Automated Detection of Surface Defects on Barked Hardwood Logs and Stems Using 3-D Laser Scanned Data

Liya Thomas

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

Doctor of Philosophy
in
Computer Science

Clifford A. Shaffer, Co-Chair
Lamine M. Mili, Co-Chair
Dennis G. Kafura
Layne T. Watson
A. Lynn Abbott
D. Earl Kline

September 8, 2006
Blacksburg, Virginia

Keywords: Range Image Processing, Robust Estimation, Machine Vision, Log Defect Detection

Copyright 2006, Liya Thomas

Automated Detection of Surface Defects on Barked Hardwood Logs and Stems Using 3-D Laser Scanned Data

Liya Thomas

(ABSTRACT)

This dissertation presents an automated detection algorithm that identifies severe external defects on the surfaces of barked hardwood logs and stems. The defects detected are at least 0.5 inch in height and at least 3 inches in diameter, which are severe, medium to large in size, and have external surface rises. Hundreds of real log defect samples were measured, photographed, and categorized to summarize the main defect features and to build a defect knowledge base. Three-dimensional laser-scanned range data capture the external log shapes and portray bark pattern, defective knobs, and depressions.

The log data are extremely noisy, have missing data, and include severe outliers induced by loose bark that dangles from the log trunk. Because the circle model is nonlinear and presents both additive and non-additive errors, a new robust generalized M-estimator has been developed that is different from the ones proposed in the statistical literature for linear regression. Circle fitting is performed by standardizing the residuals via scale estimates calculated by means of projection statistics and incorporated in the Huber objective function to bound the influence of the outliers in the estimates. The projection statistics are based on 2-D radial-vector coordinates instead of the row vectors of the Jacobian matrix as proposed in the statistical literature dealing with linear regression. This approach proves effective in that it makes the GM-estimator to be influence bounded and thereby, robust against outliers.

Severe defects are identified through the analysis of 3-D log data using decision rules obtained from analyzing the knowledge base. Contour curves are generated from radial distances, which are determined by robust 2-D circle fitting to the log-data cross sections. The algorithm detected 63 from a total of 68 severe defects. There were 10 non-defective regions falsely identified as defects. When these were calculated as areas, the algorithm locates 97.6% of the defect area, and falsely identifies 1.5% of the total clear area as defective.

Dedication

To Louis and Patrick

Acknowledgments

I would like to thank my co-advisor, Dr. Clifford A. Shaffer for his management. He ensured that I made progress toward the completion, and gave advice how to describe, evaluate, and test my defect detection algorithm. I appreciate very much my co-advisor Dr. Lamine Mili for his advice on this study. I appreciate his effort on conducting my study from Washington DC area, and travels to Blacksburg campus to work with me. I would like to thank Dr. Abbott for his advice on Computer Vision Systems. I also thank Dr. Kline for his constant encouragement and his advice on wood science and forest products. Special thanks go to Dr. Watson for his advice on robust statics and numerical methods. Many thanks to Dr. Kafura for advising my study. I would like to thank U.S. Department of Agriculture for partially sponsoring under Grants No 01-CA-11242343-065 and No 02-CA-11242343-083.

My gratitude also goes to Dr. Wiedenbeck for supporting and supervising my research, to Dr. Buehlmann for his constant support. Bowing to Dr. Sosonkina, Joel Weiss, Donghang Guo, and Dora Zeng for their friendship, encouragement, help, and stimulating discussions throughout my study. I am deeply grateful to my family members for their love and support through all these years. My sons Louis and Patrick give me strength and perspectives to complete this process. It is with my husband Ed's encouragement, counsel, and insight that I reached the end of my study. Thanks to my parents who ensured that I focused on this work. I am extremely lucky to have the most understanding and supportive parents in-law. My most sincere appreciation to all colleagues, mentors, friends, and supporters!

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Automatically locating and classifying log defects helps to improve lumber yield, in terms of both volume and quality. Traditional defect inspection is done by the sawyer's naked eye within a matter of seconds. Visual inspection has a high error rate, and is easily influenced by the operator's physical and mental conditions. Thus, researchers have been developing a variety of computerized defect detection and classification systems to assist the sawyers' decision-making process [8].

1.1 Background

In 1991, USDA, NIST, US Department of Commerce, Hardwood Research Council, and the University of Maine sponsored an investigation to identify the hardwood industry's current needs. One of the four most pressing priorities is external and internal defect detection to optimize hardwood logs and lumber processing [8]. The ability to detect defects on hardwood trees and logs holds great promise for the hardwood forest products industry. At every stage of wood processing, there is the potential for improving value and recovery: from bucking hardwood stems into round wood products using optimal grading strategies controlled by

surface scanning data processing, to log breakdown using inferred internal defect data based on external indicators. Before a hardwood log is sawn, an assessment of its quality is usually performed, typically via a mill operator's visual inspection, which can be quite variable and subjective.

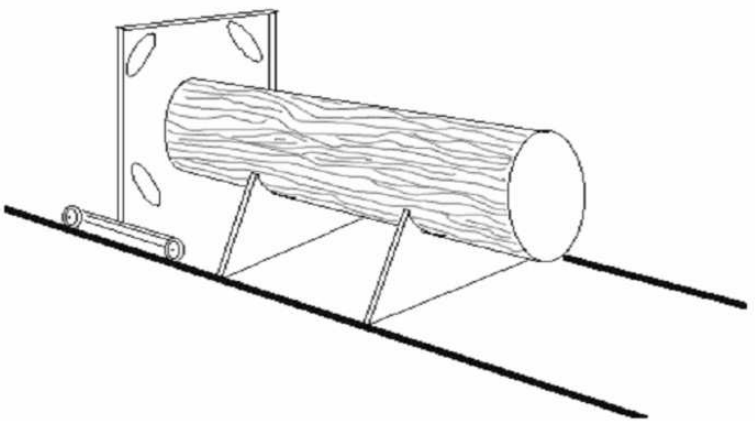
Log quality is inversely proportional to the presence of defects. Log defects include both internal and external defects. External defect indicators consist of knots, splits, holes, and circular distortions in the bark pattern. Key data collected to characterize these indicators include surface rise, length and width as well as type. Defect detection on hardwood trees and logs can be categorized into two areas: internal and external detection. External defect detection refers to the detection of defects on a log's surface, and internal detection, the detection of defects inside a log. The difference between high and low quality logs is determined by defect type, size, and location. Detecting and measuring defects accurately and rapidly is often difficult [94]. Accurate external log defect data would permit bucking of stems to the highest-valued log combination possible. During sawing these data can lead to improved cutting strategies that optimize log yields, that is, preserving the largest possible area of clear wood on a board face.

The last two decades have seen the emergence of various scanning technologies for both the softwood and hardwood industries. Various internal defect inspection methods have been developed using X-ray/CT (Computer Tomography), X-ray tomosynthesis, MRI (Magnetic Resonance Imaging), microwave scanning, ultrasound, and enhanced pattern recognition of regular X-ray images [96, 101, 40, 20, 3, 74]. External log-scanning equipment and accompanying optimization software systems are also available on the market that aid in the sawing of logs into lumber. Most of these scanning systems were developed for the softwood lumber industry and only a handful for hardwoods. Available external hardwood log scanning systems gather information about external log characteristics such as diameter, taper, curvature, and length [72]. Optimization software systems then focus on using this profile information to better position the log on the carriage and improve the sawyer's decision-making ability. Supplying external defect information to these optimization software systems is a natural

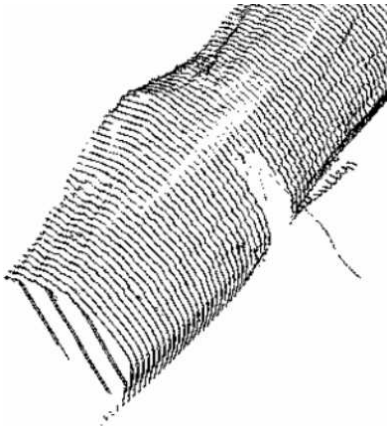
extension of current technology. Using texture analysis, Tian et al. [93] developed a computer vision software system for external defect recognition using photo images of softwood tree stems. Hardwood defect types and morphology are sufficiently different from softwoods to prevent a direct application. Further, the nature of the data, that is, gray-scale photo images, that Tian et al. analyzed are different from the 3-D range data in this research. So far no technology is available that can provide external defect information on hardwood logs and stems.

With the aid of a hardwood log surface defect scanning system, decision making at the headrig can be improved during processing. If scanning occurred early enough in the processing flow, defect information could be used to determine the best product or market by grading logs and/or optimally bucking stems. This would also automate current data collection systems that use an operator to manually identify defects on logs as an aid to processing and grading. Recently, several companies including Perceptron, Inc. [4], have designed 3-D laser-scanning systems to collect log and stem external profile data. Figure 1.1 illustrates the scanner as well as the log data. A computerized detection system is needed to process the 3-D range data and extract defect information. To accomplish the detection process, the system will need to apply multidisciplinary knowledge including wood and forestry science, computer vision, image processing, computer science, and statistics. For it to be practical in the sawing process, the system must also be fast. In this document, we use English measurements for length: inch and foot, commonly adopted by U.S. forest product society.

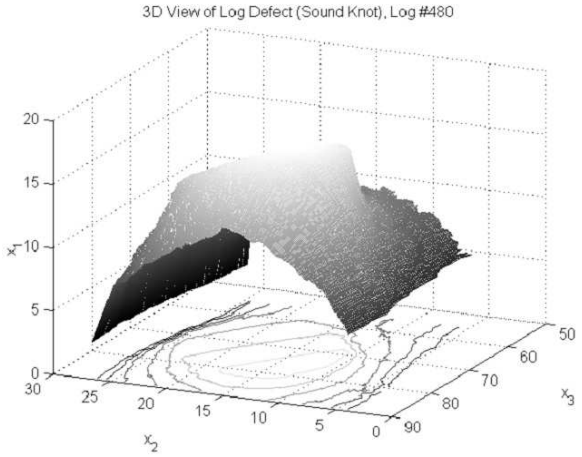
X-Ray/CT technology has been used to locate internal hardwood log defects in the laboratory [40, 101]. Log defects exist both externally and internally. As X-Ray/CT technology is capable of penetrating material, the resulting images display internal defects through density variations. While X-Ray/CT-based detection approaches generate successful experimental results with a 95% detection accuracy [40], several obstacles prevent them from being used in industrial applications. First, the data collection speed is extremely slow due to the large data volume, varying anywhere from 5 minutes to 4 hours per log. Second, variation in moisture content in the log causes the intensity of scanned images to vary, making detection



(a) Schematic diagram of the laser log scanner.



(b) Portion of the 3-D projection of laser-scanned range data for a log sample, a red oak.



(c) 3-D mesh projection of partial data (15,998 points). This portion is a large sound knot as partially shown in (*b*), roughly in the size of 20 inches×13 inches, rising approximately 4 inches above the log surface

Figure 1.1: *The 3-D laser scanning system and the range data. The curves on the* $x_2\,x_3$*plane are contour plots indicating the heights at different elevations on the log surface. In the plot, a high gray value, that is, a light-shaded gray color, indicates a large* x_1 *value.*

results unstable. Third, it presents an environmental hazard, as penetrating such a large object requires a tremendous amount of X-ray energy. Finally, the high cost of the scanning equipment—on average one million U.S. dollars—few sawmills can afford and thus has little practical value.

In contrast, 3-D laser scanner technology uses relatively low-cost equipment that is more affordable to sawmills. Laser scanning equipment collects the external log shape information using triangulation technology. Since only surface data are collected, data collection speed is much faster. The system employs low-energy laser-scanning units, which are safe to operate. Moisture content does not interfere with 3-D profile data. However one main disadvantage for this method is that it only provides external defect information, which might prove insufficient for lumber processing. To address this problem, a sister study [92] to determine the correlation of external and internal defects is ongoing at the USDA Northeastern Forest Research Laboratory in Princeton, WV.

Strong correlations have been found to exist between external indicators and internal characteristics. For the most severe defects, the models can predict internal features such as total depth, midway point defect width and length, and penetration angle, with a low measurement error. For less severe defects such as adventitious knots and medium and light distortions, the correlations are less significant. An adventitious knot is a knot resulting from a branch that sprouted from the main trunk. These types of knots are often small (less than 0.75 inch) and do not penetrate all the way to the center of the tree as do other knots.

Logs can be classified into softwood and hardwood. In general, most softwoods have a fast growth rate and identical, clustered defects mostly caused by branch pruning. By contrast, hardwood trees generally grow more slowly, and have more valuable products. Studies have demonstrated that the use of defect data improves cutting strategies that optimize log recovery or yield, that is, preserving the largest possible area of clear wood on a board face [27, 79]. This is a challenging task to achieve because the distribution, types, and sizes of hardwood defects are random and irregular.

1.2 General Research Objectives

The key objective of this research was to develop an algorithm capable of locating surface defects on hardwood logs using laser profile data. In order to accomplish this objective, several inter-related sub-objectives had to be met:

1. Characterization of hardwood defect types. This required the collection, measurement, photographing, and analysis of external hardwood defect samples;

2. Development of non-linear regression models that are able to perform the detection tasks;

3. Development of a machine vision system based on the knowledge of external hardwood defect samples for defect detection based on contour levels derived from radial distances;

4. Quantification of the control parameters of the detection algorithm. This required testing the accuracy of the algorithm using a range of values to determine the optimal combination of parameters. The optimal combination is one which returns the highest number of correctly identified defects, the lowest number of falsely identified defects, and the lowest number of unidentified defects.

To the best of our knowledge this is the first investigation of detection methods for locating defects on the surface of hardwood logs and stems using laser-scanned 3D Cartesian coordinates [91, 87]. The laser-scanning system is a commonly available industrial system manufactured by Perceptron, Inc. [4]. The scanner generates high-resolution profile images of the log surface in three dimensions. The scanner was primarily developed for the softwood industry, where the scanner would be used to determine the shape and size of the log being sawn in three dimensions. Ideally, an optimizer would take the scanned data and determine the sawing pattern for the log in terms of maximizing volume of lumber sawn. The system resolution is high enough such that defects can be manually located in the scan data by the

human eye. The obvious question is: how to get the computer to extrapolate internal defects given known relationships between surface features and internal log features.

1.3 Achievements

Most severe log defects are associated with a localized surface rise at least 0.5 inch. To detect these features, an automated defect detection algorithm has been developed using laser-scanned profile data. Circles are fit to data cross sections, and then radial distances are computed between the fitted circle and the data [86]. Also explored is the possibility of fitting ellipses or cylinder to log data. From the radial distances a gray-scale image was generated showing height changes on the log surface. Further, radial distances are used to determine a contour plot of the log surface, from which the large and/or protruding defects are determined. However, some types of severe defects do not lead to significant height changes against the surrounding bark, and thus are not detected by the algorithm presented in Chapter 6. Pattern-based methods to identify these kinds of severe defects might be developed in future work. Currently only those defects with a significant height rise were examined.

Log data were obtained from two commercially important north-east America hardwood species: yellow poplar (Tulipifera Liriodendron), and red oak (Quercus Rubra). Over 160 log data samples were collected, each consisting of cross sections along the log length at 0.8-inch intervals (Figure 1.2). Each cross section comprises approximately 1,000 3-D coordinates with adjacent points roughly 0.05 inches apart, so it is much denser along the cross sections than between them. Typically a log's length ranges between 8 and 16 feet. Thus, each log data sample has about 120,000 to 240,000 points. Clearly, the log surface data are range data. Due to blockage by the log's supporting structure during scanning, there are missing data as well as severe outliers introduced. Calibration problems with the scanning units and log diameters also caused missing or duplicated data. Because of the presence of a small

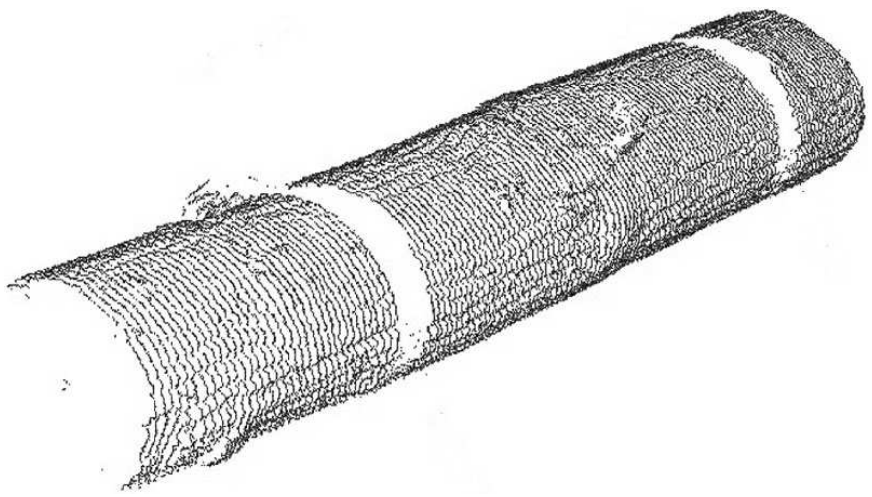


Figure 1.2: *Dot cloud projection of 3-D log data. Shown is part of the data for one log sample. A bump-like external defect (lower left), missing data, and outliers caused by loose bark (upper-middle left) are visible.*

percentage of severe outliers together with segments of missing data over the log surface, conventional least-squares fitting performs poorly. This calls for the development of robust curve fitting methods, which leads to the application of robust statistics and the development of 2-D curve-fitting generalized M-estimator (GME) [21, 91, 86].

Actual defect locations, sizes, types, etc. for these log samples were measured manually. Color digital images of the log surface, four images per log (at 90^o intervals) were captured. About 200 external-defect samples were studied, measured, and their photos taken. These defect samples were analyzed to provide indicators and classification of external defect characteristics. Statistics for these defect classifications are used to define the defect-detection algorithm, and to improve it through comparing its simulation output data against the statistics. These are the training data for the defect detection algorithm, and are further discussed in Sections sec:dt4 and 6.1. In our experiments there are a total of 68 severe defects, of which 63 were correctly identified. There were 10 non-defective regions falsely identified as defects. The 68 defects are testing data to the defect detection algorithm. When these were calculated as areas, the algorithm locates 97.6% of the defect area, and falsely identifies 1.5% of

the total clear area as defective.

The defects under consideration are at least 0.5 inch in height and at least 3 inches in diameter, which are large, severe external surface rises. Testing results are found in Section sec:das5. The method proceeds in three major steps. First, it determines an appropriate reference level—a 2-D circle—to the scanned data cross section along the log length. Next, it obtains radial distances that show surface rise and depression. Finally, it locates severe external defects using the contour image generated from radial distances. This process requires 2-D quadratic curve fitting. A small percentage of outliers exist in the log data among the hundreds of 3-D points per cross section. Statistically, outliers are observations that deviate from the pattern formed by the majority of a data set. In this application they are caused by loose bark or supporting structure of the scanning equipment. Note that currently the defect detection system are implemented using two programming languages: Java for the circle-fitting part, and Matlab for the defect detection part. It is not yet integrated with laser scanning equipment, thus the simulation results are all from lab computers. In next phase, we will integrate both two programs in Java, and experiment with a scanner equipment in real time.

Many least squares 2-D curve-fitting methods have been proposed in the literature; see for example [17, 14, 82, 13]. However the log data are extremely noisy and include large outliers along with missing data. This implies that non-robust least squares fitting fails as it assumes that data are free of outliers and complete. For this application, a good fitting to log data is crucial because subsequent analysis completely relies on its results. It turns out that estimation methods proposed in field of robust statistics, such as the M-estimators introduced by Huber in 1965 and the Least Median of Squares (LMS) estimator proposed by Rousseeuw [21, 30, 70], do not meet the requirements of good resistance to outliers and low computational complexity for circle fitting. This need prompted the development of a new generalized M-estimator whose objective function makes use of scale estimates calculated by means of projection statistics and incorporated in the Huber objective function such that the influence function of the estimator is bounded. The projection statistics algorithm uses the

2-D radial vector coordinates instead of the row vectors of the Jacobian matrix. The vectors start from the fitted circle to the log data cross-section, and pass through the center of the fitted circle. This nonlinear approach proves effective here in that it successfully identifies severe outliers in data, which otherwise would not be identified as outliers by conventional linear methods.

The remaining dissertation is structured as follows: Chapter 2 reviews related work, methodologies, and theories in defect detection, range image processing, image structure modeling, and robust statistical estimation. It also discusses the relationship between internal and external defects and why detecting external defects provides sufficient information for internal ones. Chapter 3 describes various external log defects and their developments during the tree growth. Chapter 4 outlines the detection algorithm, including both the circle fitting process to log data, as well as defect identification from radial distances. Chapter 5 presents the new GM-Estimator and proves that it is influence bounded. Chapter 6 shows the defect detection algorithm, parameter-value testing results, and experiments with data mining technology. Finally, Chapter 7 provides concluding remarks and describes what is planned to be accomplished in the next phase.

Chapter 2

Literature Review

2.1 Defect Detection Systems

There are both internal and external defect detection software systems available for the softwood industry as well as for the hardwood industry. Most internal defect inspection methods on hardwood logs and stems employ technologies using X-ray/CT, X-ray tomosynthesis, MRI, microwave scanning, ultrasound, and enhanced pattern recognition of regular X-ray images. Using CT data, computer vision algorithms are able to accurately locate and describe internal log defects. Wagner et al. [96] investigated a CT scanner that operated at an ultrafast speed, which approached the speed required by commercial sawmill and veneer plants. Internal defects could be seen in the scanned images acquired at such a speed. Thus, the authors concluded that it is possible to develop image analysis techniques to automatically identify internal defects. Guddanti et al. [20] developed the TOPSAW computer program to compare virtual boards generated by analyzing X-ray/CT log images, with the actual boards sawn from the same position in the same log at a sawmill. In one simulation when the boards are graded, both virtual and actual, it was found that the value of the virtual boards is only 3% less than that of the actual boards. Thus, the authors showed that it was possible to assist the sawing process based on internal log structure obtained from

X-ray/CT imagery.

Zhu and Beex [100] experimented with a stochastic texture modeling method for a machine vision log inspection system using computerized tomography (CT) imaging to locate and identify internal defects in hardwood logs. In one simulation, correlation-classification was conducted with a training set, and the resulting classification accuracies were 71.4%, 100%, and 100% for decay regions, bark regions, and knots, respectively. Defect recognition accuracies obtained with the testing set are: 80% for bark regions with 3 out of 15 bark regions misclassified as decay regions; 81% for decay regions with 4 out of 14 decays misclassified as knots; and 100% for knot regions with no misclassification.

Zhu et al. [102] further developed a computer vision system for locating and identifying internal defects in hardwood logs using CT imagery. The algorithm consists of a number of processing steps:

1. an adaptive filter smooths each 2-D CT image to eliminate annual ring structure while preserving other details;

2. a multithreshold 2-D segmentation scheme is used to separate potential defect areas from areas of clear wood on each image;

3. by generalizing 8-neighbor connectivity to 3-D structures, sequences of consecutive and segmented 2-D slices are then analyzed to find connected 3-D regions.

To deal with the imprecision and ambiguity in assigning labels to the 3-D regions, a set of hypothesis tests were employed that used a set of basic features capturing common 3-D characteristics of wood defects, and the Dempster-Shafer theory of evidential reasoning was used to classify defect objects. No quantitation information was given with respect to the performance of this system.

Using CT technology, Li et al. [40] investigated internal log inspection and developed a feed-forward multilayer Artificial Neural Network (ANN) system, trained by a back-

propagation method. ANN includes a training phase and an operation phase. The ANN classifier used here is the Multi-Layer Perceptron (MLP) architecture trained using the back-propagation algorithm [12]. A perceptron can learn from examples, and needs be trained to recognize the correct input vectors. By normalizing CT density values, these classifiers can accommodate several hardwood species such as northern red oak, water oak, yellow poplar, and black cherry. They can also accommodate three common defect types such as knots, splits, and decay. Local 3-D data are used to extract defect features, and a pixel-by-pixel classification accuracy of 95% was achieved. Analysis of a CT slice with 256×256 elements, each corresponding to a volumne of $2.5 \times 2.5 \times 2.5mm^3$, on a Macintosh Quadra 650 with a MC680403/33MHz CPU requires about 25 seconds. Sarigul et al. [74, 73] further refined these classifiers in a subsequent post-processing step, by developing a rule-based approach to region refinement to augment the initial emphasis on local information. The resulting rules are domain dependent, utilizing information that depends on region shape and defect type. Compared to ANN, the Intellipost system developed by Sarigul et al. improved segmentation accuracy for hardwood log datasets were 1.92% for the red oak datasets and 9.45% for the datasets provided by Forintek Canada, Inc [73]. For the case of medical datasets, improvement for two datasets were 4.22% and 0.33%, respectively. Similar excution time as ANN is expected.

Bhandarkar et al. [3] developed CATALOG, a system for detection and classification of internal defects in hardwood logs via analysis of computer tomography images. Defect detection and classification in CATALOG consists of two phases:

1. Segmentation of a single CT image slice, resulting in the extraction of 2-D defect-like regions;

2. Correlation of the 2-D defect-like regions across CT image slices in order to establish 3-D support.

The segmentation algorithm includes multiple-value thresholding that exploits both the

knowledge of wood structure, and the gray-scale characteristics of the image. The extraction algorithm locates the pith of a log cross section, groups pixels in the segmented image based on their connectivity, and classifies each 2-D region as a defect or non-defect region using shape, orientation, and morphological features. From the cross-section CT images, CATLOG performs 3-D reconstruction and rendering of the log and its internal defects. It also simulates and renders key machine operations such as sawing and veneering. Overall, the entire process of defect identification, defect localization, 3D model reconstruction, and rendering on a 200-MHz PentiumPro workstation with 256 MB of RAM took between 3 and 4 minutes for all the log species that were considered. The graphical simulation of the sawing operation averaged 38 seconds for a cut defined by two sawing surfaces. The graphical simulation of the rotary-peeled veneering operation averaged 8 seconds for a veneer of length 1.2 meters and width of 1 meter. No quantitative evaluation of the Catalog system was given, yet it was claimed to be capable of detection and 3D rendering of defects such as knots, cracks, holes and bark/moisture pockets in hardwood logs of select hardwood species. The species that were considered were Red Oak, Black Walnut, White Ash and Hard Maple, which account for over 80% of the lumber poduction in the United States.

