic relaxation scheme. *IEEE Transaction on Pattern Analysis and Mach Intelligence* PAMI-2:362-369.

Peli, T., and D. Mahah. 1982. A study of edge detection algorithms. *Computer Graphics and Image Processing* 20:1-21.

Persoon, E. 1975. A new edge detection algorithm and its applications in picture processing. *TR-EE 75-38 Purdue University*. West Lafayette, Indiana, Oct.

Petrou, M., and J. Kittler. 1991. Optimal edge detectors for tamp edges. *IEEE Transaction on Pattern Analysis and Machine Intelligence* 1(5):483-491.

Pietikainen, M., and A. Rosenfeld. 1981. Image segmentation by texture using pyramid node linking. *IEEE Transaction on system and Man Cybernetics* SMC-11(12):822-825.

Pietikainen, M., and D. Harwood. 1986. Edge information in color images based on histogram of differences. *Proceedings of International Conference on Pattern Recognition*, 594-596.

Poelzleitner, W. 1986. Hough transformation method to segment images of wooden boards. *International Conference on Pattern Recognition*, 262-262.

Pong, T.C., L.G. Shapiro, L.T. Watson, and R.M. Haralick. 1984. Experiment in segmentation using a facet model region grower. *Computer Vision, Graphics and Image Processing* 25:1-23.

Pratt, W.K. 1978. *Digital Image Processing*. New York: Wiley.

Prewitt, J.M.S. 1970. Object enhancement and extraction. Picture Processing and Psychopictorics, ed. B.S.Lipkin and A.Rosenfeld, 75-149. New York: Academic Press.

Prewitt, J.M.S. and M.L. Mendelsohn. 1966. The analysis of cell images. *Annual New York Academic Science* 128:1035-105.

Pun, T. 1980. A new method for grey level picture thresholding using the entropy of the histogram. *Signal Processing* 2:223-237.

Pun, T. 1981. Entropic thresholding: A new approach. *Computer Vision, Graphics and Image Processing* 16:210-239.

Raafat, H.M., and A.K.C. Wong. 1988. A texture information-directed region growing algorithm for image segmentation and region classification. *Computer Vision, Graphics, and Image Processing* 43:1-21.

Reed, T.R. 1993. A review of recent texture segmentation and feature extraction techniques. *Computer Vision, Graphics, and Image Processing: Image Understanding* 57(3):359-372.

Reed, T.R., H. Wechsler, and M. Werman. 1990. Texture segmentation using a diffusion region growing technique. *Pattern Recognition* 23(9):953-960.

Roberts, L.G. 1965. Machine perception of three dimensional solids. *Optical and Electro-Optical Information Processing Journal.*

Robinson, G.S. 1977a. Edge detection by compass gradient masks. *Computer Graphics and Image Processing* 6:492-501.

Robinson, G.S. 1977b. Color edge detection. *Optical Engineering* 16(5):479-484.

Rosenfeld, A., and A.C. Kak. 1976. *Digital picture Processing*. New York: Academic Press.

Rosenfeld, A. and L.S. Davis. 1978. Iterative histogram modification. *IEEE Transaction on System and Man Cybernetics* SMC-8(4):300-302.

Rosenfeld, A., R.A. Hummel, and S.W. Zucker. 1976. Scene labeling by relaxation operations. *IEEE Transaction on System and Man Cybernetics* SMC-6:420-430.

Rosenfeld, A., and R.C. Smith. 1981. Thresholding using relaxation. *IEEE Transaction on Pattern Analysis and Machine Intelligence* PAMI-3(5):598-606.

Rosenfeld, A., and M. Thurston. 1971. Edge and curve detection for visual scene analysis. *IEEE Transaction on Computer* C-20:562-569.

Rosenfeld, A., M. Thurston, and Y. Lee. 1972. Edge and curve detection further experiments. *IEEE Transaction on Computer* C-21:677-715.

Rosenfeld, A., and P.D.L. Torre. 1983. Histogram concavity analysis as an aid in threshold selection. *IEEE Transaction on Systems and Man Cybernetics* SMC-13(3):231-235.

Sahoo, P.K., S. Soltani, and A.K. Wong. 1988. A survey of thresholding techniques. *Computer Vision, Graphics, and Image Processing* 41:233-260.

SAS institute Inc. 1988. *SAS/STAT user's guide*, release 6.03 edition. Cary, NC, USA.

Schachter, B.J., L.S. Davis, and A. Rosenfeld. 1977. *Some experiments in image segmentation by clustering of local feature values*. TR-410, Computer Science Center, University of Maryland.

Schettni, R. 1993. A segmentation algorithm for color images. *Pattern Recognition Letters* 14:499-506.

Shapiro, S.D. 1975. Transformations for the computer detection of curves in noisy pictures. *Computer Graphics and Image Process* 4:328-338.

Shen, J., and S. Castan. 1986. An optimal linear operator edge detection. *Proceedings of International Conference on Pattern Recognition*, 109-114.

Shiozaki, A. 1986. Edge extraction using entropy operator. *Computer Vision and Image Processing* 36:1-9.

Shu, J., and H. Freeman. 1987. An expert system for image segmentation. *Proceedings of SPIE International Society Optical Engineering* 829:240-252.

Souza, P.D. 1983. Edge detection using sliding statistical tests. *Computer Vision, Graphics and Image Processing* 23:1-14.

Stansfield, S.A. 1986. ANGY: A rule-based expert system for automatic segmentation of coronary vessels from digital subtracted angiograms. *IEEE Transaction on Pattern Analysis and Machine Intelligence* 8(3):188-199.