Tian et al. [93, 94] developed an automated camera-based vision system based on texture analysis that can locate and identify certain classes of defects on freshly harvested Radiata pine logs (a type of softwood). The system applies the algorithm computing the orientation field for a flow-like texture, originally developed by Rao et al. [61]. The basic structure of their system consists of a feature extraction module estimating an oriented texture field based on the original tree stem image, and an object analysis and recognition module for processing the oriented texture field. Visual texture is defined as repeating patterns of local variations in image intensity that are too fine to be distinguished as separate objects at the observed resolution [32]. The system uses an texture-oriented filter that analyzes gradients using a 2-D Gaussian function. It is made up with two modules:

1. A feature extraction module for estimating the oriented texture field from the raw

image of a log surface;

2. A scene analysis and detection module for analyzing the oriented texture field.

The system is able to accurately detect different types of defects on barked log surfaces. The key difference between the experiments by Tian et al. and this research is the original input data, where the former used gray-scale digital images of softwoods, and the latter, three-dimensional surface profile data of hardwoods. Also, Tian was looking only for pruned branch stubs and overgrown pruned branches. This research explores a wider range of defect types. Tian's system can accurately detect more than 95% of knot positions and more than 90% of knot sizes. The system named KnotVision was programmed using Borland's C/C++ and Borland's Turbo Assembler with an image processing and analysis library devloped in Tian's research. However, no details on equipment or execution time were given.

Kline et al. [36] applied a method to evaluate the performance of color camera machine vision in automated furniture rough mill systems. 134 red oak boards were used to compare the performance of automated gang-rip-first rough mill yield based on a color camera lumber inspection system with both estimated optimum yield and actual measure yield. Three sawing patterns were studied, including gang-ripsaw, ripsaw, and chopsaw. For each sawing pattern, board area that the system classifies as clear is reported, which is compared to the observed clear area. Defect detection accuracy was measured in terms of false negative error and false positive error. False negative error was defined as defect regions on the board that the scanning system classified as clear wood. False positive error was defined as actual clear wood region that the scanning system classified as defect.

2.2 Estimation Methods for Circle Fitting

Fitting geometrical model to given data in the plane or space involves minimization of the sum of squared distances between the data and the model using least-squares methods. Such

distances include algebraic, geometric, and orthogonal distances [17, 6]. Let $f(x) = 0$ denote a 2-D curve, and $x1$ denote a 2-D data point. Then $f(x1)$ is the so-called algebraic distance. Geometric distances depend on the type of curves. For example, a circle is a closed conic curve with a center. Assume the center is p. Now let $x1$ denote a 2-D data point, and $x1 \neq p$. Let l denote the line passing through both $x1$ and p and intersecting the circle at $x2$. There are two intersections between l and the circle, and $x2$ is the closest one to $x1$. Then the geometric distance is defined by the distance between $x1$ and $x2$. The orthogonal distance between the point $x1$ and the curve $f(x) = 0$ is the radius of the smallest circle centered at $x1$, which is tangent to the curve [6]. From the above definitions, the minimization algorithms in order of increasing complexity are: algebraic, geometric, and orthogonal distances. The computation intensity increases in the same order as well. Orthogonal distance minimization, or regression, is advocated when errors exist not only in the dependent variables, but also in the independent variables, and is a method to minimize both errors.

Many least-squares algorithms and software minimize the sums of squared algebraic, geometric, and orthogonal distances; some of them apply weighted least-squares methods. However, there is no mechanism in these algorithms capable of identifying severe outliers to correctly estimate the model parameters. Robust estimators have been applied in many fields. Classes of robust estimators include A, D, L, M, P, R, S, and W estimators [21]. The M-estimators consist of many varieties, for example the conventional Least-Squares estimators, Least Absolute Value estimators, and the Huber estimator. The S estimators include the Least Median of Squares estimators (LMS) and Least Trimmed Squares estimators (LTS). For image structure analysis, Meer et al. [45, 46] applied the LMS estimator to recover piecewise polynomial surface fits. The LMS estimator is robust against outliers up to 50% of the image data.

Under the Gaussian distribution, LMS estimates are less accurate than GME estimates, since they only uses the middle residual value and hence assumes that the data set contains a 50% fraction of noise [70]. When the data set contains less than 50% noise, the LMS estimates suffer in terms of accuracy, since not all the good points are used in the estimation.

This is a major drawback. The LMS estimator can be applied to provide initial conditions for the GM algorithm. However, this class of estimators is typically implemented via computationally intensive algorithms that are inappropriate for the defect detection application. For the application in our research, we would liket to investigate algorithms that have fast execution time. In sawmills, the average time for an operator to inspect a log is 9 seconds. Thus, the detection system should be no slower than a human. To circumvent this difficulty, we developed a simple and very fast method based on the log data characteristics, which provides reasonably good initial conditions. Section 5.2 discusses the method in detail.

In the region based surface and shape-fitting techniques, Besl et al. proposed robust rectangular constant-coefficient window operators for performing local image smoothing and determining derivative estimation for edge detection [2]. The theory of robust statistics is applied, and a variable order surface approximation algorithm was developed that includes model identification. Parameters are tuned for redesending M estimators using weight functions, pixels having similar properties are grouped together, and the smoothing across discontinuities is prevented. Mainguy et al. [42] further applied Monte Carlo simulation in the study of M estimation and LMS estimation for piecewise continuous image surface approximation, and proposed a variable order facet model paradigm in M estimation. Robust M estimators and their variants have been found to be tolerant to occlusion and other outlier contamination, and more computationally efficient than high breakdown operators. Thus, they are currently gaining popularity in computer vision.

The data for the research work proposed here are log surface measurements containing 3-D coordinates. Essentially they are range data, not the intensity values commonly referred to as gray scales, which is 2-D. For example, Haralick et al. [24] defined the topographic primal sketch for gray-scale intensity images. Harris [26] developed the coupled depth/slope model and tested on synthetic gray-scale surface data. Terzopoulus et al. [84] proposed quasi-symmetric 3-D shape models applicable for both gray-scale images and 3-D range data. An active contour model named "snake" was developed by Kass et al. [33] and tested on gray-scale and other type of 2-D images. The snake model was further improved by Cohen [9].

Experiments were performed on gray-scale images rendered from various medical images.

Our algorithm identifies objects in 3-D range data images, which requires different techniques from those developed for gray-scale images. Many algorithms have been developed for gray-scale images with pixel values commonly between 0 and 255, a few were for range images. In computer vision, surface reconstruction, comprising surface interpolation and approximation algorithms, fits a smooth surface to the image data to determine features such as slope, and orientation based on the image intensity. Haralick et al. [24] gave a complete treatment for describing the topographic primal sketch of the underlying gray-scale intensity surface of a digital image. Eight main shapes are described, each has a unique label and is invariant, for example, peak, ridge, ravine, saddle, hillside. A 2-D cubic polynomial of the facet model is fitted to estimate the image surface. Tests for the model were performed on synthetic images and scene images of manufactured objects.

Harris [26] developed the coupled depth/slope model that explicitly computes the slope and depth representations, and allows for varying amounts of smoothness. The author applied finite difference approximations to derive a parallel and iterative algorithm from the model, which was tested on synthetic gray-scale surface data. More details about these methods are introduced in the remaining section. Mainguy et al. [42] applied robust statistical procedures to study the underlying piecewise continuous surface of a gray-scale image and proposed a robust variable order facet model. The image was tested both with and without added noises at different levels.

To handle outliers, Kim et al. suggested using robust techniques with a relatively high efficiency [35]. A Breakdown point can be used to measure robust algorithms, which is the smallest fraction of outliers present in the input data that may cause the output estimate to be arbitrarily wrong. For instance, L_1, L_2, and L_p estimators have a breakdown point at $1/n$, where n is the number of data items. Another measure of robust statistical procedures is their "relative efficiency" defined by Kim et al in [35], as the ratio between the lowest achievable variance for the estimated parameters (the Cramér-Rao bound), and the actual

variance provided by the given method, so that the best possible value is 1. The Cramér-Rao inequality, named in honor of Harald Cramér and Calyampudi Radhakrishna Rao, expresses a lower bound on the variance of a statistical estimator, based on Fisher information [21]. Kim et al. also note that the least mean squares estimator in the presence of Gaussian noise has an asymptotic (large sample) efficiency of 1, while the least median squares estimator's efficiency is only 0.637.

In statistical estimation, there is a trade-off between algorithms with high breakdown points versus those with high efficiency. Further, most research in robust statistics was done for linear problems. To ensure that robust techniques work for solving nonlinear problems, one needs to carefully choose the initial estimate values, such that they are close enough to the true solution. In this document, we present neither the breakdown point, nor efficiency measure of our new GM-estimator. Instead, we prove theoretically in Section 5.4 that it is robust by deriving its influence function. Our nonlinear GM-estimator, proposed in Chapter 5, applies an iteratively reweighted least squares algorithm. It starts with a simple but effective initial estimate, making it robust, efficient, and effective.

Tirumalai and Schunck [95] also introduced a robust statistical LMS regression algorithm for surface approximation using least median of squares regression. Quadratic surface fitting was performed on monocular as well as binocular stereo 2-D data. Rao and Schunck [61] proposed oriented texture analysis methods and experimented with both synthetic gray-scale images, and real images of manufactured parts with relatively simple geometric shapes. Taubin [82] addressed the problem of parametric representation and estimation of complex planar 2-D curves and 3-D surfaces. Simulations were performed on both gray-scale images and 3-D range data. Both images captured man-made objects. An algorithm estimating the parameters of a linear model in presence of heteroscedastic noise employs errors-in-variables (EIV) model arising from the linearization of bilinear form [38]. It fits ellipses and achieves accuracy of nonlinear optimization at low computational cost. Synthetic 2-D data as well as a bridge gray-scale image were evaluated in experiments. Matei and Meer [44] proposed an improved maximum likelihood estimator for ellipse fitting based on the heteroscedastic

EIV regression algorithm, which was tested on synthetic data as well as gray-scale images of man-made objects.

The defect detection algorithm analyzes 3-D lot data, extracts features such as length width, surface rise, gradients, and identifies objects described by 3-D Cartesian coordinates. Various methods were proposed in 3-D objects description and identification. In the domain of dynamic 3-D modeling and 3-D object reconstruction, Terzopoulus et al. [84] proposed quasi-symmetric 3-D shape models that can be considered as deformable bodies made of elastic material. Such models are active because they change shape by attaining stable equilibrium between the internal energy of the model and external forces from the image. The model stops changing only when the energy function is minimized and the shape in the image is determined. The model was tested on two images of real objects with quasi-symmetric features (squash, potato, and pear). Such a model was further developed to fit complex 3-D shapes using a superquadrical model that can deform both locally and globally [83]. Superquadrical objects are 3-D, whose equations (in Cartesian coordinates) are of the second or higher degree. They were first discovered by Hein [18]. Take superellipsoids for example. A special class of superellipsoids are the familiar ellipsoids. One may express an ellipsoid centered at the origin, with a, b, and c representing the three semiaxes, respectively, as $x^2/a^2 + y^2/b^2 + z^2/c^2 = 1$. Now the superellipsoids take the generalized form $x^n/a^n + y^n/b^n + z^n/c^n = 1$, where $n \geq 2$ is a real number, and .

The systems introduced in [84, 83] were executed at interactive rates on a graphic workstation. The dynamic equations make the models responsive to forces derived from image or range data, and compels them to conform to the data. This model is suitable for detecting shapes with relatively smooth surfaces, for instance, eggshells and mugs. An active contour model named "snake" was developed by Kass et al. [33], which is an energy-minimizing closed spline, analogous to the 3-D deformable model. It can be used to detect vision objects, such as edges, lines, and contours. Various types of 2-D images were tested for the model. The snake model was further improved by Cohen [9] to give more stable results, and the curve behaves like a balloon to guarantee the algorithm converges to the correct solution. Exper-

imentation with medical images, such as ultrasound, MRI, of human internal organs, was performed to illustrate the technique. When an operator observes that the initial curve lies inside an image object, which is to be detected, the system identifies the curve. The balloon technique is effective in that it is able to expand the curve to fit the object boundary.

ODRPACK developed by Boggs et al. is a mathematical software for solving weighted orthogonal distance regression problems [5]. The algorithm for ODRPACK finds parameters that minimize the sum of the squared weighted orthogonal distances from the data to a curve or surface [6]. It implements an efficient and stable trust region (Levenberg-Marquardt) procedure. The algorithm minimizes both model and measurement errors. However the influence of extreme outliers cannot be downweighted.

2.3 Relationship between External and Internal Defects

During the past 50 years there has been a significant amount of research conducted examining the relationship of external hardwood log defect indicators to internal defect characteristics. The majority of internal defects are where a branch has been slowly grown around or over, to form a sound knot defect. Depending on specie, knots can be the same color as surrounding wood, but are usually somewhat darker. Internal knots are characterized by a tight circular grain pattern contrasted from the straight grain of the surrounding clear wood. This change in grain pattern creates a weakness in the wood that is transferred to any board cut that contains the knot. For any given knot, it is largest near the surface, and tapers more or less uniformly to a point at the center of the log. Knot size is highly variable between examples, and ranges from less than .5-inch to nearly as large as the log they are contained on. In general, the larger and more knots a log has, the lower its grade and dollar value. Internal defects may also be rotten or decayed, often referred to as unsound. Although less frequent than sound knots, unsound internal defects are more serious, and lower the log value and its

products more so than sound knot defects. Unsound defects start as knots, wounds, holes or splits, that decay due to the introduction of bacteria. In most cases, an unsound defect started out as a knot from a branch that died or was broken off and the tree could not grow over it quick enough before the onset of decay. Unsound defects are typically larger than sound knots due to the nature of decay spreading in wood. Holes and splits are the least frequent internal defects, and are often associated with insects/animals or harvesting damage. When they are 2 inches long or more, they have a significant impact on value and strength of the boards produced. In the smallest example, the log could have small worm holes which have little or no impact on value. In the worst example, the log could be cracked nearly in half due to poor harvesting technique.

Several guides and pictorial series have been published illustrating various external and internal defect characteristics and their relationship for various hardwood species [43, 62, 63, 64, 65, 68, 66, 67]. While these guides are useful references for providing insight on the external/internal relationship, only one or two examples of each defect type are provided. Thus, while informative, they do not fulfill the need of a definitive model capable of predicting internal defect features based on observable external defect features. Further, most studies are limited in scope with small samples and examine a narrow range of defect types and features.

Hyvärinen used Marden's maple defect data to explore the relationships among the internal features of grain orientation and height of clear wood above an encapsulated knot defect and the external features of surface rise, width, and length [31]. The sugar maple defect data were collected from 44 trees covering three sites in upper Michigan. Hyvärinen used simple linear regression methods to find good correlations among clear wood above defects, bark distortion width, length, and rise measurements, as well as age, tree diameter, and stem taper. However, the best simple correlation was with diameter inside bark (DIB) ($r = .66$) and a 0.66-inch standard error of estimate. A coefficient of correlation of .74 and a standard error of 0.60 inch were obtained using a stepwise regression method with bark distortion vertical size and DIB variables being the most significant indicators.

A similar study was conducted on a sample of 21 black spruce trees collected from a natural stand 75 km north of Quebec City [39]. Three trees, each with three logs, were selected from which a total of 249 knot defects were dissected and their data recorded. The researchers found better correlations between external indicator and internal characteristics in the middle and bottom logs as compared to the upper logs. Strong correlations ($r > .89$) were found to exist between external features such as branch stub diameter and length to the width and length of internal defect zones. The defects were modeled as having three distinct zones, corresponding to the manner in which the penetration angle changes over time in black spruce. This study examined only branches that had not been pruned or dropped and thus could not examine encapsulation depth. Encapsulation depth refers to the amount of clear wood that has grown over a defect. The greater the encapsulation depth, the greater the opportunity for a clear board to be sawn from wood over the knot.

2.3.1 Defects need to be detected both externally and internally

One of the major areas of study today in hardwood research is the development of equipment and a methodology that can accurately sense internal defect locations and structures. Determining the location and characteristics of defects located inside logs promises to dramatically improve current log recovery in terms of both quantity and quality. In addition, accurate internal defect information would permit researchers to refine, expand, and analyze log grading rules, multi-product potential, stand differences, and silvicultural treatments in ways previously not available or economically feasible.

Studies have demonstrated that the use of external or internal defect data improves cutting strategies that optimize log recovery or yield, that is, preserving the largest possible area of clear wood on a board face [79]. The value of the lumber that can be recovered depends on the presence and location of defects. This is especially true for hardwood logs. In the production of hardwood lumber, boards are sawn to fixed thicknesses and random widths. The presence and placement of defects on the boards affect board quality and value, so

much attention is focused on log surface defects during processing. Thus, while detecting external defects is useful for determining overall log quality characteristics, internal defect information is the key to improving lumber value and volume in the sawmill.

2.3.2 External defects have a high correlation with internal defects

A recent study has discovered strong relationships between external defect features and internal defect characteristics for severe defects: overgrown knots, sound knots, knot clusters, and unsound knots [90]. This study harvested a total of 66 yellow-poplar trees from two sites separated by approximately 220 miles. 300 severe knot defects were randomly sampled from the trees. The samples were dissected and measured. A series of stepwise multiple-linear regression analyses were performed to determine if any significant correlations between external and internal features existed.

In most instances, strong correlations were found to exist among external defect indicators and internal characteristics for severe defect types: overgrown knots, overgrown knot clusters, sound knots, and unsound knots. The number of overgrown knot cluster defects was not sufficient sample size for establishing a defect prediction model. Because of this, overgrown knot clusters and overgrown knots observations were grouped together.

The correlation results for severe defects are shown in Table 2.1. The strength of the correlations (adjusted multiple R^2) between interior halfway point width measurement and exterior features ranged from 0.48 to 0.75. Similar results were found to exist among external features and the halfway point length measurement (adjusted multiple R^2 from 0.45 to 0.75). Most of the severe defect observations terminated at the pith, approximately the center of the slab for most samples. This is demonstrated in the strong relationship among penetration depth and external features - specifically diameter, with adjusted multiple R^2 ranging from 0.63 to 0.81. The strongest correlation with penetration angle was with sound knots (adjusted multiple $R^2 = 0.70$). However, in most cases, the relationship between penetration angle and external features was not as strong with adjusted multiple R^2 ranging from 0.23 to 0.39

Table 2.1: *External/ internal defect correlation results*

Defect type	**Internal defect feature**	**Adjusted R squared**	**Mean absolute error**	**Residual standard error**
Overgrown Knot	Halfway width	0.49	0.28	0.36
	Halfway length	0.45	0.50	0.63
	Penetration angle	0.39	7.79	10.49
	Depth	0.76	0.41	0.56
Overgrown Knot / Overgrown Knot Cluster	Halfway width	0.47	0.27	0.36
	Halfway length	0.46	0.48	0.59
	Penetration angle	0.22	11.13	13.82
	Depth	0.73	0.42	0.59
Sound Knot	Halfway width	0.75	0.31	0.42
	Halfway length	0.75	0.53	0.76
	Penetration angle	0.70	8.75	11.33
	Depth	0.63	0.41	0.54
Unsound Knot	Halfway width	0.71	0.26	0.39
	Halfway length	0.65	0.67	0.93
	Penetration angle	0.39	8.16	10.79
	Depth	0.74	0.45	0.65

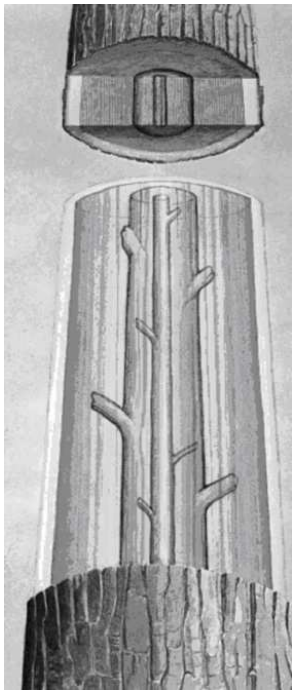
for the other severe knot defects. All correlations between external indicators and internal features were significant at the 99% level. Further, the low mean absolute errors (0.25 to 0.70 inch) indicate that internal features can be reliably predicted. Additional testing is planned to determine if the error rates would affect processing decisions based on the inferred internal information.

Chapter 3

Defect Taxonomy

3.1 The Structure and Nature of Log Defects

A tree should be thought of as having multiple layers. Every growing season the tree produces a completely new layer of wood and bark tissue. In a sense a new layer envelops the old tree every year [75]. As the tree builds the new layers, places where wounds have occurred or branches have fallen or been sawn off are overgrown. Figure 3.1 shows a cutaway view of a tree showing several selected layers and associated overgrown branches. The pictures in Figure 3.1 were extracted from [76]. It is by growing in this way that the tree protects itself from animal, insect, and bacterial invasions. Thus, defects formed on hardwood logs are a response to the natural growth process or to damage. The most serious and common log surface defects consist of sound knots, unsound (rotten) knots, overgrown knots, medium and heavy bark distortions, and holes [25]. Less common, but quite severe are wounds and splits. Knot type defects can appear clustered together and are more serious than a defect appearing singly, as the underlying wood is more defective. More common, but less severe are adventitious knots. In most cases adventitious knots are not regarded as a log defect, unless they have developed into a branch.



(a) Selected layers and branch stubs that have been overgrown in a log

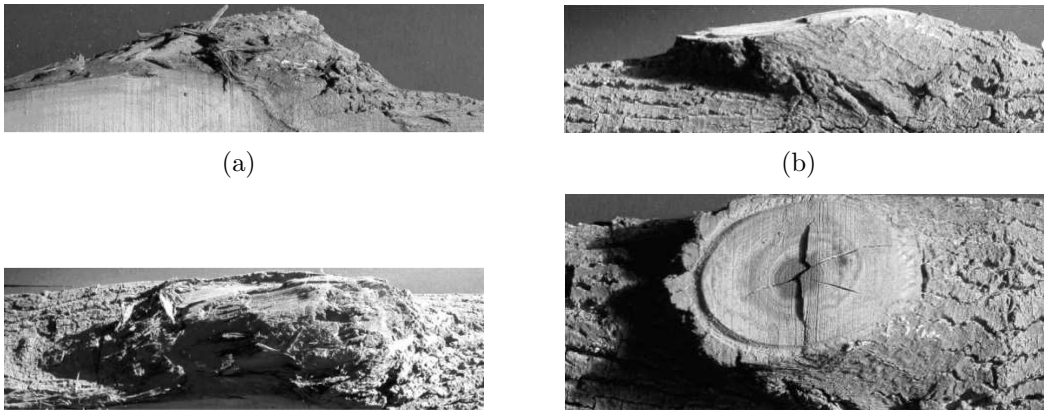
(b) Broken branch stub showing jagged break and hanging loose bark

Figure 3.1: *A cutaway view of a tree and associated overgrown branches (taken from [76].*

3.2 Branch-Related Defects

The formation of defects related to branches and knots follows a logical progression. In the first phase the branch is pruned, falls off naturally, or is torn away by natural causes, leaving a sound knot defect (Figure 3.1(b)). This leaves an abruptly raised, round area on the log surface. If the branch was naturally removed, as in a wind storm, the surface would be rough (Figures 3.2(a) and 3.2(c)). A smooth surface would have been left if it had been sawn off (Figures 3.2(b) and 3.2(d)). Knots from branch stubs can vary in size, from a few square inches in surface area to a square foot or more.

In the next phase, the sound knot (sawn or pruned branch stub) is grown over with bark and some underlying wood to yield an overgrown knot. Here the area is still significantly



(a) (b)

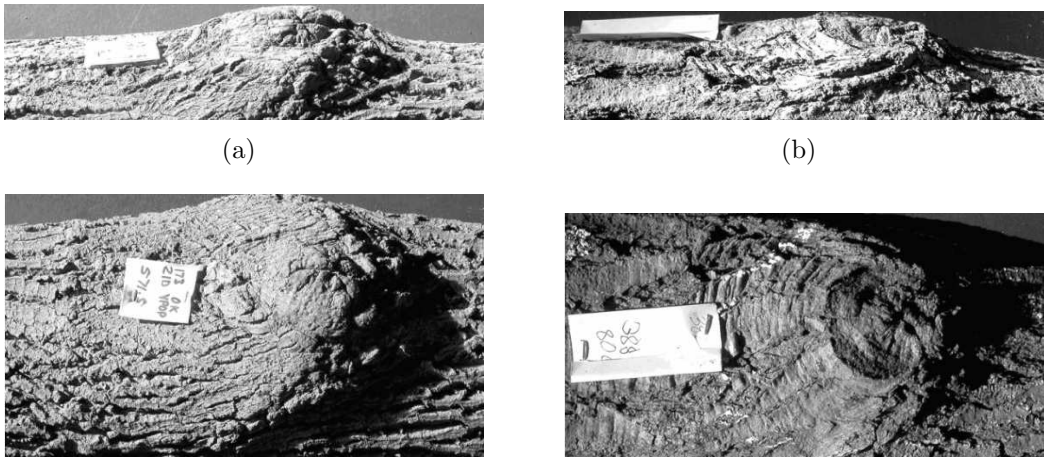
(c) (d)

Figure 3.2: *(a) and (c): side view and top view of a broken branch stub from a log. (b) and (d): side view and top view of a sawn branch stub from a log. Both are sound knots.*

higher than the surrounding wood. The bark texture of the new bark over the branch stub is smooth and usually rounded. Figure 3.3 shows two examples of overgrown knots. As the tree continues to grow, the height difference between the knot and the surrounding area decreases.