Stern, D., and L. Kurz. 1988. Edge detection in correlated noise using Latin square masks. *Pattern Recognition* 21(2):119-129.

Sullivan, J.D. 1967. Color characterization of wood: spectophotometry of wood color. *Forest Products Journal* 17(7):43-48.

Sullivan, J.D. 1967. Color characterization of wood: color parameters of individual species. *Forest Products Journal* 17(8):25-29.

Synder, W. 1990. Optimal thresholding - A new approach. *Pattern Recognition Letters* 11:803-810.

Tanimoto, S., and T. Pavlidis. 1975. A hierarchical data structure for picture processing. *Computer Graphics and Image Processing* 4:104-119.

Taxt, T., P.J. Flynn, and A.K. Jain. 1989. Segmentation of document images. *IEEE Transaction on Pattern Analysis and Machine Intelligence* 11(12):1322-1329.

Therrien, C.W. 1983. An estimation theoretic approach to terrain image segmentation. *Computer Graphics and Image Processing* 22:313-326.

Therrien, C.W. 1985. Multichannel filtering methods for color image segmentation. *Proceeding Conference on Computer Vision and Pattern Recognition*, 637-641.

Tominaga, S. 1986. Color image segmentation using three perceptual attributes. *Proceedings of Conference on Computer Vision and Pattern Recognition*, 628-630.

Trahanias, P.E., and A.N. Venetsanopoulos. 1993. Color edge detection using vector order statistics. *IEEE Transaction on Image Processing* 2(2):259-264.

Trivedi, M.M., and J.C. Bezdek. 1986. Low-level segmentation of aerial image with fuzzy clustering. *IEEE Transaction on System and Man Cybernetics* SMC-16(4):589-598.

Tsai, W. 1985. Moment-preserving thresholding: A new approach. *Computer Vision, Graphics and Image Processing* 29:377-397.

Tsuji, S., and F. Tomita. 1973. A structural analysis for a class of textures. *Computer Graphics and Image Processing* 2:216-231.

Tsuji, S., and R. Fujiwara. 1974. Linguistic segmentation of scenes into regions. *Proceedings of the 2nd International Joint Conference on Pattern Recognition*, 104-108.

Ulman, J.R. 1974. Binarization using associative addressing. *Pattern Recognition* 6:127-135.

Vliet, L.J., and I.T. Young. 1989. A nonlinear Laplace operator as edge detector in noisy images. *Computer Vision, Graphics, and Image Processing* 45:167-195.

Wechsler, H., and M.N. Kikode. 1977. A new edge detection technique and its implementation. *IEEE Transaction on System and Man Cybernetics* SMC-7(12):827-836.

Werman, M., and S. Peleg. 1985. Min-max operators in texture analysis. *IEEE Transaction on Pattern Analysis and Machine Intelligence* PAMI-7(6):730-733.

Weszka, J.S. 1978. A survey of threshold selection techniques. *Computer Graphics and Image Processing* 7:259-265.

Weszka, J.S., C. Dyer, and A. Rosenfeld. 1976. A comparative study of texture measures for terrain classification. *IEEE Transaction on System and Man Cybernetics* SMC-6:269-285.

Weszka, J.S., R.N. Nagek, and A. Rosenfeld. 1974. A threshold selection technique. *IEEE Transaction on Computer* 3:1322-1326.

Wolf, R.N. 1969. A dynamic thresholding scheme for quantization of scanned images. *Proceedings of Automatic Pattern Recognition*, 143-162. National Security Industrial Association, Washington,D.C., May.

Won, C.S., and H. Derin. 1992. Unsupervised segmentation of noisy and textured images using Markov random fields. *Graphical Models and Image Processing* 54(4):308-328.

Wong, S., and R.M. Haralick. 1984. Automatic multithreshold selection. *Computer Vision, Graphics, and Image Processing* 25:46-67.

Wu, A.Y., and A. Rosenfeld. 1982. Threshold selection using quadtree. *IEEE Transaction on Pattern Analysis and Machine Intelligence* PAMI-4(1):90-94.

Wu, X. 1993. Adaptive split-and-merge segmentation based on piecewise least-square approximation. *IEEE Transaction on Pattern Analysis and Machine Intelligence* 15(8):808-815.

Wyszecki, G., and W.S. Stiles. 1976. *Color science: Concepts and Methods, quantitative data and formulas*. New York: John Wiley and Sons.

Yakimovsky, Y. 1976. Boundary and object detection in real world image. *Journal of Association of Computer and Machine* 23:599-618.

Yanowitz S.D., and A.M. Bruckstein. 1988. A new method for image segmentation. *Proceedings of International Conference on Pattern Recognition*, 270-275.

Yoo, J.R., and T.S. Huang. 1978. *Image segmentation by unsupervised clustering and its applications*. TR-EE78-19, Purdue Univ. West Lafayette, Indiana.

You, Z. and A. Jain. 1984. Performance evaluation of shape matching via chord length distribution. *Computer Vision, Graphics, and Images Processing* 28:185-198.

Yuen, H.K., J. Princen, J. Illingworth, and J. Kittler. 1990. A comparative study of Hough transform methods for circle finding. *Image Vision Computer* 8(1):71-77.

Zhang, Y.J., and J.J. Gerbrands. 1991. Transition region determination based thresholding. *Pattern Recognition Letters* 12:13-23.

Zucker, S., R. Hummel, and A. Rosenfeld. 1977. An application of relaxation labelling to line and curve enhancement. *IEEE Transaction on Computer* C-26:394-40.