If a bacterial or viral infection occurs before the tree can completely grow over the branch stub, then an unsound knot can occur. Unsound knots have much the same overall shape and characteristics as a branch stub with the exception of a rotten area usually in the middle of the defect. The rotten area can be a hole or an exposed piece of the original branch showing signs of decay. Figures 3.4(a) and 3.4(c) show an example of an unsound knot where the branch stub has rotted away and left a hole. Figure 3.4(b) and 3.4(d) show an unsound knot that was nearly grown over, but has an exposed rotten part of the branch stub remaining.

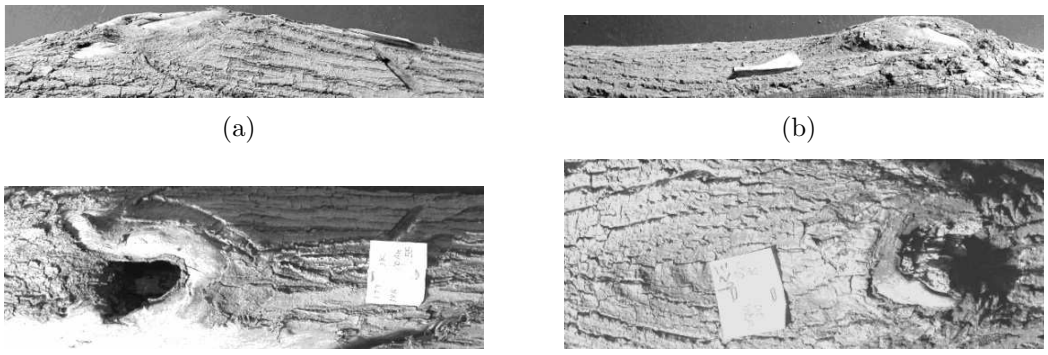
If the tree is successful in growing over the branch stub, then the overgrown knot will eventually become a heavy distortion defect. The heavy distortion looks like a flattened version of the overgrown knot. It is characterized by at least a single heavy circular ring in the bark texture. Figure 3.5 shows a heavy distortion defect from a red oak log. In this



(a) (b)

(c) (d)

Figure 3.3: *(a) and (c): side view and top view of an overgrown knot on a yellow poplar. (b) and (d): side view and top view of another overgrown knot on a red oak.*



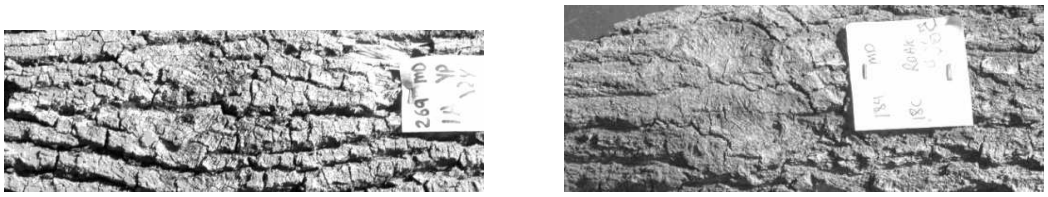
(a) (b)

(c) (d)

Figure 3.4: *(a) and (c): side view and top view of an unsound knot, a hole where branch stub has rotted away on a red oak log. (b) and (d): side view and top view of another unsound knot on yellow poplar showing rotten remains of branch stub.*

(a) (b)

Figure 3.5: *Typical heavy distortion defects showing the circular ring of bark tissue. (a) is on a red oak log, and (b) a yellow-poplar log.*



(a) (b)

Figure 3.6: *Medium distortion defects. (a) is on a yellow-poplar log, and (b) is on a red oak log.*

example there is an inner circle of smoother, younger bark which grew over the branch stub. The circular ring of the defect is easy to discern from the straight lines of the normal bark texture. At this point the branch stub is just under the bark surface.

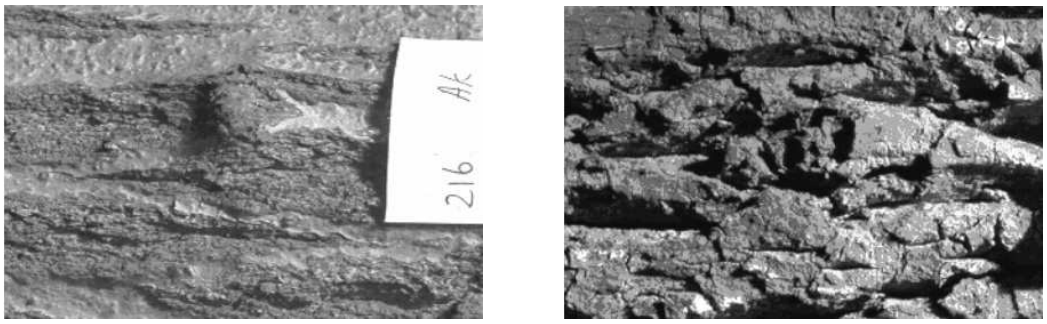
Gradually the heavy distortion will become a medium distortion. The branch stub has been overgrown to the point that it can be several inches beneath the log surface. Medium distortions lack the strong circular area of heavy distortions. The centers of medium distortions are generally more broken up and the circular area may be split in two. In general medium distortions are less circular and harder to find on the bark than heavy distortions. Figure 3.6 shows two medium distortion defects. In the center of Figure 3.6(a) a semi-circular area of disjoint bark can be seen.

Eventually the medium distortion defect will become a light distortion. Light distortion

defects are simply a slight break in the texture of the bark. These defects can be difficult to find, even by experienced loggers and processors. Because these defects indicate a defect near the center of the tree or one that is many inches below the log surface and do not affect the value or utility of the log, they are not regarded as a defect. Eventually the light distortion and all evidence of the overgrown branch will fade from the bark.

Adventitious buds or knots can be quite common on some hardwood log species. Adventitious buds exist in a dormant state within the tree until conditions are right for the bud to sprout into a branch. Such conditions can be initiated by damage to the tree, such as the loss of several branches, or a neighboring tree has been removed exposing the bark to sunlight. Adventitious knots range in diameter from less than 0.25 inch to more than 2 inches and average approximately 1 to 1.5 inches. They are characterized by a small circular ring distortion of the bark, and the center can be raised 0.25 inch or more. In very minor examples, the indicator of an adventitious bud is simply a small, 0.25×0.25 inch^2, or smaller, rounded raised point. Figure 3.7 shows two examples of adventitious buds. Figure 3.8(a) shows an example where a small branch has started from the adventitious bud and has been cut off. Only in the case where a branch has started are these defects considered serious. Branches from adventitious buds are called suckers. If the sucker is successful it will grow into a branch, otherwise it will become a branch stub, which will form an overgrown or unsound knot depending on its circumstances. Although a sawn-off sucker may resemble a sound knot, there are key differences. Specifically the size of a sound knot is generally much larger, 4 inch^2 or more in surface area, compared to 1 to 2 inch^2 for a sucker. In addition, the area around a sucker remains flat with little surrounding height change. A sound knot often raises a large area of surrounding bark (Figure 3.2).

All of the knot defects mentioned above can occur in clusters. Clusters of defects are regarded as more serious than a single occurrence. As the internal defect manifestation is more severe. The common names for clustered defects are adventitious knot cluster (Figure 3.8(b)), sound knot cluster, overgrown knot cluster, unsound knot cluster. In general, the cluster defects have the same characteristics as comparable single examples. However, cluster



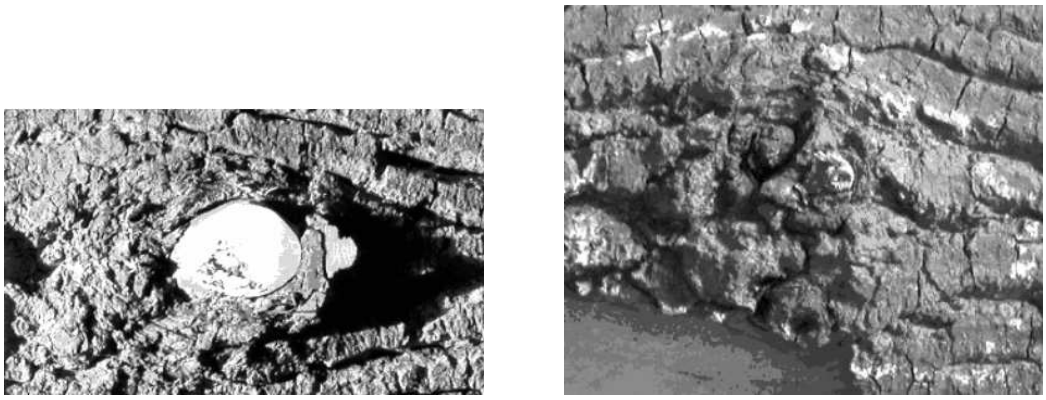
(a) (b)

Figure 3.7: *Adventitious knot examples. (a) is on a red oak log, and (b) is on a yellow-poplar log.*

defects can have greater surface rise since the branch defects are growing out and over each other. In addition, due to the extra branches and competition among the branches, the bark will be more heavily distorted and the area of the distortion will be wider.

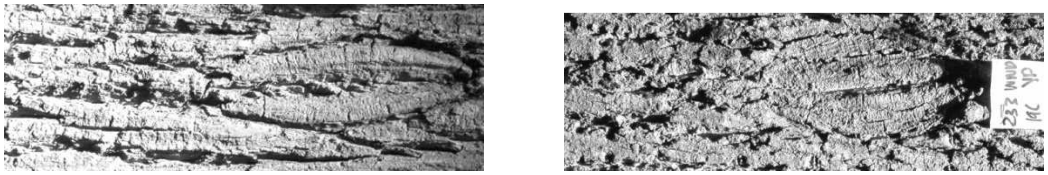
3.3 Damage Defects

Damage defects include the defect classes of holes and wounds. Holes are abrupt depressions into the log surface. The surrounding bark can be completely normal with no distortion or other indicator of a defect. Holes are most often caused by animals, insects, or decay. Figures 3.4(a) and 3.4(c) showed a hole defect in the middle of an unsound knot. Wounds are where damage to the bark surface has occurred. Like holes the surrounding bark can appear completely normal. Depending on the severity of the wound and how much wood was removed, a depression can exist in the middle of the wound. Figure 3.9 shows two examples of wounds. Normally a wound is characterized by smooth bark with a split down the middle. The split is the meeting point of the bark tissue when it grew over the wound.



(a) (b)

Figure 3.8: *(a). An adventitious bud that developed into a branch. This type of branch is sometimes referred to as a "sucker". This picture shows the sawn branch stub. (b). A cluster of adventitious knots on a yellow poplar log.*



(a) (b)

Figure 3.9: *Two examples of wounds on yellow poplar logs.*

3.4 Defect Taxonomy From the Laser-Data Perspective

As discussed in Sections 3.1 through 3.3, there are many external defects on hardwood logs, including sound knots, overgrown knots, unsound knots, holes, gouges, bumps, close bird beck, adventitious knots and their clusters, hard distortion, and medium distortion, and wounds. These standardized external defect descriptions were developed by a group of forest research scientists for log grading rules [7]. To better distinguish external log defect types that are useful for the laser-scanned 3-D data, we now categorize them from a different point of view. The **knob** defect type includes both the sound and unsound overgrown knots by

Table 3.1: *Defect taxonomy and characteristics from the laser-data perspective.*

Code	Name	Average Height
KNOB	Knob	1.5
SWK	Sawn Knot	1
HOLE	Hole	1.3
AKC	Adventitious Knot Cluster	0.5
HD	Heavy Distortion	
MD	Medium Distortion	
WOUND	Wound	
LOOSE BARK	Loose Bark	4.5

traditional forestry definition, which are referred to as overgrown knots and unsound knots, respectively. The **sawn knot** defect type includes both the sound and unsound sawn knots by traditional forestry definition, which are referred to as sound knots and unsound knots, respectively. The reason that we categorize these two types in such a way is that, using the 3-D laser data there is no significant distinction between sound and unsound overgrown knots, or between sound and unsound sawn knots. At this stage, we group them into knobs and sawn knots. Putting them together allows us to analyze their characteristics, such as length, width, and surface rise, which are used in our detection algorithm development.

Table 3.1 presents defect taxonomy and characteristics from the laser-data perspective. The measurements are collected from about 200 real external defect samples of both red oak and yellow poplar. Note the defect types are listed in decreasing order of the height (surface rise). The following are the indicators and definitions of defect types listed in Table 3.1.

Knob

Indicator An abrupt surface rise (usually .5 inch or more) and texture change 2 to 8 inches in diameter. Some may have a surface rise with a depression or hole in the middle.

Definition Indicates a knot just below the bark surface. Some may have a portion rotten.

Sawn Knot

Indicator An abrupt surface rise (usually .5 inch or more) and texture change 2 to 8 inches in diameter that is characterized by a flat sawn top. Some may have a surface rise with a depression or hole in the middle.

Definition Location where a branch has been sawn off of the log. Some may have a portion rotten.

Hole

Indicator An abrupt circular surface depression ($\geq$ 1.5 inches in diameter and 2 inches in depth). The edges of the hole may have surface rise.

Definition A hole is most often rotten. A hole can be result from a branch that dropped off and rotted back into the tree. A hole can also be caused by animals, which will eventually become rotten. A severe defect because of the staining and decay associated with the defect.

Adventitious Knot Cluster

Indicator A grouping of two or more Adventitious Knots. Can be associated with small distortion defects representing past AK's that have sprouted, fallen off, and been overgrown.

Definition More severe than a single adventitious knot. A group of suppressed buds that will develop into branches when conditions are favorable.

Heavy Distortion

Indicator Slight surface rise and circular texture pattern consisting of several concentric rings. The horizontal and vertical diameters of the defect are approximately equal.

Definition A heavy distortion is knot (branch stub) that has been recently completely overgrown by the surrounding wood.

Medium Distortion

Indicator Circular texture pattern consisting of one or two circular rings that have been broken by the background bark texture. The horizontal diameter of the defect is usually noticeably greater than the vertical diameter.

Definition A medium distortion is a knot or branch stub that has been overgrown to the point that it is now several inches below the log surface.

Wound

Indicator A scar on the bark with no surface rise, usually elongated with a center seam where the edges of the wound grew together. Depending on the severity of the damage the bark may have a slight depression.

Definition Damage to the bark and possible underlying wood caused by insects, bacteria, animals, or past logging operations.

Table 3.2: *Statistics of defect measurements categorized from the laser-data perspective. The units are inches, and the format for the data is: first quartile-median-third quartile.*

Type	Width	Length	Surface rise	Surface Depression
KNOB	5.0-5.5-6.8	5.5-6.5-8.3	1.0-1.5-1.5	None
SWK	4.9-6.3-8.6	6.5-9.5-10.5	0.5-1.0-1.5	0.6-1.0-1.0
HOLE	5.5-5.5-5.5	8.9-9.3-9.6	1.1-1.3-1.4	1.9-2.8-3.6
AKC	3.9-5.0-5.1	4.0-4.3-5.0	0.5-0.5-0.5	0.0-0.0-0.0
HD	4.5-4.8-5.3	4.0-5.0-5.1	0.5-0.5-0.5	0.5-0.5-0.5
MD	3.5-3.5-3.6	2.9-3.0-3.1	0.5-0.5-0.5	None
LOOSE BARK	1.5-2.0-3.5	6.3-9.5-15.5	2.5-4.5-6.8	None

Loose Bark

Indicator Bark pieces dangle or protrude from the log surface. Generally they are long and narrow bark strips or fallen leaves, with one of the narrow ends attached to the log. 1 - 2 inches wide, and 2 - 10 inches.

Definition Leaves or sections of bark torn or loosened during harvesting and/or handling that are attached to log surface.

There are other defect types, such as gouges, that are possible to detect yet extremely rare. Due to the difficulty in collecting sample data, they are not listed here. Since clusters of sawn knots and those of knobs can be detected and classified as individual defects, here we omit them. Next, we discuss each defect type using the defect data collected and analyzed based on the measurements of the defect sample collection. Table 3.2 contains statistics obtain from the sample collection. The defect types are listed in decreasing order of the surface rise. For several defect types, their measurements and statistics are absent because they are trivial or not available due to the nature of such defect types. The format for the data is: first quartile-median-third quartile. The information was analyzed for establishing defect models in the defect detection algorithm discussed in Chapter 6.

Note that although the defect median height in Table 3.2. Although they seem to be not trivial, for example, 1.5 inches for knobs and 1.0 inches for sawn knots, the height was measured as the highest point of the defect. In our contour-based defect detection algorithm, very likely only a small portion of the defects (represented by the corresponding radial distances) are enclosed in the contour. Therefore, the relative significant median heights in above table do not indicate an sure sign of correct identification of the defects. Nonetheless, the metric indicates the likelihood for the contour-based detection algorithm to locate the defects. Evidently, knobs and sawn knots are the majority to be identify. The median height for holes is 1.3 inches. Such a height is cause by the "ridge" surrounding it. This seems to suggest that we are likely to detect many holes. However, statistics show that the percentage of holes in external defects are very low, which is about 1%. These data referenced for the detection algorithm development. We refer to them as training data.

Chapter 4

Overview of the Detection Algorithm

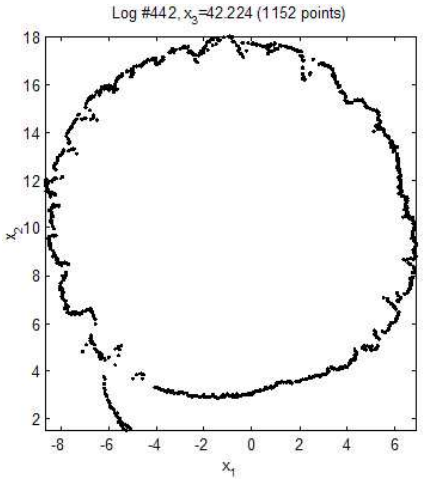
Severe external defects that correspond to rises or depressions on the log surface can be observed from the three-dimensional log surface image. This suggests that one way to determine their location is to extract the height change on the log surface from its 3-D image. To do so, a series of circle fitting to log cross-section data sets were applied to obtain ground zero reference levels of the log surface. Because the laser range data sets may include either missing data or irrelevant deviant data points, a new, robust estimator was developed to estimate in a reliable manner the centers and the radii of the fitted circles. Radial distances between the latter and the log data points are thus indicative of the local height changes. Defects characterized by significant (in a statistical sense) surface rises or depressions are then located using appropriate statistical methods. The following sections give an overview of the GM-Estimator in circle fitting, as well as an overview of the defect detection algorithm. Both are discussed in details in Chapters 5 and 6, respectively.

4.1 Fitting Circles to Log Data Using a New GM-Estimator

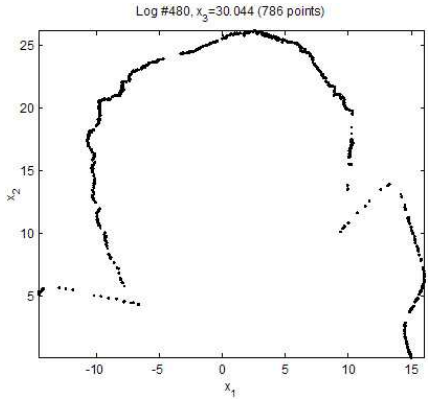
To convert the 3-D log surface data to 2-D images for processing, a reference surface must be imposed on the log data from the scanner. Since logs are natural objects that are approximately circular or elliptical along the cross sections, circle- and ellipses-fitting to log data were experimented with. Fitted circles and ellipses all together form a reference surface, or virtual log, that is needed for defect detection. Defects that correspond to rises or depressions on the log surface can be detected using contour levels estimated from the orthogonal distances between the virtual log surface and any point of the cross section.

Fitting quadratic curves (i.e., circles, ellipses) to 2-D data points is a nonlinear regression problem [17]. Classic least-squares fitting methods fail in our case because the laser log cross-section data contain either missing data and/or large deviant data points, termed outliers in the statistical literature. These data characteristics are caused by both logs and the scanning system. As depicted in Figure 4.1, the laser data sets include deviant data generated by dangling loose bark, duplicate and/or missing data caused by scanner calibration errors, unwanted data from the supporting structure under the log, and missing data due to the blockage of the log by the supporting structure. In robust statistics, outliers are defined as data points that strongly deviate from the pattern formed by the majority of the measurements. To overcome the non-robustness of the least-square fitting, we resort to the theories and methods of robust statistics [21]. The nonlinear form of the circle equation prompt us to develop a new, robust estimation method that is an outgrowth of the one proposed by Mili et al. [48].

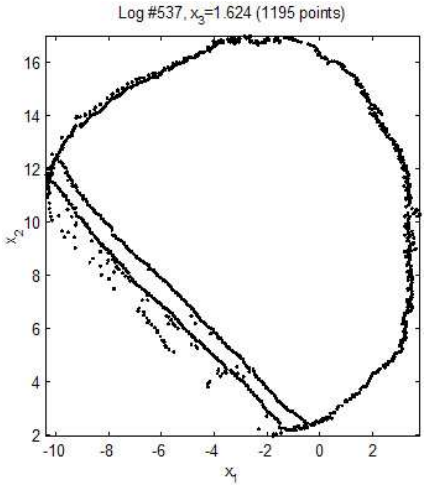
The nonlinear regression circle-fitting estimator is a generalized M-estimator termed GM-estimator for short [86]. As shown in Figure 4.2, it filters out not only the errors in the measurements, but also the errors in the circle model that is applied to a given cross-section data set. For example, for a log sample with 120 cross sections, an equal number of circles



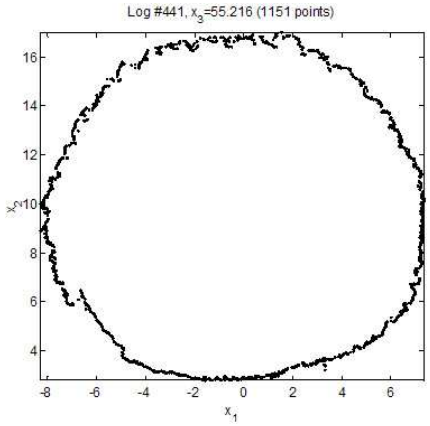
(a) loose bark flakes in lower left corner



(b) Outliers in form of scanning support structure and missing data due to structure



(c) Outliers and shape of log at one end where the log was cut diagonally instead of squarely



(d) A good log data cross section containing no outliers

Figure 4.1: *Various formations of outliers present in cross-section data from laser scanning.*

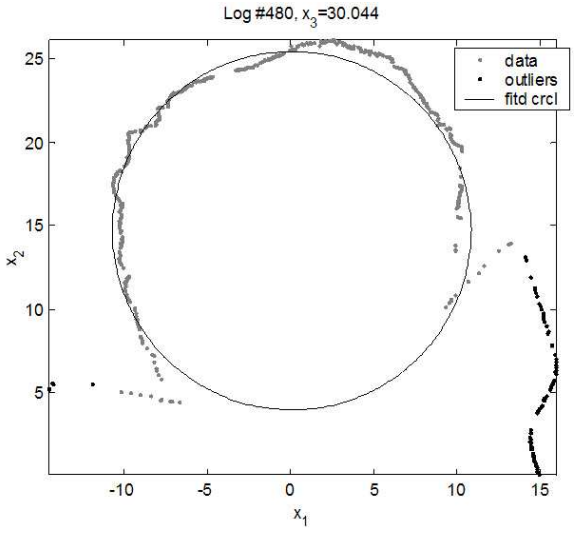


Figure 4.2: *Circle fitting to a cross section that contains a portion of the log support.*

are fitted, forming a virtual log for the radial-distance extraction as depicted in Figure 4.3. Unlike the method described in [48], the estimator minimizes an objective function that makes use of a weight function that levels off for large scaled radial distance between the associated data point and the fitted circle. It does this at every step of the iterative algorithm that solves the estimator. The robust measure of the scale of these distances is performed by means of projection statistics [19, 49, 71] while the minimum of the objective function is found through the iteratively re-weighted least-squares algorithm [29]. Chapter 5 provides detailed information regarding the robust circle-fitting GM-Estimator.

To check that the nonlinear circle-fitting GM-Estimator is robust against outliers, its influence function was derived, which is a measure of the estimator's sensitivity to infinitesimal data contamination [21]. If this function increases without bounds as a data point is moved farther and farther away from its true value, the estimator is said to be non-robust; otherwise it is said to be robust. It can be shown that the influence function of our estimator can be decomposed as the product of two terms, one reflecting the influence of model (i.e., the circle equation), and another reflecting the influence of measurement errors (i.e., radial distances). It can be shown that both terms are bounded, making the estimator robust against extreme

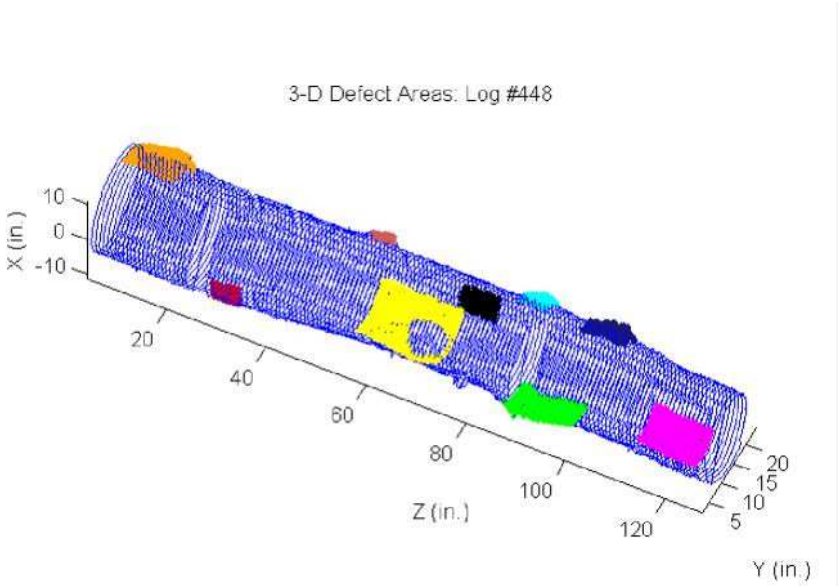


Figure 4.3: *A 3-D rendering of the log data with automatically detected defects marked by patches. Such an image might be used by sawyers to maximize the value of wood products.*

outliers.

The robustness of the estimator was tested on real log data samples. It was found that the resulting fitted circles vary little among neighboring cross sections. This yields a smooth fitting over the entire data of one log. Figure 4.2 displays a circle that was fitted to a cross section with a non negligible fraction of outliers and missing data. Outliers identified by this method are plotted in bold. The smoothness of the fitting is further reinforced by smoothing the parameters using a box filter [23]. Note that approximately 3 percent of the points are labeled as outliers, and hence suppressed from the data set.

4.2 Generating the residual gray-level image

The next step is converting the three-dimensional laser-scanned Cartesian coordinates into a two-dimensional, 256 gray-level image (Figure 4.4). In this process, the log surface is unrolled onto a 2-D coordinate space. In essence, this process creates a "skin" of the log surface representing the pattern of log bark along with bumps and bulges associated with most defects. Using the adjusted, fitted circle to each cross section, radial distances were calculated between circle and log surface points, typically ranging from -0.5 to 0.5 inch. The radial distances are scaled to range from 0 to 255 and mapped to gray-levels to create a 2-D image. Originally the log data are not in a grid format, so they are processed and interpolated linearly to fill any gaps between data points. The x_3 value in 3-D data is the coordinate in the third dimension or the z-axis value, which is the position along the log's length. It is mapped to the 2-D image as the x_2 value, given by a row number. The x_1 value of the image, given by a column number, is calculated by scaling the angle of a cross section's point from the center of fitted circle.

If the desired image is to be 750 pixels wide, the scaling factor would be $750/(2\pi)$. On average, the size of an unrolled log output image is about 2 MB (Mega Bytes), or $1,400 \times 1,600$ pixels at 1 byte per pixel. To save space and future processing time, the resolution of output gray-level image from log-data unrolling is reduced. The Gaussian pyramid algorithm [23] is applied and a 5×5 window is used to smooth and subsample the image. The image is reduced to 25 percent of the original size, that is, roughly 500 KB/image. Since the density in a cross section is nearly 20 times that of along the log length, only data in cross sections are reduced, the total number of cross sections is not reduced. This speeds additional analysis of the image with little or no loss of data of interest.

Experiments with fitting ellipses to the log data showed that while each individual ellipse does generate radial distances, resulting radial distances tend to reveal more surface details, hence the log surface "height" map contains more undesirable information, primarily due to the difference of axes orientation between neighboring ellipses. The resulting image tends to

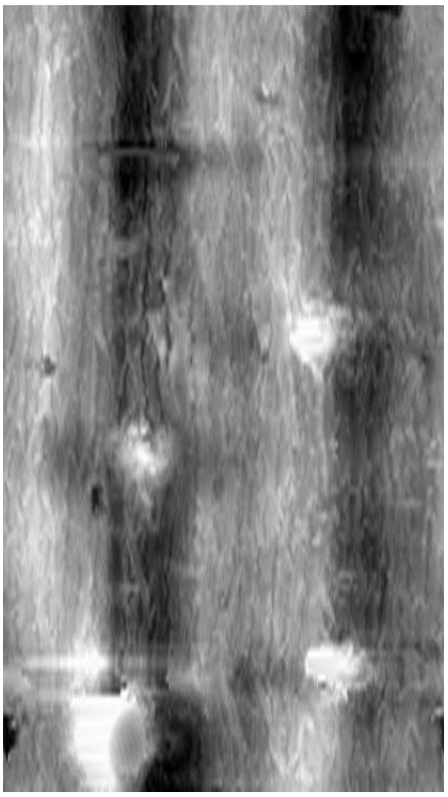


Figure 4.4: *Radial distances generated by the log-unrolling process presented as a gray-level range image. Light pixels represent protrusions from the log surface, and dark pixels represent depressions. This log is approximately 9 feet in length with a diameter of 2 feet.*

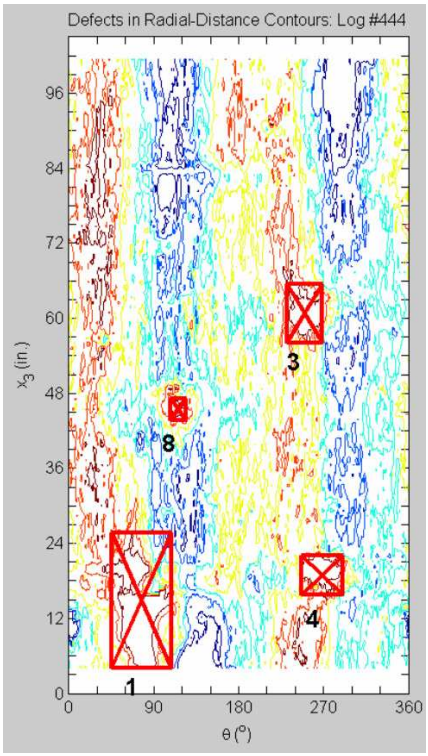
be noisier. Unlike white noise, the noise has low frequency. It adds unwanted information that camouflages true defective regions. However, fitting only circles to the data does cause the rolling or striping effect in the height map along the cross-section direction, as shown in Figure 4.4. As log cross sections are generally round, they are often not totally round. Nor they are perfectly elliptical. Thus, radial distances extracted from circle fitting inevitably introduce the striping effect in many cases. We have attempted to resolve the issue by applying a filtering method to reduce overall radial distances along the rising strips, and increase them along the dipping ones. However, since logs come in unlimited number of shapes, this method causes unwanted side effect which is worse than the striping one. This is a complicated issue that can be addressed in future research.

4.3 Identifying Defects Based on the Radial Distances

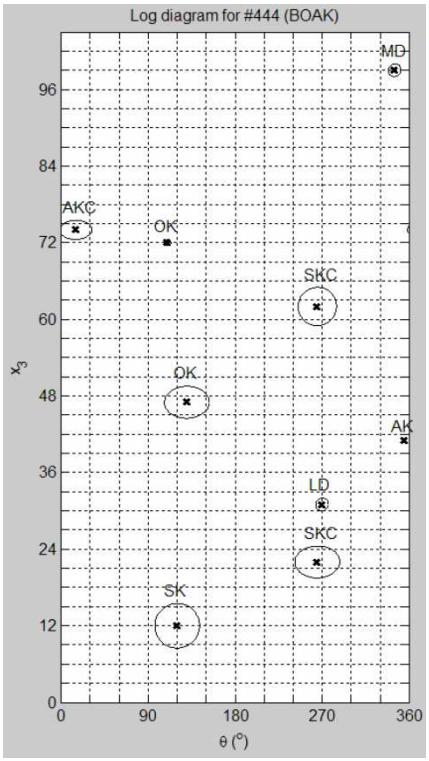
To accommodate the countless possible defect sizes, heights, shapes, types, etc. in the 3-D log data, we developed a machine vision system to implement the defect detection task. The current version of our system uses the contour image generated from the radial distances, which provides a map of defect height change against the surrounding bark. Also used are the measured 3-D log data. Expert knowledge is applied in a stepwise fashion to rule out regions as potential defects, including regions in sizes smaller than a given threshold, nested in other curves, or long and narrow (determined by the "actual" width to length ratio, referred to as w/l for short). By "actual" we refer to the width to length ratio acquired through the calculation of the statistical medium of the widths of the region enclosed in the selected contour curve.

The data resolution (0.8 inches per cross section) and the nature of external defect shapes restrict search scope in the algorithm. The ones visible through the log data are the most obvious defects based on their external characteristics, such as protrusion on surface, certain width-length ratio, and area. They have a relatively significant height change on the surface ($\geq$0.5 inches), and/or a relatively significant size ($\geq$3 inches in diameter). Using radial distances visualized by the gray-level image in Figure 4.4, the algorithm generates a contour plot as depicted in Figure 4.5, and determines rectangle-enclosed regions. The rectangles are bounding boxes of contour curves at the highest level. Then some regions are selected if they are big enough or with a significant height. More detailed discussion of the algorithm is found in Chapter 6. In Figure 4.5 four out of the nine surface defects are found using this method. Figure 4.5 also shows a manually recorded map of the defects on the same log. The defect types represented in the map include SKCs (sound knot clusters) and OKs (overgrown knots).

Further, the algorithm includes a statistical procedure to examine the region surrounding a selected small region for relatively straight line segments. If the coverage of straight line segments is sufficient, the defect region is adjusted to cover the entire defect surface, rather



(a) Contour plot of a log surface with the four most-obvious defect regions marked with crossed rectangles labeled in the descending order of area



(b) Defect diagram illustrating the "ground truth"

Figure 4.5: *Contour plot and the "ground truth". Note that only five small and/or flat defects were not detected. Both plots were generated by Matlab programs, while defect regions in (a) were determined by the detection algorithm.*

than just a corner. The algorithm examines angle changes between the lines connecting log data points along cross section at certain intervals. If the changes are small enough ($\leq 25^o$), the corresponding segments are recorded as nearly straight. Then the coverage of the "straight" segments is determined. If there are a sufficient number of straight segments, this region is identified as a flattop, which is likely a sawn top, either sound (not rotten), or unsound (rotten).

Many severe defects are associated with a localized height change, a height analysis of the residual image provides information about the presence of such severe defects. A substantial,

localized, and abrupt surface rise or depression greater than 1.0 inch is almost always a defect. The reason 3 inches was chosen as the threshold for defect diameter is that the log-data resolution-0.8 inch per cross section-is not high enough to well capture defects whose diameters are smaller than that. Since the pixel values in the gray-level image represent radial distances between the fitted circle and the log surface, the analysis is straightforward. In the contour plot image, it is possible to discern regions containing likely defects based on height information alone.

Region-removal rules are given as: regions smaller than a given threshold are mainly tiny fragments; regions enclosed in curves nested in other curves are removed, as there will only be up to one defect in the same location; those being long and narrow are normal bark regions; regions that are smaller than 50 inch^2 and are too close to the selected large ones. Some regions are removed for further consideration if they contain a severe portion of missing data. Although not illustrated in Figure 4.5, certain defects, in particular the sawn ones, are often detected partially in the contour. This is because they are relatively low-lying and flat, and often only a small portion of a sawn knot, for example, a relatively high-raised corner, is enclosed in the highest contour. The algorithm adjusts the boundaries of this type of identified regions. Regions may include elevated yet non-defective log surface. Typically they are covered with tree bark, thus associated with distinctive bark patterns. Finally, due to the lack of "depressed" defect samples in the log data, at this stage of development the system does not detect such defect types.

Chapter 5

A Novel Robust GM-Estimator

A typical log size is 10 to 20 inches in diameter and 8 to 16 feet long. The scanned data density is about 0.04×0.78 inch^2 per point. Typically, each cross section of log data can be approximated by a closed curve resembling a circle or an ellipse. Hence, one of the problems that we dealt with is to fit a quadratic curve or surface to the recorded log data. It turns out that these data are corrupted by gross errors as bark on logs often becomes loose, forming flakes. Furthermore, the supporting structure underneath the log blocks the scanner, causes missing data, and the shape of the structure can be seen in the scanned images.

Statistically, measurements with large errors, known as outliers, can be regarded as observations that deviate from the pattern formed by the majority of the data set. Consequently, classical estimators based on the least-squares method cannot be used here to carry out curve or surface fitting because they generate incorrect estimates in presence of outliers. We need instead to resort to robust statistics as initiated by Huber [30]. This is a collection of theories aimed at designing estimators and statistical tests that enjoy a certain degree of insensitivity to departure from the assumptions, including resistance to outliers [30, 21, 70]. Based on these theories, a new generalized M-estimator was developed to fit circles to log data, which is able to downweight all types of outliers, hence bounding their influence on the estimates. The corresponding regression models were developed to extract residuals for

further analysis.

5.1 The New Estimator

To obtain a good circle fitting to the recorded data for a given log cross-section, a new generalized M-estimator was developed and an algorithm proposed that implements it. We show that it is influence bounded and robust against all types of outliers. Outliers are data that are far apart from the main bulk of data. For an estimator, when the outliers move farther and farther away from the main bulk, two cases could happen. The first is that no matter how far away the outliers move, the estimator still converge to the correct solution. The second is that the estimator diverges as the outliers move away. An estimator could diverge even when the fraction of contamination is infinitively small. The estimator in the first case is considered influence bounded, while the one in the second case, not influence bounded.

The 3-D log surface data consist of a collection of 3-dimensional range data points grouped as circular-shaped cross sections from the scanner. Each cross section has the same x_3 value. Let $\{\tilde{\mathbf{x}}_1, \tilde{\mathbf{x}}_2, \ldots, \tilde{\mathbf{x}}_m\}$ denote the set of data points of a given cross section, where $\tilde{\mathbf{x}}_i = [x_{i1}, x_{i2}, x_{i3}]^T$ for $i = 1, \ldots, m$. Our intension is to fit a circle to these data points, which all lie on a plane defined by a constant third coordinate, x_3. Note that the Boggs et al. considered the general case of finding orthogonal distances to a curve [6]. For the circle fitting, the radial distance are easily calculated, because they are along the radius of the fitted circle. Thus, it is unnecessary to use an iterative algorithm to calculate them. By contrast, the ODRPACK software developed by Boggs et al. [5] is appropriate for ellipse fitting. Now on the plane defined by a constant third coordinate, x_3, one can define a nonlinear regression model given by

$$(x_{i1} - p_1 + \eta_{i1})^2 + (x_{i2} - p_2 + \eta_{i2})^2 - p_3^2 + e_i = 0, \tag{5.1}$$

where $p = [p_1, p_2, p_3]^T$ is the parameter vector containing the center coordinates (p_1, p_2) and

the radius p_3 of the circle, and where $\mathbf{x}_i = [x_{i1}, x_{i2}]^T$ is the two-dimensional measurement vector in the cross section under consideration. In Equation 5.1, the measurement error vector is defined as $\boldsymbol{\eta}_i = [\eta_{i1}, \eta_{i2}]^T$, while the model error is denoted by e_i and accounts for the uncertainty in the assumed circle model. Note that this uncertainty exists even if the measurements are perfect. The model given by Equation 5.1 can be written in a compact form as

$$f_i(\mathbf{p}, \mathbf{x}_i, \boldsymbol{\eta}_i) + e_i = 0, \text{for } i = 1, \ldots, m. \tag{5.2}$$

The problem is hence to robustly estimate the parameter vector $\mathbf{p}$ in Equation 5.2 from a 2-dimensional measurement vector $\mathbf{x} = [x_{11}, x_{12}, \ldots, x_{m1}, x_{m2}]^T$. For this model, conventional M-estimators are not robust because their influence function is not bounded for the error vector, $\boldsymbol{\eta}_i$, as it is shown in Section 5.4. A Schweppe-type generalized M-estimator is more appropriate here. Termed GM-estimator for short, this estimator minimizes an objective function of the form

$$J(\mathbf{p}) = \sum_{i=1}^{m} w_i^2 \rho(\frac{r_i}{s\, w_i}). \tag{5.3}$$

Here $\rho(\cdot)$ is the Huber function expressed as

$$\rho(\frac{r_i}{s\, w_i}) = \begin{cases} \frac{1}{2}(\frac{r_i}{s\, w_i})^2 & \text{for } |\frac{r_i}{s\, w_i}| \leq b \\ b|\frac{r_i}{s\, w_i}| - \frac{b^2}{2} & \text{for } |\frac{r_i}{s\, w_i}| > b \end{cases}, \tag{5.4}$$

and the residual r_i is defined as

$$r_i = -h_i(\mathbf{p}, \mathbf{x}_i), \tag{5.5}$$

with

$$h_i(\mathbf{p}, \mathbf{x}_i) = (x_{i1} - p_1)^2 + (x_{i2} - p_2)^2 - p_3^2. \tag{5.6}$$

Note that the only difference between the two functions, $h_i(\mathbf{p}, \mathbf{x}_i)$ and $f_i(\mathbf{p}, \mathbf{x}_i, \boldsymbol{\eta}_i)$, is the presence of the measurement error vector, $\boldsymbol{\eta}_i$, in the latter. Pick $b = 1.5$ in Equation 5.4 to have a good statistical efficiency at the Gaussian distribution while not increasing too

much the bias under contamination [30, 21]. Writing Equation 5.5 in compact form for $i = 1, \ldots, m$, one can get the m-dimensional residual vector $r = -h(\mathbf{p}, \mathbf{x})$, where $h(\cdot)$ is an m-dimensional vector-valued function. In Equation 5.3, s is a robust estimator of scale of the residuals given by $s = 1.483\text{median}_i|r_i|$, and $w_i \in (0, 1]$ is an appropriate weight function that makes the estimator robust against outliers in $\mathbf{x}_i$. The w_i are being introduced to bound the influence of the measurement errors, $\boldsymbol{\eta}_i$ in the model given by Equations 5.1 and 5.2. The errors $\mathbf{e}$ and $\boldsymbol{\eta}_i$ are assumed to follow the ϵ-contaminated model, $F = (1 - \epsilon)\Phi + \epsilon H$, where $0 \leq \epsilon \leq 1$. It defines a full neighborhood of the Gaussian probability distribution, Φ, which includes asymmetric distributions. For small ϵ, this model indicates that there is a large fraction $(1 - \epsilon)$ of the errors that follow Φ while the remaining fraction, ϵ, follow an unknown distribution, H. Such a model will be used in Section 5.4 to derive the influence function of the GM-estimator.

The estimator $\hat{\mathbf{p}}$ is a solution to

$$\frac{\partial J(\mathbf{p})}{\partial \mathbf{p}}|_{\boldsymbol{p}=\hat{\boldsymbol{p}}} = \sum_{i=1}^{m} \frac{w_i}{s} \frac{\partial \rho\left(\frac{r_i}{s\, w_i}\right)}{\partial\left(\frac{r_i}{s\, w_i}\right)} \frac{\partial r_i}{\partial \mathbf{p}} = \mathbf{0}. \tag{5.7}$$

Assuming that w_i is constant in the neighborhood of $\mathbf{p}$ and defining the scalar function $\psi(u) = \frac{\partial \rho(u)}{\partial(u)}$, then

$$\sum_{i=1}^{m} w_i \mathbf{H}_i(\mathbf{p}, \mathbf{x}) \psi\left(\frac{r_i}{s\, w_i}\right) = \mathbf{0}. \tag{5.8}$$

The vector $\mathbf{H}_i(\mathbf{p}, \mathbf{x})$ in Equation 5.8 denotes the transpose of the ith row of the $m \times 3$ Jacobian matrix $\mathbf{H}(\mathbf{p}, \mathbf{x})$ given by

$$\mathbf{H}(\mathbf{p}, \mathbf{x}) = \frac{\partial \mathbf{h}(\mathbf{p}, \mathbf{x})}{\partial \mathbf{p}} = -2 \begin{bmatrix} x_{11} - p_1 & x_{12} - p_2 & p_3 \\ x_{21} - p_1 & x_{22} - p_2 & p_3 \\ \vdots & \vdots & \vdots \\ x_{m1} - p_1 & x_{m2} - p_2 & p_3 \end{bmatrix}. \tag{5.9}$$

The function w_i is calculated based on the projection statistics defined in Section 5.3.2. It is such that it equals one for a good measurement $\mathbf{x}_i$ and decreases asymptotically to zero as the radial distance of $\mathbf{x}_i$ to the fitted circle increases beyond a given threshold. Consequently, the objective function given by Equations 5.3 and 5.4 will not down-weight a good measurement with small standardized residual, $r_i/(s\,w_i)$, because in this case the term $w_i^2\rho(r_i/(s\,w_i))$ in Equation 5.3 reduces to $r_i^2/(2\,s^2)$; but for an outlier, it becomes $b|w_i\,r_i/s| - (b\,w_i)^2/2$, down-weighting it. Thus, the estimator is influence-bounded; this property will be made clearer in Section 5.4 by showing that its influence function is indeed bounded.

5.2 The Iteratively Reweighted Least-Squares Algorithm

A solution to Equation 5.8 is found through the iteratively reweighted least-squares (IRLS) algorithm [30, 29]. To derive its expression, first divide and multiply the $\psi-$function in Equation 5.8 by the standardized residual to get

$$\sum_{i=1}^{m} w_i \mathbf{H}_i(\mathbf{p},\mathbf{x})\; q\left(\frac{r_i}{s\,w_i}\right)\; \frac{r_i}{s\,w_i} = \mathbf{0}, \tag{5.10}$$

where $q\,(r_i/s\,w_i) = \psi\,(r_i/s\,w_i)\,/\,(r_i/s\,w_i)$. Then, put Equation 5.10 in a matrix form to get

$$\mathbf{H}(\mathbf{p},\mathbf{x})^T\;\mathbf{Q}\;\mathbf{h}(\mathbf{p},\mathbf{x}) = \mathbf{0}, \tag{5.11}$$

where $\mathbf{Q} = \mathrm{diag}(q(r_i/s\,w_i))$ is a $m \times m$ weight matrix. Performing a first-order Taylor series expansion of $\mathbf{h}(\mathbf{p},\mathbf{x}_i)$ about the value of $\mathbf{p}$ obtained at the kth iteration, $\mathbf{p}^{(k)}$, gives

$$\mathbf{h}(\mathbf{p},\mathbf{x}) \approx \mathbf{h}(\mathbf{p}^{(k)},\mathbf{x}) + \mathbf{H}(\mathbf{p}^{(k)},\mathbf{x})\;(\mathbf{p}-\mathbf{p}^{(k)}). \tag{5.12}$$

Substituting Equation 5.12 into Equation 5.11, and putting $\mathbf{p} = \mathbf{p}^{(k+1)}$ to obtain

$$\mathbf{p}^{(k+1)} = \mathbf{p}^{(k)} + [\mathbf{H}(\mathbf{p}^{(k)}, \mathbf{x})^T \, \mathbf{Q}^{(k)} \, \mathbf{H}(\mathbf{p}^{(k)}, \mathbf{x})]^{-1} \mathbf{H}(\mathbf{p}^{(k)}, \mathbf{x})^T \, \mathbf{Q}^{(k)} r^{(k)}. \tag{5.13}$$

The initial conditions for the IRLS algorithm given by Equation 5.13 are not determined by the conventional least-squares method [17]. This is because the latter provides a solution that is too biased due to the action of severe outliers, especially those that stems from the supporting scanner structure under the log. One alternative method would be to resort to the least median of squares estimator or any other high breakdown estimator [69]. However, this class of estimators is typically implemented via computational intensive algorithms that are inappropriate for this application. To circumvent this difficulty, a simple and very fast method was developed based on the log data characteristics, which provides reasonably good initial conditions. It consists of the following three steps:

1. Identify all the cross sections that have a sufficiently large number of data points, say larger or equal to 80% of the average number of data points per cross section; the remaining cross sections are considered corrupted and will be excluded from the computation in the next two steps.

2. For each of these cross sections, pick as an estimate of the x_1 and x_2 coordinates of its center, the midpoints of the minimum and maximum values along the x_1 and x_2 axes, respectively; pick as an estimate of its radius, the midpoint of the width and the height of the bounding rectangle.

3. Smooth out the center point values and radii by replacing each of them with the corresponding averages taken over three consecutive cross sections, known as box filter [23].

5.3 Defining the Weight Function w

Unlike the GM-Estimator developed for linear regression, the weights w_i in Equation 5.3 are not calculated from the residuals, r_i given by Equation 5.5, which are algebraic distances; they are rather determined from the radial distances between the data points and the circle. Furthermore, they are evaluated in a robust manner by means of the projection statistics, which can be viewed as a robust version of the classical Mahalanobis distances of a collection of points in n-dimensions.

The above mentioned radial distances are defined as follows. Let $\mathbf{c} = [p_1, p_2]^T$ denote the center of the circle and let $\mathbf{d}_i = [d_{i1}, d_{i2}]^T$ denote the radial vector between the point $\mathbf{x}_i$ and the circle with radius p_3. The vector $\mathbf{d}_i$ is then given by $\mathbf{d}_i = (\mathbf{x}_i - \mathbf{c})(1 - p_3/\left\|\mathbf{x}_i - \mathbf{c}\right\|)$, where $\|\mathbf{u}\|$ stands for the magnitude of a vector $\mathbf{u}$. The vectors $\{\mathbf{d}_1, \mathbf{d}_2, \ldots, \mathbf{d}_m\}$ identify a point cloud in a plane.

5.3.1 Classical Outlier Identification Methods based on Mahalanobis Distances

The conventional method for identifying outliers makes use of the Mahalanobis distances. In statistics, Mahalanobis distance is a distance measure introduced by P. C. Mahalanobis in 1936. It is based on correlations between variables, by which different patterns can be identified and analyzed. It is a useful way of determining similarity of an unknown sample set to a known one. It differs from Euclidean distance in that it takes into account the correlations of the data set and is scale-invariant, that is, not dependent on the scale of measurements. Formally, the Mahalanobis distance from a collection of m points in n-dimensions, $\{\mathbf{d}_i, i = 1, \ldots, m\}$, with the sample mean $\hat{\boldsymbol{\mu}} = \sum_{i=1}^{m} \mathbf{d}_i/m$, and the sample covariance matrix $\hat{\sum} = \sum_{i=1}^{m}(\mathbf{d}_i - \hat{\boldsymbol{\mu}})(\mathbf{d}_i - \hat{\boldsymbol{\mu}})^T/(m-1)$ is defined as:

$$MD_i = \sqrt{(\mathbf{d}_i - \hat{\boldsymbol{\mu}})^T \hat{\sum}^{-1} (\mathbf{d}_i - \hat{\boldsymbol{\mu}})}.$$

A well known result is that when the $\mathbf{d}_i$'s are drawn from a multivariate normal distribution $N(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, the MD_i^2 follow approximately a chi-squared distribution with n degrees of freedom, χ_n^2 [1]. Therefore, there is a probability of approximately 97.5% that a point $\mathbf{d}_i$ will fall inside the tolerance ellipsoid given by $MD^2 = \chi^2_{n,0.975}$. A sensible approach would then be to flag as deviant points, termed outliers, all the data points that fall outside that ellipsoid. While this method seems to be reasonable at first glance, it is unfortunately prone to the masking effect of multiple outliers because the sample mean is attracted by them and the sample covariance matrix is inflated to the extent that some or all of them may fall inside the tolerance ellipsoid.

5.3.2 Robust Outlier Identification Based on Projection Statistics

Initiated independently by Stahel and Donoho in 1982 [78, 11], the projection method was inspired by the following equivalent expression of the Mahalanobis distance:

$$MD_i = \max_{\|\mathbf{v}\|=1} \frac{|\mathbf{d}_i^T \mathbf{v} - L(\mathbf{d}_1^T \mathbf{v}, \ldots, \mathbf{d}_m^T \mathbf{v})|}{S(\mathbf{d}_1^T \mathbf{v}, \ldots, \mathbf{d}_m^T \mathbf{v})}, \tag{5.14}$$

where L and S are respectively the sample mean and the sample standard deviation of the projections of data points $\mathbf{d}_i$ on the direction of vector $\mathbf{v}$ and where the maximum is taken over all the possible directions. A robust version of Equation 5.14 is then obtained in a straightforward manner by replacing L and S by robust statistics, for example by the sample median and the Median-Absolute-Deviation from the median (MAD) of the projections.

A practical implementation of this method was advocated by Gasko and Donoho [19], who proposed to investigate only those directions originating from the coordinate-wise median $\mathbf{M}$

of the point cloud and passing through each of the data points, yielding a total of m directions to be examined. The directional vector of a data point $\mathbf{d}_i$ is defined as $\mathbf{v}_i$, $i = 1,\ m$. Termed projection statistic, the resulting estimate for a data point, say the ith point, is indicative of the distances that it has with respect to the bulk of the point cloud in the worst one-dimensional projection. Formally, it is defined as

$$PS_i = \max_{\|\mathbf{v}_i\|=1} \frac{|\mathbf{d}_i^T\mathbf{v}_i - med_j(\mathbf{d}_j^T\mathbf{v}_i)|}{1.4826\, med_k|\mathbf{d}_k^T\mathbf{v}_i - med_j(\mathbf{d}_j^T\mathbf{v}_i)|}.$$

The algorithm that calculates projection statistics can be found in Section 5.5. Note that this estimator is different from the one proposed by Mili et al. [49] for power system state estimation as here PS_i is determined based on the radial vector, while in the latter, it is based on the row vectors of the Jacobian matrix $\mathbf{H}(\mathbf{p}, \mathbf{x})$ given by Equation 5.9, which revealed to be not robust in this application. The weights w_i are calculated from the projection statistics, which are a robust version of the Mahalanobis distances. The PS_i accounts for the correlations between the radial distances. ODRPACK does not calculate weights as we are proposing here. For ellipse fitting, we may run ODRPACK to find the radial distances, and then calculate the PS_i and w_i values.

Rousseeuw and Van Zomeren in [71] showed through Monte Carlo simulations that when a collection of data points in n-dimensions are drawn from a multivariate Gaussian distribution, their squared projection statistics follow roughly a chi-squared distribution with n degrees of freedom. Since in our case observations are in 2 dimensions, a statistical test was applied at a significance level of say 97.5% to tag as an outlier any data point $\mathbf{d}_i$ that has $PS_i^2 > \chi^2_{2,0.975}$. This allows us to define a weight function w_i as $w_i = \min(1, \tau/PS_i^2)$, where $\tau = \chi^2_{2,0.975}$, which is used in the objective function of the GM-estimator given by Equation 5.3. Note that this weight function decreases as the squared PS gets larger than threshold τ.

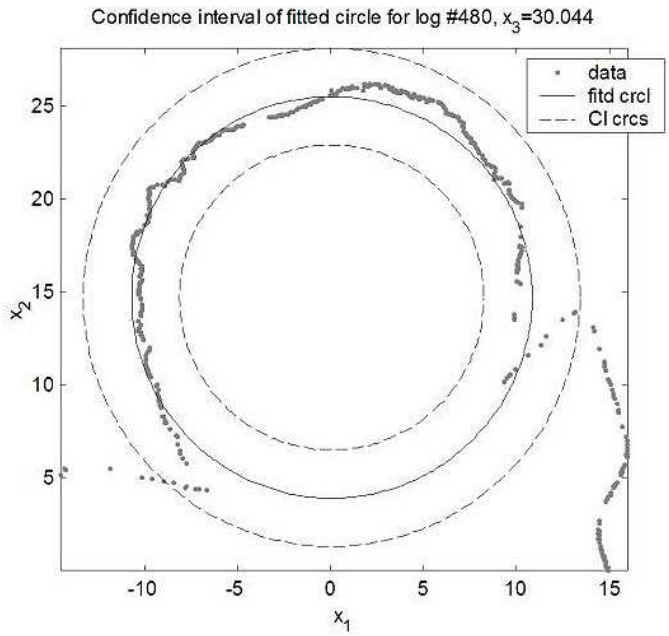


Figure 5.1: *Most outliers are excluded from the confidence ring.*

5.3.3 Determining Confidence Rings of the Fitted Model

The extreme data points in the log data can be detected by determining the confidence ring of a fitted circle. Such points are composed of outliers, as well as data that are part of a log defect with significant protrusion or depression. The 95% confidence ring is the region between two circles both centered at (p_1, p_2), with radius $(p_3 - \Delta p_3)$ and $(p_3 + \Delta p_3)$, respectively, where $\Delta p_3 = 2 \times 1.428 \times \text{median}_i|d_i|$. If a data point is outside that confidence ring, it may belong either to a loose bark or to a defect with large protrusion or depression. Figure 5.1 demonstrates such a method.

5.4 Deriving the Influence Function of GM-Estimator

Following Neugebauer and Mili [54, 55], we derive the asymptotic influence function of the GM-estimator and show that it is bounded. To this end, consider a set of 2-dimensional

measurements of size m, $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_{m-1}, \mathbf{x}$ whose coordinates are contained in vector $\mathbf{x}^*$. Suppose that the first $m-1$ measurements, whose coordinates are contained in vector $\mathbf{z}$, are independent and identically distributed (i.i.d.) according to the Gaussian cumulative probability distribution function, $\Phi(\mathbf{z})$, while the last measurement point, $\mathbf{x}$, takes on arbitrary values on R^2, yielding a fraction of contamination, $\epsilon = 1/m$. Also, suppose that the vector $\mathbf{z}$ is independent from the m-dimensional model error vector, $\mathbf{e} = [e_1, e_2, \ldots, e_m]^T$, whose components are assumed to be i.i.d. according to a cumulative probability distribution function $K(e)$. Let $F(\mathbf{z}, \mathbf{e}) = \Phi(\mathbf{z})K(\mathbf{e})$ denote the joint probability distribution function of $\mathbf{z}$ and $\mathbf{e}$.

By processing the measurement vector $\mathbf{x}^*$, the GM-estimator, $\hat{\mathbf{p}}$, provides an estimate for $\mathbf{p}$ by seeking a solution to an implicit equation given by

$$\sum_{i=1}^{m} \boldsymbol{\lambda}_i(\mathbf{x}^*, \mathbf{p}) = \mathbf{0}, \tag{5.15}$$

where

$$\boldsymbol{\lambda}_i(\mathbf{x}^*, \mathbf{p}) = w_i \, \frac{\partial h_i(\mathbf{x}^*, \mathbf{p})}{\partial \mathbf{p}} \, \psi(\frac{r_i}{s \, w_i}). \tag{5.16}$$

Now, let m grow to infinity, leading to an infinitesimal fraction of contamination as $\epsilon \to 0$. Therefore, the cumulative probability distribution function of the random vectors $\mathbf{z}$, $\mathbf{x}$ and $\mathbf{e}$ may be expressed as the contamination model given by

$$G(\mathbf{x}^*, \mathbf{e}) = (1 - \epsilon)F(\mathbf{z}, \mathbf{e}) + \epsilon \Delta_{\mathbf{x}}, \tag{5.17}$$

where $\Delta_{\mathbf{x}}$ is the unit probability mass at point $\mathbf{x}$. Letting $\mathbf{T}(G)$ denote the asymptotic functional form of $\hat{\mathbf{p}}$, Equation 5.15 reduces to

$$\int \boldsymbol{\lambda}(\mathbf{x}^*, \mathbf{T}(G)) \, dG(\mathbf{x}^*, r) = \mathbf{0}. \tag{5.18}$$

The asymptotic influence function of the estimator T(G) at F is defined as the Gâteaux derivative given by

$$\mathbf{IF}(\mathbf{x}; F) = \frac{\partial \mathbf{T}(G)}{\partial \epsilon}|_{\epsilon=0} = \lim_{\epsilon \downarrow 0} \frac{\mathbf{T}((1-\epsilon)F + \epsilon\Delta_{\mathbf{x}}) - \mathbf{T}(F)}{\epsilon}. \tag{5.19}$$

It is the directional derivative of $T(G)$ in the direction of $\Delta_{\mathbf{x}}$ at F. To derive it, let us first substitute Equation 5.17 into Equation 5.18 to get

$$\int \boldsymbol{\lambda}(\mathbf{x}^*, \mathbf{T}(G))\, dF + \epsilon \int \boldsymbol{\lambda}(\mathbf{x}^*, \mathbf{T}(G))\, d(\Delta_{\mathbf{x}} - F) = \mathbf{0}. \tag{5.20}$$

Differentiating with respect to ϵ, it follows

$$\frac{\partial}{\partial \epsilon} \int \boldsymbol{\lambda}(\mathbf{x}^*, \mathbf{T}(G)) dF + \int \boldsymbol{\lambda}(\mathbf{x}^*, \mathbf{T}(G)) d(\Delta_{\mathbf{x}} - F) + \epsilon \frac{\partial}{\partial \epsilon} [\int \boldsymbol{\lambda}(\mathbf{x}^*, \mathbf{T}(G)) d(\Delta_{\mathbf{x}} - F)] = \mathbf{0}. \tag{5.21}$$

The Huber function $\psi(\frac{r}{s\,w})$ is continuous and measurable on F, and $\psi'(\frac{r}{s\,w})$ is measurable on F. Thus, by Equation 5.16, we know for our case, $\boldsymbol{\lambda}(\mathbf{x}^*, \mathbf{T}(G))$ is continuous and measurable on F, and its derivative measurable on F. Evaluating Equation 5.21 at $\epsilon = 0$, assuming Fisher consistency given by $\int \boldsymbol{\lambda}(\mathbf{z}, \mathbf{T}(F))\, dF = \mathbf{0}$, $\boldsymbol{\lambda}(\mathbf{x}^*, \mathbf{T}(G))$ satisfies regularity conditions [34], and interchanging differentiation and integration in the first term of the summation, then

$$\int \frac{\partial}{\partial \epsilon} \boldsymbol{\lambda}(\mathbf{x}^*, \mathbf{T}(G))|_{\epsilon=0}\, dF + \int \boldsymbol{\lambda}(\mathbf{x}^*, \mathbf{T}(F))\, d\Delta_{\mathbf{x}} = \mathbf{0}. \tag{5.22}$$

On page 301 of [34], Theorem 7.10.1 states the regularity conditions are: (1) Function $f(a_1, a_2)$, where a_1 and a_2 are independent variables, has the property that $\frac{\partial}{\partial a_2} \int f(a_1, a_2)\, da_1$ exists. (2) Further, function $f(a_1, a_2)$ should be continuous, and has a continuous first-order partial derivative with respect to a_2. Applying the chain rule to the kernel of the first integral

and using the sifting property of the Dirac impulse, we obtain

$$\int \frac{\partial}{\partial \mathbf{p}} \boldsymbol{\lambda}(\mathbf{z}, \mathbf{p})|_{\mathbf{T}(F)} \frac{\partial \mathbf{T}(G)}{\partial \epsilon}|_{\epsilon=0} \, dF + \boldsymbol{\lambda}(\mathbf{x}, \mathbf{T}(F)) = \mathbf{0}. \tag{5.23}$$

Solving for $\mathbf{IF}(\mathbf{x}; F) = \frac{\partial \mathbf{T}(G)}{\partial \epsilon}|_{\epsilon=0}$, then

$$\mathbf{IF}(\mathbf{x}; F) = -\left(\int \frac{\partial}{\partial \mathbf{p}} \boldsymbol{\lambda}(\mathbf{z}, \mathbf{p})|_{\mathbf{T}(F)} dF \right)^{-1} \boldsymbol{\lambda}(\mathbf{x}, \mathbf{T}(F)). \tag{5.24}$$

Deriving $\lambda(.)$ given by Equation 5.16 with respect to $\mathbf{p}$ while assuming that w and s are independent of $\mathbf{p}$ over the neighborhood where the derivative is applied, it follows

$$\frac{\partial \boldsymbol{\lambda}(\mathbf{z}, \mathbf{p})}{\partial \mathbf{p}} = w \left[\frac{\partial \psi(\frac{r}{s\,w})}{\partial \mathbf{p}} \right] \left[\frac{\partial h(\mathbf{z}, \mathbf{p})}{\partial \mathbf{p}} \right]^T + w \, \psi(\frac{r}{s\,w}) \, \frac{\partial^2 h(\mathbf{z}, \mathbf{p})}{\partial \mathbf{p}^2}$$

Applying the chain rule to the derivative of $\psi(.)$ with respect to $\mathbf{p}$ and using the fact that $\partial r / \partial \mathbf{p} = -\partial \mathbf{h}(\mathbf{z}, \mathbf{p}) / \partial \mathbf{p}$, the following equation is obtained

$$\frac{\partial \boldsymbol{\lambda}(\mathbf{z}, \mathbf{p})}{\partial \mathbf{p}} = -\frac{1}{s} \psi'(\frac{r}{s\,w}) \left[\frac{\partial h(\mathbf{z}, \mathbf{p})}{\partial \mathbf{p}} \right] \left[\frac{\partial h(\mathbf{z}, \mathbf{p})}{\partial \mathbf{p}} \right]^T + w \, \psi(\frac{r}{s\,w}) \, \frac{\partial^2 h(\mathbf{z}, \mathbf{p})}{\partial \mathbf{p}^2} \tag{5.25}$$

where $\psi'(u) = d\psi(u)/d\,u$. Substituting Equations 5.16 and 5.25 into the expression of $IF(.)$ given by Equation 5.24 to get

$$\mathbf{IF}(\mathbf{x}; F) = w \, \psi(\frac{r}{s\,w}) \, \frac{\partial h(\mathbf{x}, \mathbf{p})}{\partial \mathbf{p}} / A, \tag{5.26}$$

where

$$A = \int \left\{ \frac{1}{s} \psi'(\frac{r}{s\,w}) \left[\frac{\partial h(\mathbf{z}, \mathbf{p})}{\partial \mathbf{p}} \right] \left[\frac{\partial h(\mathbf{z}, \mathbf{p})}{\partial \mathbf{p}} \right]^T - w \, \psi(\frac{r}{s\,w}) \, \frac{\partial^2 h(\mathbf{z}, \mathbf{p})}{\partial \mathbf{p}^2} \right\} |_{\mathbf{T}(F)} \, dF.$$

It was observed that the influence function, $IF(.)$, is bounded because $\psi(.)$ is bounded

and because the weight function w is decreasing from one to zero for an outlier, $\mathbf{x}$, and thereby is bounding the influence of the column vector, $\partial h(\mathbf{x}, \mathbf{p})/\partial \mathbf{p}$.

5.5 Algorithm for Projection Statistics

The algorithm for projection statistics consists of the following main steps:

1. For a certain j in $[1, n]$, let $med_{i=1,\ldots,m}(d_{ji})$ denote the median of $\{d_{j1}, d_{j2}, \ldots, d_{jm},\}$. Calculate the coordinate-wise median given by

$$\mathbf{M} = [med_{i=1,\ldots,m}(d_{1i}), med_{i=1,\ldots,m}(d_{2i}), \ldots, med_{i=1,\ldots,m}(d_{ni})]^T$$

2. Calculate the directions $\mathbf{u}_i = \mathbf{d}_i - \mathbf{M}, i = 1, \ldots, m$. Whenever $\mathbf{d}_i == \mathbf{M}$, yielding $\mathbf{u}_i == \mathbf{0}$, disregard the corresponding direction in subsequent computation.

3. Calculate $\mathbf{v}_i = \mathbf{u}_i / \|\mathbf{u}_i\| = \mathbf{u}_i / \sqrt{u_{1i}^2 + u_{2i}^2 + \cdots + u_{ni}^2}, i = 1, \ldots, m$

4. Calculate the standardized projections of $\{\mathbf{d}_1, \mathbf{d}_2, \ldots, \mathbf{d}_m\}$ on $\mathbf{v}_k, k = 1, \ldots, m$, which are given by

$$z_{1k} = \mathbf{d}_1^T \mathbf{v}_k; z_{2k} = \mathbf{d}_2^T \mathbf{v}_k; \ldots; z_{mk} = \mathbf{d}_m^T \mathbf{v}_k; k = 1, \ldots, m$$

5. Calculate $med(z_{1k}, \ldots, z_{mk}) = z_{med,k}, k = 1, \ldots, m$.

6. For a certain k in $[1, m]$, let $med_{j=1,\ldots,m}|z_{jk} - z_{med,k}|$ denote the median absolute deviation of $\{z_{1k}, z_{2k}, \ldots, z_{mk}\}$ from $z_{med,k}$. Then for every k in $[1, m]$, calculate the k-th median-absolute-deviation from the median, that is

$$MAD_k = 1.4826\, med_{j=1,\ldots,m}|z_{jk} - z_{med,k}|, k = 1, \ldots, m$$

7. For all i in $[1, m]$, and all k in $[1, m]$, calculate the standardized projections:

$$P_{ik} = |z_{ik} - z_{med,k}|/MAD_k,\ i, k = 1, \ldots, m$$

8. For all i in $[1, m]$, calculate the projection statistics:

$$PS_i = \max\{P_{i1}, P_{i2}, \ldots, P_{im}\},\ i = 1, \ldots, m$$

5.6 Simulation Results

Simulations for the developed robust estimators were performed using several complete log samples, some were executed on single data cross sections, while the rest on the entire log data. First, we discuss the results obtained using data cross sections. Then, radial distances are analyzed, and contour curves generated for defect identification. The circle fitting procedure was implemented in Java programming language [80]. The version we use is Sun Java virtual machine 1.5. The maximum number of iterations needed for estimating the circle parameters per cross section is limited to 5. In most cases no more than 4 iterations are required. One log data sample has about 80 to 100 cross sections. It takes less than 1 minute to complete all the circle fitting to cross sections on a HP notebook with a 3.06 GHz Intel Pentium 4 processor with hyperthread. The reason we choose a personal computer for the simulation is that ultimately the system will be used by sawmills where high-end personal computers are more affordable than high-performance work stations or other more powerful computers. The programs have not been tested on any laser scanning equipment. They were only executed on the HP notebook personal computer.

Table 5.1: *Statistics of Some Log Data*

i	x_{i1}	x_{i2}	d_{i1}	d_{i2}	PS
1	-14.29	5.49	-4.74	-3.03	**12.44**
2	-9.16	8.93	0.47	0.29	0.77
3	-14.25	5.45	-4.72	-3.04	**12.39**
4	-9.09	8.75	0.43	0.28	0.73
5	-14.47	5.15	-4.99	-3.27	**13.06**
6	-9.01	8.31	0.28	0.19	0.65
7	-11.85	5.47	-2.88	-2.23	**7.88**
8	14.94	0.03	7.11	-6.80	**18.47**
9	14.76	0.55	6.83	-6.37	**17.67**
10	14.55	1.33	6.46	-5.77	**16.60**

5.6.1 Circle fitting using the GM-estimator

The simulations were carried out on the cross section of log data as shown in Figures 5.2 and 5.3. These are data points of log# 480 at length 30.044 inches, which is a cross section with 786 data points. Table 5.1 displays the projection statistics calculated from the projection statistics assessed from the radial distances, which are denoted by PS. The square root of the 97.5 percentile of the chi-squared distribution with 2 degrees of freedom, $\sqrt{\chi^2_{0.975}} = 2.7$, is the threshold chosen for PS beyond which a point is flagged as an outlier. It was observed that PS identifies all the outliers in the data.

Figures 5.2 and 5.3 further demonstrates the robustness of the GM algorithm. Here, in the presence of severe outliers, the GM solution is very satisfying.

The experiments with the circle fitting robust regression model brought insight to the research work of external log defect detection of hardwood logs and stems. First, it is essential to perform a model fitting to the log data, because the fitted solutions help to sort the input data and provide a reference level of the log surface for defect detection, segmentation, and classification. Depending on the skill of the operator and the log size, the four units may or may not be calibrated well. The parameters of the fitted model, that is, the center of a fitted circle, can be applied to remove redundant data, and to sort data

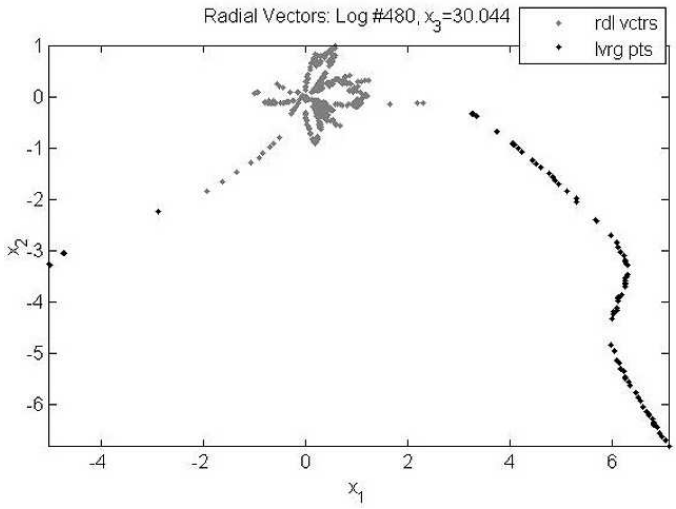


Figure 5.2: *End points for radial vectors (from the origin) of one data cross section. Outliers (leverage points) are marked in darker color, and are visibly separable from the good data.*

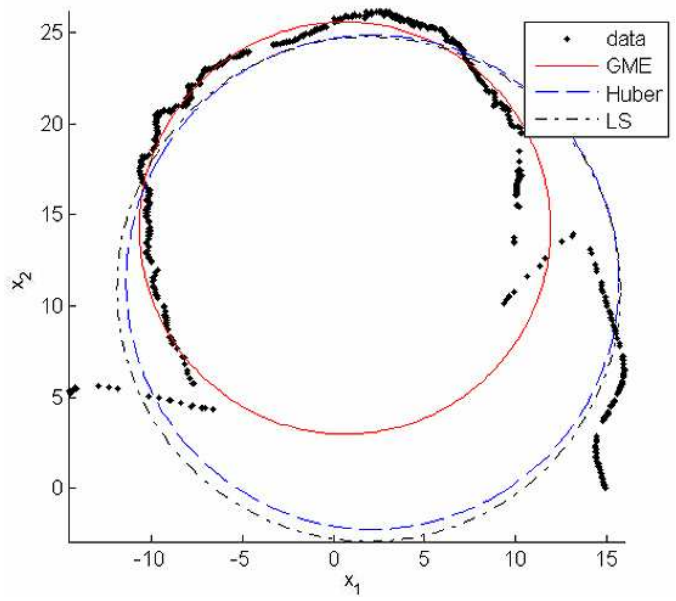


Figure 5.3: *A cross section of log data with a large segment of missing values along with outliers marked and three fitted circles superimposed. These circles have been fitted using the nonlinear robust GM-Estimator (solid red), the Huber M-estimator (dashed blue), and the least squares estimator (dashdot black). The robust GM-fitted circle passes through the good data points while the other two fitted circles are attracted by the loose bark data points, namely the outliers.*

points in an increasing order of angles of the vectors passing through the circle center and data points with respect to the horizontal axis.

Moreover, a robust 2-D circle fitting helps to amplify the variation on log surfaces that contain external defect information. The criterion considered for a good fitting algorithm is the one for which the solution minimizes the variance of the regular bark areas, and maximizes that of the defect regions. To do so, the weight function of the data, defined in Section 5.3, should give more weight to data in the bark area, and give less weight to data in the defect region. Statistically, it is assumed that a regression model, for instance, a circle, approximately fits a log data cross section. Typically, the bark region tends to fluctuate around the assumed model with small variations for the majority of the log data cross section, thereby revealing a large protrusion or depression that departs from the fitted model significantly.

5.6.2 The Radial Distance Images

Radial distances obtained from the log data are the signed values. To create a radial distance image, the radial distances are converted to gray-scale values as depicted in Figure 4.4. Typical radial distances range from -2 to 2 inches, and the gray-scale values, 0 to 255. Since the log data are not originally in a grid format, the corresponding radial distances are not in a grid format either. To form a grid, the radial distances are interpolated linearly to fill up any gaps. This is carried out as follows. First, the x_3 value (position along the log's length) in the 3-dimensional data is mapped to the row number, i in the 2-dimensional image, $i = 1, \ldots, m$. Secondly, the column number, j, is calculated by scaling the angle of a point from the center of the fitted circle, $j = 1, \ldots, n$. If the desired image is to be 750 pixels wide, then the scaling factor would be $750/(2\,\pi)$. Thus, the 3-dimensional point (x_1, x_2, x_3) with a radial distance of rd_{ij} would become the 2-dimensional point (i, j) with a gray-scale value of c_{ij}. The radial distances are linearly interpolated that each point (i, j) is associated with a certain value. After such a conversion, they are referred to as gridded. To convert

the gridded radial distance, rd_{ij}, to a gray value, c_{ij}, the maximum, rd_{max}, and minimum, rd_{min}, of all the radial distances are first determined an the c_{ij} is calculated through

Since the number of rows and columns are out of proportion ($10^2\, vs.\, 10^3$), another linear interpolation is performed to insert rows between the original radial distance rows. This creates a radial distance image resembling the log surface. This is illustrated in Figure 6.5, where residual images were generated by circle fitting along with the log defect diagram for comparison.

Chapter 6

Algorithm for External Defect Detection Using Radial Distances

6.1 Algorithm Overview and Pseudo Code

The external-defect detection procedure includes two major steps. The first step is to obtain the radial distances by fitting 2-D circles to log-data cross sections using the robust GM-Estimator described in Section 5.6.1 and in further detail in [91]. The program is written in Java. It outputs of a matrix of radial distances from the fitted circles to the actual log data (see Figure 4.4). The second step of the procedure is to determine the actual defects on the log surface. Current implementation for this phase is in Matlab 7. The detection program incorporates expertise that was obtained through measuring, photographing, and analyzing of approximately 500 external-defect samples.

Before describing the detection algorithm, let us first define the "defects" that the algorithm is expected to detect. The scanning technology limits the types of defects that can be found. Defects should be at least 5 inches in diameter, otherwise the defects are undetectable under the 0.8-inch resolution along the log length provided by the scanning system.

The current detection algorithm only detects defects with a minimum one inch surface rise, because the algorithm is height (surface rise) based. Thus, "large defects" means those with at least a one inch surface rise, five inches in diameter, and a width to length ratio between 0.5 and 2. In the 14 log data samples, 60 defects of this type were observed, and 59 were located. "Medium defects" mean those with a distinctive bark pattern, a medium rise (0.5 to 1 inch), and a medium diameter (3 to 5 inches). Eight such defects were observed in the log samples. There are 8 of this type, 4 of which were identified in 14 log sample data. Both "large defects" and "Medium defects" are severe. The defect detection algorithm was tested using these 68 defects.

As discussed in Chapter 3, the external defect characteristics provided a foundation for us in the development of the detection algorithm. The surface rise information suggests that most of the defects to be identified would be knobs ans sawn knots. Their surface rise are commonly 1 inch, with a few exception of 0.5 inch. The base width and length of knots are 5-6 inches, indicating contour curves will likely enclose a region with a 5 inch diameter. The median length of sawn knots is 9.5 inches, which is misleading. Due to the nature of these defects, often only a small corner of the defects are enclosed in the contour. Thus, we need another method to determine the entire surface of a sawn knot, which is referred to as sawn top in the remaining discussion of this chapter. From Table 3.2, we observe that is a quarter of the knobs and sawn knobs are about 1 inch high (tip), 5 inches wide, and 5.5 inches long at the base. This indicates a small group of knobs might be identified as no higher than 1 inch, and no wider than 5 inches. This is indeed the basis for the "medium defects" detected by the algorithm in Section 6.3.

In the remainder of this document, the following terminologies are used:

- A *contour*, or contour curve in a plot, is a curved line connecting points with the same surface rise;
- A *rectangular region* (typically referred to simply as a region) is a solid rectangle enclosed by the bounding box for a contour.

Here is a pseudo code overview of the defect detection algorithm:

1. **Find large defects**

   (a) Using radial distance data, obtain contours at a set of evenly spaced levels. The first level is the lowest; the highest level is usually greater than 1.5 inches. From this point, most processing is on the bounding boxes (regions). See Section 6.2.1 for detailed explanation.

   (b) Eliminate regions whose area is less than a certain threshold; sort the remaining regions in descending order of area. See Section 6.2.2.

   (c) Eliminate various other regions that are unlikely defective. See Section 6.2.3.

   (d) Adjust bounding boxes that do not enclose entire sawn tops. The adjusted bounding boxes are referred to as adjusted regions. Remove adjusted regions with severe missing data, and remove adjusted regions that are too small. See Section 6.2.4.

   (e) The remaining regions are reported as possible defects.

2. **Find the less protruding and smaller diameter defects no more than 1 inch in height and 3 to 5 inches in diameter**.

   (a) Using the original 3-D log data, determine gradients parallel to the long axis of the log. See Section 6.3.1.

   (b) Find regions whose gradients are within a defined range for this defect class. See Section 6.3.2.

   (c) These areas are reported as defects.

6.2 Algorithm for Detecting Large Defects

6.2.1 Generate Contours

A Matlab built-in function inputs and analyzes radial-distance data to generate contour curves. The curves are then analyzed to locate where surface defects might exist. Recall that radial distances are generated by the circle-fitting procedure of Section 5.6.1. A gray-scale image is only a graphical way to illustrate them. Now for each contour curve, the algorithm determines its borders. The width, length, area, width/length ratio, and length/width ratio are then computed. Because the radial distances are generally less than 5 inches, it was found that partitioning the contours into six levels proved effective for the algorithm to determine the defects. Presently, only the highest level contours were analyzed, as they enclose the highest rising regions and thus the most protruding defects. Usually each log sample has anywhere from a few dozen to a few hundred contour curves at the highest level.

The original 3-D log data are then read in. Depending on the scanner calibration and the diameter of the log, the original log data may contain a certain amount of identical points. Therefore the algorithm removes duplicates. For each data point, a line is drawn from it to the cross-section's fitted-circle origin. The angle between this line and a horizontal line is computed. The points on a cross-section are then sorted by their angle values.

The main idea throughout the remainder of the algorithm is to take a series of steps to eliminate non-defective regions from the potential candidates. This is achieved by using statistics from measured and calculated log data, and wood-science expert knowledge in a stepwise fashion.

6.2.2 Elimination of Non-defective Regions

Remove Small Regions First, the algorithm removes regions whose area is less than 7.5 inch^2, because the data resolution (0.8 inch between cross sections) means they cannot

be reliably recognized as defects. Most defects are associated with a 0.75 width/length ratio. Thus, regions with an area less than 7.5 inch^2 would have less than 3 cross sections intersecting it, and so could not be detected. Next, the remaining regions are sorted in order of their areas. This makes it easy to determine whether a smaller region is nested inside a bigger one. Any contour nested within another is removed from consideration because there can only be one defect in the same location.

Small regions (with area less than 10 inch^2) that are within two inches of the top or bottom of the image are rejected as well. They either enclose partial defects (part of the defect is lost), which the algorithm is incapable of detecting, or a small defect that cannot be detected due to current data resolution. Since this is an artifact of the original scanning process, defects near or outside the scanned region were not identified for the purpose of testing the algorithm.

Region Adjustment At the beginning of the algorithm, to get a rough estimation of potential defect locations, only the widths and lengths of contour bounding boxes are used. However, this is not accurate enough to determine the true extent of certain defects. To make sure the entire region of an external defect is identified, the algorithm adjusts the width, length, and width-length ratio of the region. It is done as follows. First, for each selected candidate rectangle, an extended region surrounding the curve is analyzed. The top and bottom boundaries of the enclosing rectangle are expanded each by a length of 10 cross-sections (8 inches) along the log length. The "widest consecutive segment" of each cross section refers to a set of continuous data points with radial distances greater than the contour level. A segment is a set of lines connecting the adjacent log-data points in the same cross section and enclosed in the contour curve. This step provides us with precise shape information about the potential surface-defect regions, shown in Figure 6.1.

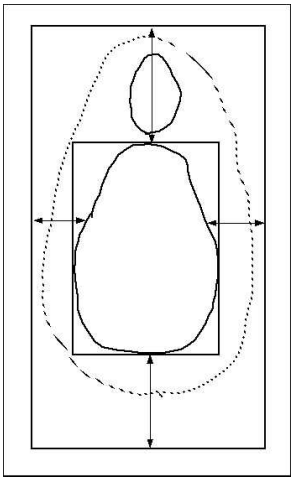


Figure 6.1: *An illustration showing that by expanding the bounding box, it may help to determine the true defect region, rather than a portion of it. The dashed curve encloses the base of a defect.*

6.2.3 Deletion of Non-Relevant Regions

Bark Regions Bark regions are not considered defective. They have (1) an area larger than 25 inch^2, (2) at least 75% of the segments inside the contour are associated with the following characteristic: the ratio between the widest consecutive part of each segment, and the total width of the region is less than 0.8. Regions with these features are unlikely to be defective, and so are rejected from further consideration.

Remove Fragments For the remaining regions, segments that are wide enough (width of the widest consecutive segment greater than 1/4 of the bounding rectangle width) are identified. The algorithm then determines whether the top or bottom of an enclosed region is a narrow and long fragment along log length with a width less than 1/4 that of the rectangle, indicating bark, instead of being part of an actual defect. If such a fragment exists, such as the part being marked by a cross in Figure 6.2, the top or bottom boundary for the region is adjusted to remove the bark artifact. Thus, some regions might be rejected as being long and narrow, and thus non-defective.

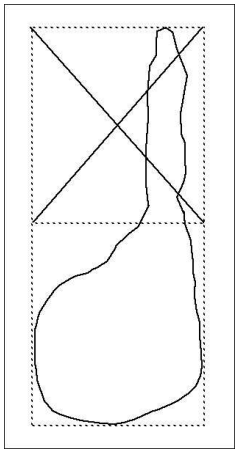


Figure 6.2: *A contour encompassing a defect with a long and narrow portion that is not part of it, which the algorithm identifies and removes.*

Remove Regions too Close to Large Regions Regions that are smaller than 50 inch^2 and are too close to larger candidates (less than 3.5 inches apart horizontally or vertically) are excluded. Due to the nature of defect distribution on hardwood log surfaces, the larger ones more likely indicate the true defects, while the smaller ones are simply continuations of the same defect. This is how it is done: Among candidates with a length less than 7 inches, or longer than 7 inches and width/length ratio greater than .2, those less than 50 inch^2, and less than 3.5 inches apart from the selected larger ones, are excluded.

Remove Regions with Non-Defective Shapes or with Missing Data When the area is less than 15 inch^2 and the width/length ratio is out of range (less than 0.5 or greater than 2), they are also removed as they are too small and are not shaped like a defect. Candidates are then checked for amount of missing data. If there are more than 20 points missing in a segment (i.e., the data cross section has a gap wider than 1 inch), it is classified as a corrupted segment. If there are more than 50% corrupted segments enclosed in the contour, the region is classified as severely missing data and is rejected. Figure 6.3 illustrates such a situation.

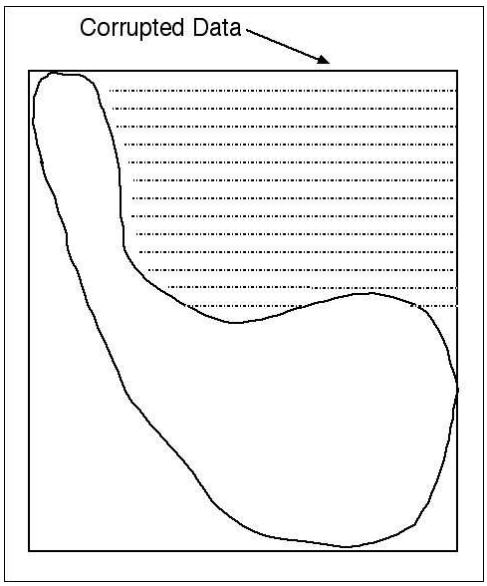


Figure 6.3: *A region with a large portion containing corrupted data and therefore is rejected as possibly defective.*

6.2.4 Determine Sawn Tops

A sawn top is a type of external defect where the tree limb was removed by loggers in the woods. Often it is not completely leveled with respect to the log surface, but instead tilted at a small angle. Since it's a natural human operation, the sawn top is often not completely flat. Sawing on natural wood material leaves a sawn pattern. Typically, part of the sawn top will fall below the highest contour level, and this section of the defect needs to be recognized. The algorithm is able to locate such regions using a "straight-line" segment technique described below, and is capable of adjusting the boundaries to identify the entire flattop region.

The procedure to find sawn tops is as follows: For remaining regions with an area less than 25 inch^2, in its surrounding region the algorithm examines angle changes between straight lines connecting log data points at an interval of five points along the cross sections. If changes are small enough (less than 25^o), these segments are recorded as relatively straight. Then the range of "straight" segments is determined. If over half of the segments contain straight parts, this region is identified as a sawn top, either sound (not rotten), or unsound

(rotten). The boundary of the identified region is adjusted to surround all "straight" segments, so as to capture that portion of the sawn top that falls below the contour level.

Some regions may be falsely identified as a sawn top, because they contain severe missing data causing the algorithm to generate an incorrect result. Thus, they are rejected depending on how severe the missing data are. Since the process of identifying sawn tops is often accompanied by adjustment of the defect region boundaries, which affects the geometric relationships among the detected regions, regions completely nested or partially overlapped are identified and removed. To this point, those candidates that have survived are considered to be large defects. Their rectangular borders are plotted on the contour image, and are labeled with their rank number in decreasing order of region areas.

6.3 Finding Medium Defects

So far, the algorithm has attempted to locate the most obvious defect types (Part 1 of the pseudo-code description). They are large bump-like knots, either old (healed broken stubs) or new (sawn at harvest). They may be large (20 inches diameter) or relatively small (4 inches diameter), protruding (at least 3 inches high) or with a more gentle rise. They can also be unsound or sound. There is another group of severe defects, with medium rise (0.5 to 1 inch), and medium diameter (3 to 5 inches). Due to these characteristics, they are not enclosed in the highest contour curves and thus not identified by the procedure described so far. However, they have a distinctive pattern (surface rise and diameter). Thus, an algorithm explicitly designed to identify these defects was developed, for what we referred to as medium defects. In a sample of 14 logs, eight such defects were observed and the algorithm was able to detect 4 of them.

Initially, the original log data points are processed by removing outliers outside of the 99th percentile, which is roughly 2 inches in radial distance. Then the data points are sorted according to the angles of vectors passing through the circle center and points. The approach

applied here requires that there be no missing data. Thus the algorithm "fixes" regions with missing data in the matrix of radial distances by using a linear interpolation.

6.3.1 Determining Gradients

The next step is to determine the existence of upward slopes and downward slopes that fall within 0.15 to 0.3. Such a range indicates that a protruding region is high enough, but not so high as to represent a protruding defect that should have been detected in the first stage. A slope here refers to a group of adjacent data points, whose radial distances increase or decrease along the log length in a general trend, similar to a slope in a mountain. During the process, a group of adjacent data points along the log length (x_3-axis) are examined. In this procedure, the type of defects are not large or protruding—those defects should have been detected earlier. If the gradient falls within a certain range, it is tagged. Also note that the predominant surface feature of a log is bark, which has an uneven texture. Therefore the data points on a slope usually do not form a strict straight line. The algorithm detects such slopes by judging their tendency, either going up or down, and an appropriate tolerance threshold—no more than 1 slope is out of the range—is applied.

6.3.2 Finding Defective Regions

Based on the results from slope detection, those regions satisfying the following conditions are determined.

1. width and length of 3 to 5 inches;
2. height of 0.5 to 1 inch, and
3. no more than 1 slope is out of the range.

This kind of defect can also include rotten and non-rotten, sawn, or naturally formed defects. The detected medium defects are plotted in the same contour image with the large defects previously identified. This completes the algorithm.

6.4 Simulation Results and Discussions

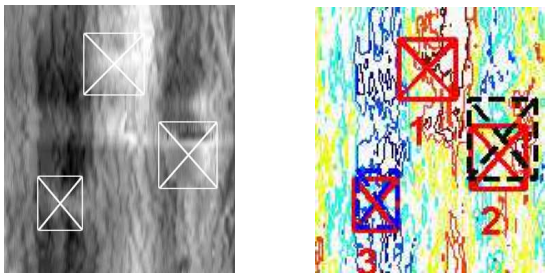
Fourteen log data samples were chosen based on their data characteristics, and analyzed using the defect detection system. The algorithm was written in Matlab using Matlab version 7.0. As mentioned in Section 5.6, it is implemented on a high-end notebook computer with a Pentium 4 processor. It takes less than 1 minute to finish the calculation of contour curves, defect detection, and results output. The programs have not been tested on any laser scanning equipment. They were only executed on the HP notebook personal computer. The defect diagrams of all external defects present on log samples were collected manually by the USDA Forest Service lab in Princeton, WV. Since logs are heavy (1,000 to 5,000 pounds), and come in various taper, sweep, and diameters at the two ends, accurately measuring the defect locations and sizes, and classifying defect types, proved challenging. Consequently the diagrams are often erroneous, ambiguous, and inaccurate. Further, they often only contain the width and length of a defect, but not its height, or surface rise. External defects may not always be visible in the color images of a sample log, and the angle order of each side of the color images are often incorrectly arranged. Among the 160 or so scanned log data samples, 45 of them are poor quality and not usable. These problems cut down the number of log samples that could be experimented with.

The defect diagrams illustrate not only the defects visible in the radial-distance gray images, but also those undetectable due to the adopted methods and/or the data resolution limits. The information from the diagrams, as well as from the color images were combined (Figure 6.4). Observed defects were marked in gray-scale images (Figure 6.5(a)). We will refer to them as "ground truth". The coordinates of the marked rectangles are measured and

Figure 6.4: *Four digital intensity image of a log sample at* 90^o *per side. These images are used in part to determine the correctness of the machine generated defective regions.*

recorded. We can overlay them on the contour plot (Figure 6.5(b)) so as to compare them to the regions detected by the algorithm. In the contour plot, the predicted (observed) defect regions are marked in solid crossed rectangles, while the automatically detected regions are displayed with dashed crossed rectangles. The locations, widths, and lengths of automatically detected regions are reported by the programs. To determine whether a marked region in the contour plot correctly indicates an external defect, it is compared with the ground truth.

Table 6.1 gives a breakdown for each log sample of observed defect numbers, automated defect numbers, falsely identified defect numbers, and missed defect numbers. Table 6.2 gives a breakdown for each log sample of the surface area, automated defect area, false



(a) (b)

Figure 6.5: *The algorithm finds two of three defects, where a correct identification is defined as the center of a detected region falling inside the observed one, and vice versa. (a) The corresponding gray-scale image with manually marked defect regions. (b) A contour plot automatically generated by the defect detection Matlab programs. Dashed, crossed rectangles mark the possible defective regions, and solid and crossed rectangles are overlaid observed defective regions.*

Table 6.1: *Observed defect numbers, automated defect numbers, falsely identified defect numbers, and missed defect numbers for each log sample.*

Log #	Species	Total	Correct	False	Missed
444	BOAK	4	4		
448	ROAK	9	8	1	1
450	ROAK	4	3		1
453	ROAK	7	6		1
468	ROAK	3	3	1	
480	ROAK	6	6		
493	ROAK	6	5		1
501	ROAK	3	3		
508	ROAK	5	5	1	
521	ROAK	6	6	1	
537	ROAK	5	4	2	1
441	YPOP	2	2		
485	YPOP	6	6	2	
520	YPOP	2	2	2	
Total		68	63	10	5

Table 6.2: *The surface area, automated defect area, false identification area, and missed defect area for each log sample.*

Log #	Species	Surface	Observed	Automated	False	Missed
444	BOAK	5797	456	456		
448	ROAK	7284	1196	1105	30	91
450	ROAK	7278	570	553		17
453	ROAK	6301	1732	1671		61
468	ROAK	5453	959	959	122	
480	ROAK	7486	1256	1256		
493	ROAK	8551	364	314		50
501	ROAK	3916	445	445		
508	ROAK	4031	573	573	243	
521	ROAK	8560	496	496	113	
537	ROAK	6414	390	356	178	34
441	YPOP	4645	297	297		
485	YPOP	9352	1385	1385	309	
520	YPOP	6188	358	358	218	
Total		91257	10476	10223	1213	253

Table 6.3: *Observed defect numbers, automated defect numbers, falsely identified defect numbers, and missed defect numbers for each tree specie.*

Specie	Total	Correct	False	Missed
BOAK	4	4	0	0
ROAK	54	49	6	5
YPOP	10	10	4	0
Total	68	63	10	5

Table 6.4: *The surface area, automated defect area, false identification area, and missed defect area for each tree specie.*

Specie	Surface	Observed	Automated	False	Missed
BOAK	5797	456	456	0	0
ROAK	65275	7980	7728	687	253
YPOP	20185	2040	2039	526	0
Total	91257	10476	10223	1213	253

identification area, and missed defect area, all in inch^2. In both tables BOAK, ROAK, and YPOP refer to Black Oak, Red Oak, and Yellow Poplar, respectively. The sample numbers of each specie in turn are 1, 10, and 3. The majority of the samples are Red Oak. There is no false identification, or missed defects for the Black Oak sample, and no missed defects for all Yellow Poplar. However sample numbers of these two species are low. For each tree specie, Table 6.3 summarizes the observed defect numbers, automated defect numbers, falsely identified defect numbers, and missed defect numbers. Similarly, Table 6.4 summarizes for each tree specie the surface area, automated defect area, false identification area, and missed defect area, all in inch^2. From Table 6.1 and Table 6.2, we found that the average size of a correctly detected defect is 162 inch^2, but the average size of a missed defect is 51 inch^2. This tells us that the missed defects tend to be relatively small. In forest product industry, a large defect is worse than a small one. This shows that the detection algorithm is effective.

We used two methods to evaluate the performance of the detection algorithm. The first one, referred to as the "raw-count method", counts the number of defects detected out of the total number detected by hand to exist. In our experiments there are a total of 68 severe defects, of which 63 were correctly identified. There are 10 non-defective regions falsely identified as defects. Most non-identified defects are small (less than 5 inches in diameter) and/or relatively flat (less than 1 inch in surface rise). Nine of ten falsely identified regions contain high-rise bark regions that are enclosed in the highest contour curves. Their widths and lengths range from 6 to over 20 inches. The algorithm fails to remove them from the true defects using the criteria described in the previous section.

The other way to evaluate the algorithm performance is to calculate the surface area of detected defects against that of the ground truth. This is similar to the analysis proposed by Kline et al. [36] to evaluate the detection algorithm. It is consistent with statistical hypothesis testing [47]. The total surface areas are given as follows:

- log samples (LSA), 91,257 inch^2;
- observed external defects (ODA), 10,476 inch^2;

- automatically identified defects that match the observations (MDA), 10,223 inch^2;
- automatically identified defects that do not match the observations (FPA), an false identification, 1,213 inch^2;
- all defects determined by the detection algorithm (ADA), 11,435 inch^2;
- observed defects that are NOT identified by the detection algorithm—unidentified defects (FNA), 253 inch^2.

When the center point of a detected region falls inside the bounding box of an observed defect, and vice versa, it is said to be a correct identification, and the defect area given by the ground truth is used in calculation. If we use the defect area given by the automated detection, one may argue that the detection system could intentionally set it larger or smaller than the true value, which makes its objectiveness doubtful. Thus, we use the defect area given by the third party. Now the detection statistics are given as: the percentage of **observed clear region** is 88.5% ((LSA-ODA)/LSA×100%). The percentage of **automated clear region** is 87.5%, given by (LSA-ADA)/LSA×100%. That the latter is smaller than the former implies that the algorithm identified more defective surface area than the actual observed area. The percentage of **false positive** or the falsely identified defect regions from clear surface, is 1.5% (FPA/(LSA-ODA)×100%). The percentage of **false negative**, indicating how much the algorithm missed the defective regions, amounts to 2.4% (FNA/ODA×100%). Finally, 97.6% is the **area detection rate** for the defect detection algorithm with respect to observations, given by MDA/ODA×100%. Since the total of FNA and MDA is equivalent to ODA, the false negative rate and the detection rate add up to 1.

Thus, there are 2 sets of measures from the above two methods. By raw count, among the 63 observed defects there are 63 correct identifications, and 10 falsely identified regions. By area method, 97.6% observed defect area is detected, with 1.5% clear surface falsely declared as defective.

There are pros and cons with both evaluation methods. The pros for the raw-count method is that it is simple and easy to understand. However, in wood science and forest products, a large defect usually is much worse than a small one. Missing many small defects is unlikely as serious (economically) as missing a few large ones. Unfortunately the raw-count method cannot reflect this property. Another problem with the raw-count method is that statistically it is unclear what these numbers really mean. For instance, one should not compare the number of defects that are falsely identified (10) against those observed (68), as it is not the statistical property known as "false positive".

"False positive", or "Type I error", is the error of rejecting a null hypothesis when it is the true state of nature. In other words, this is the error of accepting an alternative hypothesis (the real hypothesis of interest) when an observation is due to chance [28, 47, 77, 97]. In medical science, for example, a false positive is a positive finding of a test when, in fact, the true result was negative. This would mean that the test results indicate that a patient had a particular condition or disease when they do not [15]. In the raw-count method, only the numbers of observed defects, correctly-detected defects, and falsely-detected defects are given. One may consider the detection rate is the ratio between the numbers of correctly-detected defects and observed defects. The false negative can be calculated as the ratio between the numbers of missed defects and observed defects. However, one cannot immediately infer from the above the detection accuracy in terms of false positive and false negative.

The area method overcomes problems occurred in the raw-count method. Both the detection rate and the false negative can be determined in a similar fashion as with the raw-count method. The false positive is not difficult to determine, because we have the total clear area, and it can be used in the calculation. Yet due to the relative inaccuracy of the calculation of defect area, the resulting numbers may not be completely reliable. This is because the area for each defect region is estimated as a rectangle, using the width and length of the matched ground-truth defect. The bounding rectangles give only a rough approximation to the "true" areas, both for the observed defects and the algorithm's reported defects. Thus, the calcula-

tions derived from those areas have relatively low precision. It is especially a problem when the statistic is only a few percent. For example, the false positive is 1.5 percent. With the lack of accuracy in our calculations, the error might be greater than that.

One may argue about the definitions for calculations in the area method as to their "reasonableness" or "fairness". Choices are made here on how these calculations were done. Would there be a methodological bias? Let us look into the facts of the calculations.

(a) Calculating the area of detected defects:
If there is a correct identification (defined as the center points for the two bounding boxes are each contained in the other), then the FULL area for the (observed) defect is credited to the algorithm, regardless of the area covered by the overlap portion of the two, and also regardless of the area covered by the algorithm's reported defect. Thus, this measure is only indirectly influenced by the size of the reported defects by the algorithm, in that smaller reported defects might reduce the chance of detecting the defect in the first place (because smaller area might reduce the probability of the center points overlapping).

(b) Calculating the area of undetected defects:
This is simply the area of the observed, undetected defects. That is, this measure is unrelated to the sizes of the defect regions reported by the algorithm.

(c) Calculating the area of false identifications:
This is directly the area of the mis-reported defects. Clearly, this is directly influenced by the sizes of these reported regions. Also clearly, an algorithm that consistently reports smaller regions gets a direct benefit in this metric.

The above statements hold true for an algorithm that consistently reports regions larger than the ground truth. The consequence would be reversed to those of (a)–(c), respectively. Thus, it is possible that these area measures are biased by the algorithm's reporting size for its detected defects. That is, an algorithm benefits in the metrics by consistently reporting

smaller or larger regions. There might be a penalty in measure (a) for smaller regions, but (b) is not influenced, and (c) clearly benefits. Note also that (a) is completely insensitive to the amount of overlap between the observed and reported rectangles, and is only sensitive to the binary decision of whether the two center points are included. In short, both the raw-count method and area method have their advantages and shortcomings. The raw-count method is simple, but does not reflect all detection performance aspects. The area method does report the complete set of numbers. However its accuracy is questionable. For our estimates of false positives and false negatives to be unbiased, the sample should have been randomly chosen. Also, the sample size should be sufficiently large so that these estimates of false positives and false negatives have not a too large variance. Obviously 14 logs are not a large number. It would be good to repeat these calculations over a much larger log sample, preferably randomly selected, to have a better evaluation of the performance of the detection method.

6.5 Testing of Parameter Values

In the algorithm description of Section 6.1, a large number of algorithm parameters are identified with specific constant values given. This naturally begs the question of why these parameter values are used. To determine whether various parameter settings used in the detection algorithm are appropriate, 10 of the most important parameters are tested. We tested the parameters individually, one value a time. From the algorithm description, we believe that it is a reasonable assumption that these parameters are independent of each other, and testing them individually can reasonably hope to improve the algorithm. Numbers of defects and falsely identified regions are small (68 and 10, respectively), therefore a change even by 1 is major. The 10 parameters are:

1. cut-off contour height
2. cut-off value for region area

3. cut-off value for bark area
4. rectangle horizontal padding for region adjusting
5. rectangle vertical padding for region adjusting
6. actual region width/length ratio
7. rectangle (region) length
8. width/length ratio during the search for a group of large defects
9. data point interval during the identification of flat tops
10. angle change during the identification of flat tops

For each parameter value testing, three sets of results were generated:

1. the number of correct identifications of each log sample given for each parameter value;
2. the number of false identifications of each log sample given for each parameter value;
3. the number of unidentified defects of each log sample given for each parameter value.

The results are shown in Tables 6.5 through 6.14. In these tables, original values of the detection algorithm and their corresponding results are shown in bold. Here we only present total number of defects, instead of total number of defect area, as the former effectively demonstrates changes along the change of parameter values.

In the original algorithm, we calculated the minimum and maximum radial distances, and determined the difference between them. Then this distance is partitioned at six even intervals. Only the topmost partition is used for determining the contours. Therefore, it could well be that adjusting this parameter up or down would yield contours that result in better detection. Therefore the range of height values between the fifth partition and the

Table 6.5: *Testing results for contour height.*

Level	1	2	3	4	5	6	7	8	9	**10**
Correct	33	38	40	39	37	44	49	49	54	**63**
False	49	36	37	31	24	25	21	24	15	**10**
Unidentified	35	30	28	29	31	24	19	19	14	**5**
Level	11	12	13	14	15	16	17	18	19	
Correct	56	49	47	50	44	40	34	28	14	
False	13	16	11	12	13	16	20	6	11	
Unidentified	12	19	21	18	24	28	34	40	54	

maximum radial distance is divided with 19 intervals. This is illustrated in Figure 6.6. Now index number 1 is assigned to the lowest of the 19 intervals, and 2 to the next interval, and so on. Index number 10 is the same as the 6th original interval, which is the original contour level used in the detection algorithm.

Since the increment among any two adjacent values of the original 6 intervals is only 1–2 inches, further partitioning with 19 intervals makes them about 0.1 inch apart. This is fine enough to capture possible effects given out scanner's resolution. Since each log sample has a unique set of radial distances, the interval distances are unique to each log. We add up the number of correct identifications for all log samples at each of the 19 levels (as labeled by an index number), even though for each log sample the real contour level for this index number is different from other log samples. The same holds true for the total numbers of false identifications, and those of unidentified defects. It can be observed from Table 6.6 that the original contour level yields the best results by all the measures. This is further illustrated in Figure 6.7.

In the remaining tables, we choose a different number of divisions for each parameter, as based on the characteristics of the parameter. In most cases the original parameter value is at the center of the testing range. The cut-off value of region area in Table 6.6 ranges from 5 to 15 inch^2. The original value is 7.5 inch^2. If we choose a value less than 5 inch^2, almost no fragments would be excluded. Similarly, we choose a value no more than 15 inch^2 because if

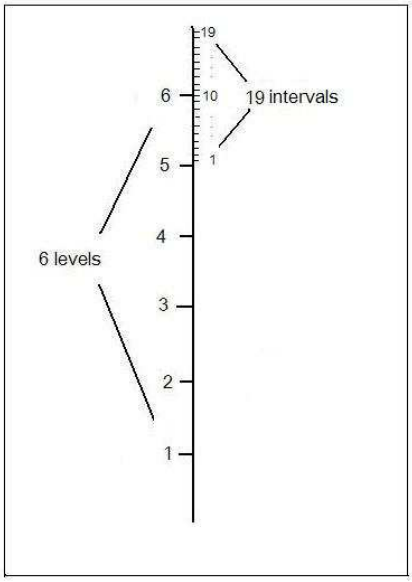


Figure 6.6: *Illustration of how radial distances are partitioned for the contour-level parameter testing.*

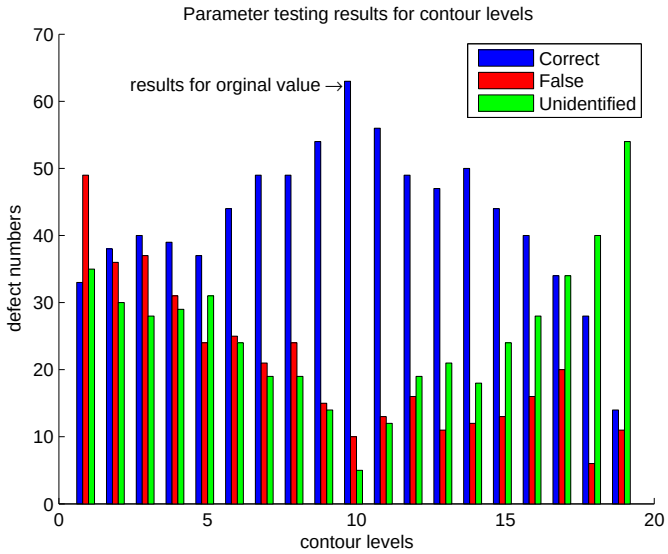


Figure 6.7: *Bar chart of parameter testing results for contour levels. Note the original contour level yields the best results by all measures.*

Table 6.6: *Parameter testing results for cut-off value of region area (in inch^2).*

Value	5.0	5.5	6.0	6.5	7.0	**7.5**	8.0	8.5	9.0	9.5	10.0	12.5	15.0
Correct	62	62	63	63	63	**63**	62	61	59	59	59	57	56
False	17	15	14	13	12	**10**	10	11	11	11	11	9	9
Unidentified	6	6	5	5	5	**5**	6	7	9	9	9	11	12

Table 6.7: *Parameter testing results for bark-region cut-off size (in inch^2).*

Value	20	**25**	30	35	40
Correct	61	**63**	62	62	62
False	10	**10**	11	10	10
Unidentified	7	**5**	6	6	6

the cut-off value is too large, many contours enclosing a defective region would be excluded. In Table 6.6, the increment is 0.5 for values between 5.0 and 10.0, and 2.5 for those between 10.0 and 15.0, because these intervals generates results fine enough for cut-off value of region area.

In Table 6.7, since bark regions that are non-defective tend to be large, the smallest value tested is 20 inch^2. However, any region larger than 40 inch^2 and is enclosed in a contour might be defective, therefore, we stop testing at this value. The interval is 5 inch^2 because the results vary only slightly.

Table 6.8 contains testing results for the parameter of rectangle horizontal padding. Since log surface data are unrolled, the horizontal value is proportional to angles between vectors

Table 6.8: *Parameter testing results for rectangle horizontal padding (left and right sides) during region adjusting (in degrees).*

Value	45	50	55	56	57	58	59	**60**	61	62	63	64	65	70	75
Correct	59	60	60	61	60	59	61	**63**	62	63	61	62	61	60	62
False	12	13	13	11	10	13	10	**10**	14	10	11	11	14	12	11
Unidentified	9	8	8	7	8	9	7	**5**	6	5	7	6	7	8	6

Table 6.9: *Parameter testing results for rectangle vertical padding (top and bottom) during region adjusting (in number of cross sections).*

Value	6	8	**10**	12	14
Correct	61	61	**63**	62	62
False	11	11	**10**	11	11
Unidentified	7	7	**5**	6	6

Table 6.10: *Parameter testing results for actual region width/length ratio.*

Value	0.3	0.4	**0.5**	0.6	0.7
Correct	63	63	**63**	61	61
False	11	10	**10**	10	10
Unidentified	5	5	**5**	7	7

of data points with respect to a horizontal axis. Thus this parameter is measured in degrees (^{o}). The values range from 45^o to 60^o. Too large or too small are not appropriate for the region adjustment. The interval is 5^o, which gives a good picture of how this parameter influences detection results.

Five different values were tested for the amount of rectangle vertical padding on the top and bottom of regions. They range from 6 to 14 cross sections, with an interval of 2 cross sections. Since the cross sections are approximately 0.8 inch apart, the parameter values range between 5 to 12 inches. That gives a fairly broad testing range. We choose 2 values less than the default one, and 2 greater, centering around the original.

The actual width and length ratio is obtained through a procedure described in Section 6.2.2. Again, 5 values of this parameter were tested, with the parameter value in the middle. Since external defects in general have a width/length ratio in the range of 0.3–0.7, that is the reason we choose these values. Table 6.10 shows that in fact the original value is the best choice.

In the algorithm during one of the search steps for large defects, the region-length parameter is set to 7 inches. We tested values from 5 to 9 inches. The interesting aspect of this

Table 6.11: *Parameter testing results for rectangle (region) length during the search for one of the large-defect groups (in inches).*

Value	5	6	**7**	8	9
Correct	63	63	**63**	63	63
False	10	10	**10**	10	10
Unidentified	5	5	**5**	5	5

Table 6.12: *Parameter testing results for width/length ratio during the search for one of the large-defect groups.*

Value	0.10	0.15	**0.20**	0.25	0.30
Correct	63	63	**63**	60	57
False	11	10	**10**	9	6
Unidentified	5	5	**5**	8	11

testing is that all outcomes are the same for all values. The procedure was double-checked to make sure it was was set up properly, and that correct values are used by the algorithm. Yet we obtained the same results. To ensure there is no bug in the programs, we used these values: -1,000, -500, 0, 500, and 1,000 for debugging. Evidently they are unreasonable for the detection purposes. Among all 14 log samples, only one got different results. It has the same number of correct identifications for all five values, which is 6, and the same number of unidentified defects, 0. For values -1,000, -500, and 0, there is no false identification, but for both values 500 and 1,000, there is 1 false identification. This indicates the algorithm is not sensitive to the reasonable range of values.

The width and length ratio parameter is examined during the process of identifying a group of large defects. The default is 0.2. Five values were tested, including the original, which generates the best number of correct identifications, and the lowest number of unidentified defects. However at the original value, the false-identification result is neither the greatest, nor the smallest, compared to those of the rest values. The value (0.3) that returns the least number of false identifications also yields the lowest number of correct identifications and highest number of unidentified defects.

Table 6.13: *Parameter testing results for data point interval during the identification of flat tops (number of points).*

Value	3	4	**5**	6	7
Correct	56	59	**63**	61	62
False	7	10	**10**	16	15
Unidentified	12	9	**5**	7	6

Table 6.14: *Parameter testing results for angle change during the identification of flat tops (in degrees).*

Value	15	20	**25**	30	35	40
Correct	52	62	**63**	62	61	60
False	8	7	**10**	14	16	17
Unidentified	16	6	**5**	6	7	8

The parameter of data point interval is critical for the identification of flat tops. It determines how far apart are two data points connected by a line whose angle is calculated. The angle changes are then inspected for "straight line segment"—a sign of the existence of a flat top. Along a cross section, neighboring data points are approximated 0.02 inch apart, thus testing values between 3 and 7 points are equivalent to roughly 0.06 and 0.14 inch. Too small a value will make the algorithm look in too much detail in terms of "straightness", but too big a value, would make the algorithm ignore the changes that reflect the "straightness" or roughness. Testing results demonstrate that the default value is the best at number of correct identifications and number of unidentified defects, but not the number of false identifications.

Similarly, the parameter of angle change is also critical to the procedure that identifications flat tops. Our past experiments showed that 25^o is an appropriate threshold, hence it was chosen as default. Five different values were tested, all evenly spaced. We decided 15^o is as small as it should be, as too small will be too strict. Recall that even a sawn top is not completely straight and/or smooth, it is only so within a certain tolerance level. 40^o is sufficiently large, because if it is too big, all type of surfaces might be selected, regardless it

is flat or not. Once again, the default value is associated with a high false identification, but the best results for the rest.

The reason these 10 parameters were chosen are:

1. These are the most important parameters for the algorithm. The change to their values will produce significantly different results. For example, the contour level determines at what radial distance possible defect regions were identified. All subsequent computation depends on this value. The "region size" cut-off of is very important as well, as it determines which regions are selected for the rest of algorithm.

2. Parameters 1 through 6 are used in the early part of the algorithm, and make a difference to all regions. For instance, the cut-off value of bark size that keeps some regions from being selected for further consideration, and the 4 paddings of rectangle regions (parameters 4 and 5).

There are many more parameters used by the algorithm that were not tested, because they only matter to the identification of a small subset of the regions. In other words, changing them will not heavily influence the entire algorithm. For example, there is a parameter used as the ratio between the width of the maximum consecutive segments, and the bounding box width. If the number of segments with a value smaller than the default ratio is high enough, then the region may be flagged as bark, should other conditions be met as well. Since this is but one of several parameters that are applied in decision making, and used only in this place, currently it is not included in this test.

In summary, for each parameter, the total number of matches at a certain value of all the log samples is calculated. The same was done for both false identification and unidentified defects. From the testing results such as in Tables 6.5 to 6.7, it was found that among 10 parameters,

1. for 9 parameters, the total number of matches at the original parameter value is the

highest. The only exception is the bark region cut-off size;

2. for 5 parameters, the total number of false identifications at the original parameter value is the lowest, where the exceptions are contour levels, cut-off size, width/length ratio, data point interval for detecting flat tops, and angle change for detecting flat tops;

3. for 9 parameters, the total number of unidentified defects at the original parameter value is the lowest. This property corresponds to that of the first property for the number of matches. That is because the sum of matching numbers and unidentified defects is always a constant, which equals the total number of ground truth.

This indicates that at the original parameter values, the algorithm tends to identify defects correctly, but there is a likelihood to claim a region is defective but in truth it is not. However, for all parameters the numbers of false identifications at the original values are not much greater than the lowest ones. In other words the algorithm with the original parameter values is oversensitive to a low degree.

6.6 Experiments with Data Mining

Data Mining (DM) theories and algorithms [16, 22, 50, 81] were explored as an alternative way to implement defect detection. An online survey was conducted, as well as a literature search on the subject to determine potential DM algorithms. Based on this survey, the following data mining strategies were investigated in detail:

- K-means clustering [98, 41, 51, 81], where n-dimensional centroids are predetermined. The algorithm assigns each object to its "closest" centroid, optimizes the classification metric, and modifies positions of K centroids until stopping criteria are satisfied.

- Spatial Aggregates Language (SAL) in Active Data Mining [59, 60], a general DM framework. With SAL, objects being mined are constantly changing in the sense that detailed, low-level ones (e.g., a dot) are aggregated to obtain objects that are more abstract and at a higher level (i.e., a set of dots forming a line). The process is repeated until the desired objects (a hill composing several curvy lines) are classified. Through the iterative processes, lower-level object features may be employed in decision making during high-level aggregations.

- Decision Trees (DT) [98, 99, 50, 57, 58, 81], where every object passes through a certain path in a tree which defines the process. Nodes of the tree correspond to sub tasks in the algorithm.

We first considered K-means clustering and other similar clustering algorithms. The principles of these methods are straightforward. Further, for the log data only two centroids are needed, representing the classes "defects" and "non-defects". However, the fact that parameters in the detection algorithm (such as bounding rectangle widths and lengths) are constantly being changed throughout the algorithm makes it impossible to establish "static" rules, which the clustering algorithm needs to implement the optimization procedure.

Next, the SAL approach was studied. Our hope was that by applying its methodology, we could come up with a data mining algorithm that would hierarchically aggregate objects from the lowest level to the highest. This would result in log surface regions being classified as either defective or non-defective. An outline of the SAL algorithm is available in [59], which was adopted to the log-defect detection algorithm. The following is the general procedure.

Level 1 (Points):

1. For each log data point, determine and normalize the gradients of its radial distance [89], both along the x_2 and x_3 directions;

2. Determine if a point is "aligned with" any of its 8 neighbors (left, right, top, bottom, upper-left, upper-right, lower-left, and lower-right).

Level 2 (Curves):

3. Connect the aligned points to form a "forward" graph;
4. For each data point, determine its "best forward neighbor" by penalizing for distance;
5. Reverse the forward graph to generate a corresponding "backward" graph;
6. Determine the "best backward neighbor" using the same strategy as in 4. At this point, each point is connected by at most two adjacent points in a trajectory.

Level 3 (Regions):

7. Group converging trajectories to form a region (e.g., a pocket, or a bump).

The problem with this approach is that the barked log surface [88] is far more complicated than any SAL example found in literature:

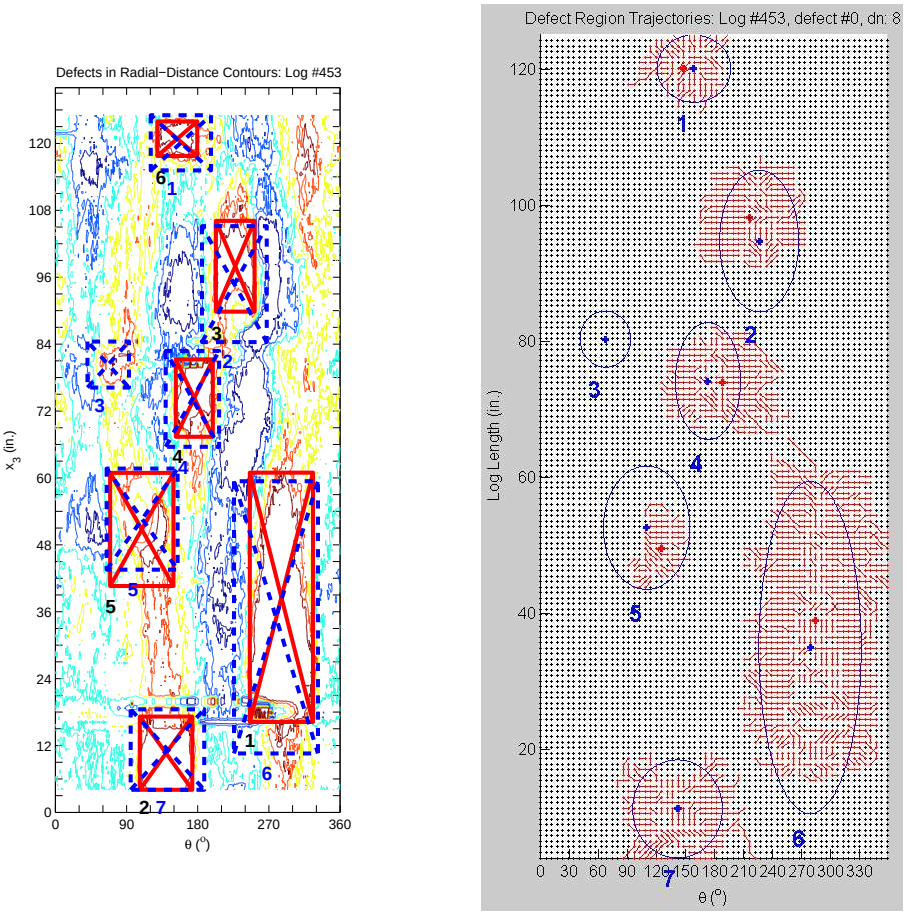
- it is covered under bark, which is a mixture of ridges and grooves in an irregular fashion;
- since a log is not perfectly circular or elliptical along cross section, the "unrolled" surface created by using fitted circles (or any other simple geometrical shape) inevitably introduces bulging or depressing regions along log direction, camouflaging true defect bumps;
- knots (defects) are covered under a complex bark pattern;
- regions with missing bark (that falls off before a log gets scanned) result in dented portions, adding more noise;
- missing data and outliers in data also make log surfaces more complicated to process.

These problems make it unrealistic for SAL programs to single out defects from the "messy" background. Simulation results were not satisfying, with results not even close

in quality to those from the detection algorithm described in Section 6.2.1. Further, we also tried applying SAL as an assisting tool to the original algorithm in that, whenever a region is determined, the best-forward neighbors are flagged as defective as well. The only change between this approach and the original is that detected regions are irregularly shaped, instead of a rectangle (Figure 6.8(b)). We had hoped that they could show the regions in more natural shapes, and more similar to the ground truth. However, that is not quite the case, perhaps because the SAL programs apply decision rules that makes sense mathematically, but meaningless in the graphical representation of defects.

As the SAL experiment was not successful, decision tree algorithms were considered. We chose non-continuous parameter decision tree algorithms, obtained a publicly available package, C4.5, which is implemented in C for Unix operating system [99, 58, 57]. Non-continuous parameter decision tree algorithms are less complex than the continuous-parameter kind. However, they are also not as effective, because they learn only axis-parallel hyperplanes, while the latter allows for better parameter tuning. As we were looking into C4.5, and planning to prepare the attribute-data file for the programs to use in defect classification, we realized the same causes as for K-means clustering is an issue here: decision tree algorithms only deal with predetermined or static attributes, and the machine-vision system algorithm modifies bounding boxes throughout the entire algorithm. In other words, to apply a decision tree algorithm would require a complete rewrite of the detection algorithm. Thus, we did not pursue this any further for now.

Due to the unique features of the defect detection algorithm using an machine-vision system and the complexity of the data, it is hard to quickly adapt it into any Data Mining algorithm. In future, we may start building a data mining system from scratch that performs the detection task.



(a) (b)

Figure 6.8: *Comparison of the defect detection algorithm using the machine-vision system and that of an data-mining approach. (a) Results from the detection algorithm using an machine-vision system. Solid rectangles enclose detected regions, and dashed ones, ground truth. (b) Results from combining the machine-vision system algorithm with SAL functions. The ellipse encloses ground truth. The two bold crosses are the center of detected region, and the ellipse center, respectively. Due to the complex structure of the defect, it is only partially detected.*

Chapter 7

Summary and Future Work

7.1 Summary

This research has created the first automated algorithm for detecting surface defects in hardwood logs using 3-D laser-scanned profile data. Due to the presence of extreme outliers and missing data in the laser log data set, robust estimation techniques are well suited to this application. The developed programs can process an entire log-data sample by transforming the original log data set, which may contain a large number of missing and/or severe deviant data, into a matrix of radial distances that better portrays surface defects. This is illustrated as a sharper and cleaner gray-level image. This shows that the radial distances lay a solid foundation for the remaining defect-detection process. It is found that contour levels derived from the radial distances make it possible to detect and further narrow down the potential defect regions. For defects that lie within the bark layer, information other than the radial distances will need to be used. This calls for further research and development.

A new robust GM-estimator has been developed that performs 2-dimensional circle fitting to detect external defects on hardwood logs and stems. Classical estimation methods based on the least-squares method revealed themselves to be unreliable because they gener-

ate strongly biased estimates due to the presence of missing data and severe outliers. This is shown in Figure 5.3. By contrast, the GM-estimator suppresses these outliers via weights calculated from projection statistics applied to the radial distances, thereby bounding the influence function of the estimator. Based on these robust circle fittings, the defect-detection programs transform the original log data into a sharper and cleaner gray-level image, determine contour levels of the radial distances, and further narrow down the potential defect regions.

We also developed a computer algorithm that identifies external defects using the radial distances generated by the circle fitting method by applying the new GM-Estimator. The generation and initial processing of radial distances are not the final steps of this work. Clearly, additional research is needed. At this point, only log unrolling and height analyses methods have been examined. A preliminary study was conducted to extract features of external defect types from randomly chosen defect samples. These features were studied to help making decision rules for the defect defection algorithm. To reach the final goal of locating and classifying surface defects, we are exploring the potential benefits of image processing, computer vision, and pattern recognition techniques using radial-distance data.

There are two methods to evaluate performance of the detection algorithm. The first one looks at the number of defects detected out of the total number of ground truth. In our experiments there are a total of 68 severe defects, of which 63 were correctly identified. There are 10 non-defective regions falsely identified as defects. The other way is calculating the surface areas that are detected against that of ground truth. To calculate this, we implemented the method to compute the false-detection rate as discussed in Section 6.2.4, which demonstrated a reasonably good algorithm (87.5% automated clear region vs. 88.5% observed clear region, and a 97.6% area detection rate). There are pros and cons with both methods. The pros for the first one is that it agrees with conventional understanding of detection rates. However, in wood science and forest products, a large defect usually is a lot worse than a small one. The second method overcomes this problem. Yet due to the relative inaccuracy of the defect area, the statistics may not be completely reliable. This is because

the area of each defect region is estimated as a rectangle, using the width and length of the matched ground-truth defect.

Many defects were not identified mainly because they do not have a significant height change. Thus, the height-based approach is not effective for these defects. Among them there is a group of severe defects with heavy distortions and flat knots. These defects often have a distinctive ring-like bark pattern. Edge detection, a computer vision technique, may help in identifying such defects. This will be implemented in a second phase of this research.

When a single cylinder is fitted to the entire log data, the number of parameters to be estimated is the fewest as compared to fitting a sequence of circles and ellipses to all cross log sections. This means that cylinder fitting provides the fewest degrees of freedom. In addition, the radial distances are extracted against a uniform surface, resulting in the smoothest image among the three. Clearly this gives a more consistent surface map for subsequent tasks. However, it may reflect less details of defective regions, as the surface of a simple cylinder like this tends to resemble very little of a log surface. Thus, radial distances between the cylinder model and log data will give few details of log surface structure, critical to the detection task.

In contrast, the circle fitting approach involves far more parameters to be estimated, which results in more degrees of freedom. However, each circle provides a better fit to each individual cross section, revealing more details on log surface while radial distances extracted between neighboring cross sections are less consistent, or noisier, than in the cylinder case. On the other hand, ellipse fitting introduces the greatest number of estimated parameters and hence, generates the most detailed radial-distance image. By the same token, radial distances from neighboring cross sections are much less consistent, or less crisp, compared to circle- or cylinder-fitting. This is primarily due to the difference of axes orientation between neighboring ellipses.

Generating and processing radial distances is not the final step of this work. An algorithm was developed that determines whether an area of interest contains a sawn knot, by locating

the approximately straight line segment in a cross-section. Parameter testing was conducted and demonstrated that the detection algorithm was capable of identifying most defects. Further, a preliminary study was conducted to extract features of external defect types from randomly chosen defect samples.

7.2 Future Work

Recently judged by our achievements, the USDA Wood Education and Resource Center (WERC) has granted new funding for the refinement and analysis of the log surface defect methods. It will provide partial financial support for our future work. Our plans include the following.

Develop a Java software package The Matlab defect-detection code that detects defects will be converted to Java and integrated with the scanning and sawing equipment. A complied Java program can be run directly on the Java virtual machine of any architecture, given version compatibility. We plan to provide a complete package that is publicly available through internet. Further, the detection results will be displayed in graphical formats to assist sawyers who can rotate, zoom, and move the virtual log marked with defects (Figure 4.3).

Overall, Java is a good choice for the real time processing and user interaction demands of this project. In the recent releases of Java 1.4.2 and later, Java's mathematical operations were further optimized to improve performance. Recent benchmarking studies show that the performance differences among Java, C, and C++ for most mathematical operations are minimal [10, 37]. Conwell-Shah reported that for integer and double-precision, Java actually outperformed gcc C code by 9.5% and 32% respectively. Similar results are reported in [37]. Lea also reports that using the server Java virtual machine (JVM) results in significantly faster execution times than when using the client JVM. One weakness with the JVM is with

trigonometric functions where C outperforms Java by as much as 33-percent.

Further, development of packages such as NINJA (Java for High Performance Numerical Computing) have made Java an even more attractive candidate for this research [52]. NINJA supports improved matrix and vector handling and faster mathematical operations on these data types. The benchmarks presented by Moreira et al. are based on a pre 1.4.2 JVM whose mathematical operations were not fully optimized. The study showed that Java with NINJA scored within 15 percent of the benchmark score of Fortran 90 on the matmul benchmark. Similarly, NINJA scored within 2.9 percent of Fortran 90 on the Cholesky benchmark. However, NINJA exceeded the Fortran 90 score on the microdc benchmark by 2.4 percent.

Fortran and C are efficient languages for scientific computation. However, Java is portable. Any compiled Java program can be executed on any platform as long as a JVM for that platform exists. Java has excellent graphical user-interface development capabilities, and has several GUI developmental packages. One notable GUI development package for Fortran is japi (`http://www.japi.de`), which provides the Java AWT Toolkit to non-object oriented Languages like C and Fortran. However, Fortran has limited GUI development support that is not integrated as well as those available in Java and C++.

Experiment with more log samples We would like to obtain more log samples, and capture profile data using Perceptron's 3-D scanning equipment [4]. From past experience, there are a few things that can be improved:

- the process of manually marking and labeling surface defects on the logs should take place prior to taking side photographs are taken. This will make the defects obvious in the heavily camouflaged bark surface, and thus make comparing simulation results of the detection algorithm with the ground truth and photographs a lot easier.
- we need to keep close watch on the scanned data, photos, and ground truth data entry. In the current experiment, some log data are unusable; some side-view log photos were

arranged in the wrong order; and ground truth data were incorrectly entered and/or some fields were missing. All these reduce the number of log samples that can be used to test the algorithm.

In future work, once such problems occur, we will fix it immediately.

Detect more types of defects We implemented the method to compute the false-detection rate as discussed in Section 6.4, which demonstrated a reasonably good algorithm. Many defects were not identified mainly because they do not have a significant height change. Thus, the contour-based approach is not effective for these defects. Among them is a group of defects that are severe, for example, heavy distortions and flat knots. These defects often have a distinctive ring-like bark pattern. Edge detection, a computer vision technique, may help in identifying such defects.

Classify defects Cluster analysis, or clustering, is an attempt to find structure in a set of observations [53]. Clustering techniques are used in two general classes of problems: those with unlabeled samples, referred to as unsupervised learning; and those with labeled sets in which given classes may consist of distinct subsets, referred to as supervised learning. Clusters are aids to interpreting and evaluating the measurements and features. Techniques such as splitting, merging, and graph theory are applied, each associated with a different criterion for assigning an object to a cluster. Objects and patterns are referred to as points in feature space. Patterns are represented in terms of features, which form n-dimensional feature vectors [85]. Approaches to clustering include error function minimization, hierarchical, and graph-theoretical clustering. The basic steps to develop a cluster algorithm are the following: feature selection, proximity measure, clustering criterion, clustering algorithms, validation of the results, and interpretation of the results. Proximity measures include dissimilarity and similarity measures, each defined by its metric [12].

We shall develop methods and algorithms for feature extraction, defect segmentation,

and classification using cluster analysis. By analyzing the characteristics of the contour levels, defects with a significant rise or depression will be located, segmented, and classified through cluster analysis. To this end, classification criteria will be set up and clustering algorithms will be developed. This task will accomplish the following classification tasks: feature selection; proximity measure; validation of the results; and interpretation of the results [12, 53, 85].

Ultimately, accurate defect locations need to be pinpointed, defect features extracted, and the final detection of external defects performed. To this end, algorithms in pattern recognition, including cluster analysis, will be investigated. Other methods, such as surface reconstruction and texture analysis, will be examined as well wherever necessary. Note that the pattern recognition methods are different from those advocated in computer vision, where image elements are categorized into identifiable classes. Here, a learning set must be built first through extensive simulations by grouping the defects into separate classes. Pattern recognition will be carried out through cluster analysis, which will be the primary method for feature extraction and defect detection of the log surface data.

Defects may be classified as knots, splits, holes, and bark distortions, each of which is characterized by a set of features that identify a cluster of points in an n-dimensional space. The features are chosen in such a way that two similar defects will have close points while dissimilar defects will have remote points in the feature space. In other words, when the features are appropriately chosen, two dissimilar defects have their associated feature points distant from each other, resulting in a clear separation between the clusters.

For the log data set, the features can be determined from the radial distances of the curve or surface fitting and the corresponding contours. Boundaries between clusters will have to be determined as well. This is again a curve-fitting problem, and robust techniques will be used here. Assume that the features of a defect are characterized by the following n variables: $\{y_1, y_2, y_3, \ldots, y_n\}$. There are several features that may characterize a given defect, including slopes, length, height, width, elongation, the boundary information of its

enclosed region, and the number of its neighboring contours with the same level within a certain range. The latter would indicate whether the rise is just a high-rise bark region or a real defect. Note that elongation refers to the ratio between the width (along the horizontal direction) and the length (along the vertical direction). The width of a contour is defined as the difference between the minimum and the maximum horizontal values of the contour, and its length, the difference between the minimum and the maximum vertical values. Bark regions in high-level contours tend to have a smaller elongation value than most defects do.

Most defects are associated with contours of the highest or lowest levels. Thus, our focus is on detecting defects in the regions inside such contours. Note that a contour is represented by its level and a set of data points on its path. The most common defects are medium-sized knots, about 4 to 6 inches in diameter, and 1-2 inches high. This indicates that we can group the corresponding contours in one class, call it Class 1. These contours are characterized by the size and elongation of the regions enclosed in them. However, some bark regions with the same contour level also have a similar size and elongation, the only difference is that there are small contours with the same level scattered around them. Using this feature, one can rule out the contour enclosing bark regions as false defects; such a contour defines a second class (Class 2). There are high-level contours enclosing large-sized regions (about $15 \times 20 \times 2.5$ inch^3). The ones with an elongation value that is not too small are likely to enclose defects, call it Class 3. For the ones with a small elongation value, their enclosed regions include three different cases:

1. A large defect with relatively straight edges along the vertical direction, defining Class 4;

2. A large bark region also with relatively straight edges along the vertical direction, resulting in Class 5;

3. A large defect with elliptical shaped edges along the vertical direction, yielding Class 6.

The main difference between Classes 1 and 2 is that a bark region tends to contain long

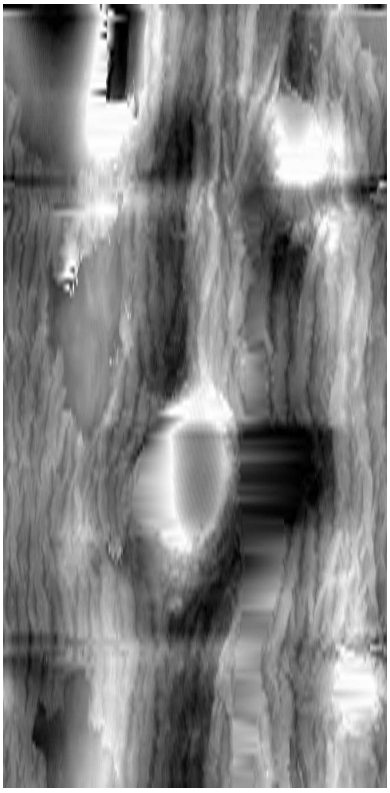


Figure 7.1: *Radial-distance image for a red oak log.*

and narrow strips, so the ratio between the median of its "solid" widths and its length is smaller than that of a defect. Here, the "solid" width refers to the one between two contour segments in the vertical direction where there is no break. The contour plot in Figure 7.1 contains a typical large bark region in the bottom-left portion at about 90^o. The main difference among Classes 1, 2 and 3 is that the first two are associated with relatively straight contour segments in the vertical direction while the third one has elliptical contours. This indicates that the scale estimator, determined via Median Absolute Deviation from the Median (MAD), of the slopes on the boundary for the first two cases will be smaller than that of the third. This separates the third into a different class. For defects in the shape of holes, splits, similar features apply, except that their contour levels are low.

Once features are defined, we can define the training set of defect classes. To this end, we

carry out a mapping between known defect classes and their associated clusters of points in the feature space. Specifically, these features will be measured for a sample of a given defect class, determine its center (i.e., coordinate-wise medians) and a 95% confidence ellipsoid that defines its boundaries. We will repeat this for every given class of defects. Once the training set is completed, we may then use it to classify any potential defect by estimating its features and finding the closest cluster. When the number of classes is large, cluster analysis is carried out with the help of decision trees. Neural networks may also be investigated here for identifying the clusters.

Improve the algorithm efficiency Various numerical methods for solving nonlinear equations will be investigated to speed up the algorithms while making them numerically stable. We will improve the method so it would be fast and numerically robust. We are presently using the iteratively reweighted least squares (IRLS) method together with QR decompositions and Householder reflections for numerical stability [56]. The execution time constraint cannot be ignored because the system targets lumber manufacturing. On average, it takes about 8 to 10 seconds for a human expert to examine a log. Eventually, the system to be developed must operate within the same time frame or less. We will investigate ways to speed up the IRLS algorithm.

Integrating both the circle-fitting and defect detection algorithms in Java makes it possible to be accessed via internet by public. This makes our system available to the forest product society in that researchers may use it in their simulations, and sawmills may use it to help improve their productivity. Further, written in Java also means the software can be ready in executable form for various platforms. Users may simply download it and it is ready to run on their computers. Developing a GUI for the system will make it a lot easier to operate. The detection results can be viewed on screen, the operators may zoom in or out, rotate, and/or move the virtual log to get a better look of the defect size, shape and distribution on the log. Detect information can also be displayed at the operator's request. All these could be accomplished by a few key strokes.

Experimenting with more log samples will help us to test the algorithm, collect detection statistics about it. As we have seen through this document, only 14 log samples were used, and among them the majority are red oak. To improve and test the algorithm, we certainly need significantly more samples for both red or black oak and yellow poplar. Currently our algorithm is only capable of detecting surface defects associated with height change. However, defects without significant surface rise, such as heavy distortion are also severe and need to be detected. To be able to detect them we shall develop an algorithm. Knowing what type of defects, for instance, a knobby knot caused by a broken branch, or a sawn top, would provide more information for the inference of the internal defect features. Internal defect information is crucial to wood processing. Thus, we would like to classify the external defect types. Finally, we want to improve the algorithm efficiency is because it implies fast execution and less breakdowns, key factors to a high-quality, useful, and robust software.

Bibliography

[1] T. W. Anderson. *An Introduction to Multivariate Statistical Analysis*. John Wiley, second edition, 1984.

[2] P. J. Besl, J. B. Birch, and L. T. Watson. Robust window operators. *Machine Vision and Applications*, 2:179–191, 1989.

[3] S. M. Bhandarkar, T. D. Faust, and M. Tang. Catalog: a system for detection and rendering of internal log defects using computer tomography. *Machine Vision and Applications*, 11(4):171–190, 1999.

[4] C. Blomquist. Perceptron mill study: Feasibility analysis of knot detection at fletcher challenge kpp log processing plant. Technical report, Perceptron Inc., Farmington Hills, MI, 1999.

[5] P. T. Boggs, J. R. Donaldson abd R. H. Byrd, and R. B. Schnabel. Algorithm 676—odrpack: Software for orthogonal distance regression. *ACM Transactions on Mathematical Software*, 15(4):348–364, 1989.

[6] P. T. Boggs, R. H. Byrd, and R. B. Schnabel. A stable and efficient algorithm for nonlinear orthogonal distance regression. *SIAM Journal of Scientific and Statistical Computing*, 8(6):1052–1078, 1987.

[7] R. D. Carpenter, D. L. Sonderman, E. D.Rast, and M. J. Jones. Defects in hardwood timber. Technical Report 678, USDA Forest Services, Washington, D.C., 1989.

[8] S. J. Chang. External and internal defect detection to optimize cutting of hardwood logs and lumber. Technical Report 3, USDA and National Agricultural Library, Beltsville, MD, 1992.

[9] L. D. Cohen. On active contour models and balloons. *Computer vision graphics and image processing: Image Understanding*, 53(2):211–218, 1991.

[10] C. Conwell-Shah. *Nine Language Performance Round-Up: Benchmarking Math and File I/O*, 2004. AVAILABLE: `http://www.osnews.com/printer.php?news_id=5602`.

[11] D. L. Donoho. *Breakdown Properties of Multivariate Location estimators*, 1982. QAULIFYING Paper.

[12] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification*. Wiley, New York, second edition, 2001.

[13] D. Eberly. *Least squares fitting of data*, 1999. AVAILABLE: `http://www.geometrictools.com/Documentation/LeastSquaresFitting.pdf`.

[14] A. Fitzgibbon, M. Pilu, and R. B. Fisher. Direct least square fitting of ellipses. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(5):476–480, May 1999. AVAILABLE: www.robots.ox.ac.uk/ awf/ellipse/ellipse-pami.pdf.

[15] *Cancer Dictionary*, 2006. AVAILABLE: `http://www.answers.com/topic/false-positive`.

[16] J. Frand. *Data Mining: What is Data Mining?*, 2006. AVAILABLE: `http://www.anderson.ucla.edu/faculty/jason.frand/teacher/technologies/palace/datamining.htm`.

[17] W. Gander, G. H. Golub, and R. Strebel. Fitting of circles and ellipses-least squares solution. Technical Report 217, Insitiut fur Eisenschaftliches Rechnen, ETH Zurich, 1994. AVAILABLE: `ftp://ftp.inf.ethz.ch/doc/tech-reports/2xx/`.

[18] M. Gardiner. The superellipse: A curve between the ellipse and the rectangle. *Scientic American*, 213:222–234, September 1965.

[19] M. Gasko and D. Donoho. Influential observation in data analysis. In *Proceedings of the Business and Economic Statistics Section*, pages 104–110, 1982.

[20] S. Guddanti and S. J. Chang. Replicating sawmill sawing with topsaw using ct images of a full length hardwood log. *Forest Products Journal*, 48(1):72–75, 1998.

[21] F. R. Hampel, E. M. Ronchetti, P. J. Rousseeuw, and W. A. Stahel. *Robust Statistics: The Approach Based on Influence Functions*. John Wiley, 1986.

[22] D. Hand, H. Mannila, and P. Smyth. *Principles of data mining*. MIT Press, Cambridge, Mass., 2001.

[23] R. M. Haralick and L. Shapiro. *Computer and robot vision*, volume 2. Addison-Wesley, 1992.

[24] R. M. Haralick, L. A. Watson, and T. J. Laffey. The topographic primal sketch. *The International Journal of Robotics Research*, 2(1), 1983.

[25] E. S. Harrar and R. A. Campbell. The major defects in southern hardwood veneer logs and bolts. Technical Report SE-19, U.S. Department of Agriculture, Forest Service, Southeastern Forest Experiment Station, Asheville, NC, 1966.

[26] J. G. Harris. A new approach to surface reconstruction: the coupled depth/slope model. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 259–268, London, UK, 1987.

[27] D. G. Hodges, W. C. Anderson, and C. W. McMillin. The economic potential of ct scanners for hardwood sawmills. *Forest Products Journal*, 40(3):65–69, 1990.

[28] R. V. Hogg and J. Ledolter. *Engineering Statistics*. Macmillan Publishing Company, New York, 1987.

[29] P. W. Holland and R. E. Welsch. Robust regression using iteratively reweighted least squares. *Communications in Statistics-Theory and Methods*, A6:813–827, 1977.

[30] P. J. Huber. *Robust Statistics*. John Wiley, 1981.

[31] M. J. Hyvärinen. *Measuring quality in standing trees—Depth of knot-free wood and grain orientation under sugar maple bark distortions with underlying knots*. PhD thesis, University of Michigan, 1976.

[32] R. Jain, R. Kasturi, and B. G. Schunck. *Machine Vision*. McGraw-Hill International Editions, 1995.

[33] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 259–268, London, UK, 1987.

[34] A. I. Khuri. *Advanced Calculus with Applications in Statistics*. John Wiley & Sons, 1993.

[35] D. Y. Kim, J. J. Kim, P. Meer, D. Mintz, and A. Rosenfeld. Robust computer vision: A least median of squares based approach. In *Proceedings of a Workshop on Image Understanding*, pages 1117–1134, Palo Alto, CA, 1989.

[36] D. E. Kline, A. Widoyoko, J. K. Wiedenbeck, and P. A. Araman. Performance of color camera machine vision in automated furniture rough mill systems. *Forest Products Journal*, 48(3):38–45, 1998. AVAILABLE: `http://www.srs4702.forprod.vt.edu/pubsubj/pdf/9826.pdf`.

[37] K. Lea. *Java vs C++ "Shoutout" Revisited*, 2004. AVAILABLE: `http://java.sys-con.com/read/45250.htm`.

[38] Y. Leedan and P. Meer. Heteroscedastic regression in computer vision: Problems with bilinear constraint. *International Journal of Computer Vision*, 37(2):127–150, 2000.

[39] H. Lemieux, M. Beaudoin, and S. Y. Zhang. Characterization and modeling of knots in black spruce (picea mariana) logs. *Wood and Fiber Science*, 33(3):465–475, 2001.

[40] P. Li, A. L. Abbott, and D. L. Schmoldt. Automated analysis of ct images for the inspection of hardwood log. In *Proceedings of the International Conference on Neural Networks*, volume 3, pages 1744–1749, Washington, D.C., USA, 1996.

[41] B. T. Luke. *K-Means Clustering*, 2006. AVAILABLE: `http://fconyx.ncifcrf.gov/lukeb/kmeans.html`.

[42] Y. Mainguy, J. B. Birch, and L. W. Watson. A robust variable order facet model for image data. *Machine Vision and Applications*, 8:141–162, 1995.

[43] R. M. Marden and C. L. Stayton. Defect indicators in sugar maple: A photographic guide. Technical Report Research Paper NC-37, North Central Experiment Station, US Department of Agriculture, Forest Service, 1970.

[44] B. Matei and P. Meer. Reduction of bias in maximum likelihood ellipse fitting. In *Proceedings of the 15th International Conference on Pattern Recognition*, 3, pages 802–806, Barcelona, Spain, September 2000. IEEE Computer Society Press, Los Alamitos, CA.

[45] P. Meer, D. Mints, and A. Rosenfeld. Least median of squares based robust analysis of image structure. In *Proceedings of Image Understanding Workshop, DARPA*, pages 231–254, 1990.

[46] P. Meer, D. Mints, A. Rosenfeld, and D. Y. Kim. Robust regression methods in computer vision: A review. *International Journal of Computer Vision*, pages 59–70, 1991.

[47] W. Mendenhall, D. D. Wackerly, and R. L. Scheaffer. *Mathematical Statistics with Applications*. PWS-Kent Publishing Co., 4th edition, 1990.

[48] L. M. Mili, M. G. Cheniae, and P. J. Rousseeuw. Robust state estimation of electric power systems. *IEEE Transactions on Circuits and Systems-1: Fundamental Theory and Applications*, 41(5):349–358, 1994.

[49] L. M. Mili, M. G. Cheniae, and P. J. Rousseeuw. Robust state estimation based on projection statistics. *IEEE Transactions on Power Systems*, 11(2), May 1996.

[50] T. M. Mitchell. *Machine Learning*. McGraw-Hill, New York, 1997.

[51] A. W. Moore. *Statistical Data Mining*, 2006. AVAILABLE: `http://www.autonlab.org/tutorials/`.

[52] J. Moreira, S. Midkiff, M. Gupta, P. Wu, P. Artigas, and G. Almasi. Ninja: Java for high performance numerical computing. *Scientific Programming*, 10(1):19–33, 2002.

[53] M. Nadler and E. P. Smith. *Pattern Recognition Engineering*. Wiley, 1993.

[54] S. P. Neugebauer. Robust analysis of m-estimators of nonlinear models. Master's thesis, Dept. Elec. And Comp. Eng., Virginia Tech, Blacksgurg, VA, August 1996. AVAILABLE: `http://scholar.lib.vt.edu/theses/available/etd-22820699602791/`.

[55] S. P. Neugebauer and L. M. Mili. Local robustness of nonlinear regression m-estimators. In *Proc. IEEE/EURASIP Workshop on Nonlinear Signal and Image Processing*, Mackinac Island, Michigan, 1997.

[56] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery. *Numerical Recipes in C, the Art of Scientific Computing*. Cambridge University Press, second edition, 1997.

[57] J. R. Quinlan. Induction of decision trees. *Machine Learning*, 1(1):81–106, 1986.

[58] J. R. Quinlan. Comparing connectionist and symbolic learning methods. *Computational Learning Theory and Natural Learning Systems: Constraints and Prospects*, pages 445–456, 1994.

[59] N. Ramakrishnan and C. Bailey-Kellogg. Sampling strategies for mining in data-scarce domains. *IEEE/AIP Computing in Science and Engineering (CiSE)*, 4(4):31–43, July/Aug 2002.

[60] N. Ramakrishnan, C. Bailey-Kellogg, S. Tadepalli, and V. N. Pandey. Gaussian processes for active data mining of spatial aggregates. In *Proceedings of the SIAM International Conference on Data Mining*, Newport Beach, CA, 2005.

[61] A. R. Rao and B. G. Schunck. Computing oriented texture fields. *Computer Graphical Models and Image Processing*, 53(2):157–185, 1991.

[62] E. D. Rast. Photographic guide of selected external defect indicators and associated internal defects in northern red oak. Technical Report Research Paper NE-511, Northeastern Forest Experiment Station, US Department of Agriculture, Forest Service, 1982.

[63] E. D. Rast and J. A. Beaton. Photographic guide of selected external defect indicators and associated internal defects in black cherry. Technical Report Research Paper NE-560, Northeastern Forest Experiment Station, US Department of Agriculture, Forest Service, 1985.

[64] E. D. Rast, J. A. Beaton, and D. L. Sonderman. Photographic guide of selected external defect indicators and associated internal defects in black walnut. Technical Report Research Paper NE-617, Northeastern Forest Experiment Station, US Department of Agriculture, Forest Service, 1988.

[65] E. D. Rast, J. A. Beaton, and D. L. Sonderman. Photographic guide of selected external defect indicators and associated internal defects in white oak. Technical Report Research Paper NE-628, Northeastern Forest Experiment Station, US Department of Agriculture, Forest Service, 1989.

[66] E. D. Rast, J. A. Beaton, and D. L. Sonderman. Photographic guide of selected external defect indicators and associated internal defects in sugar maple. Technical Report Research Paper NE-647, Northeastern Forest Experiment Station, US Department of Agriculture, Forest Service, 1990.

[67] E. D. Rast, J. A. Beaton, and D. L. Sonderman. Photographic guide of selected external defect indicators and associated internal defects in yellow birch. Technical Report Research Paper NE-648, Northeastern Forest Experiment Station, US Department of Agriculture, Forest Service, 1990.

[68] E. D. Rast, J. A. Beaton, and D. L. Sonderman. Photographic guide of selected external defect indicators and associated internal defects in yellow-poplar. Technical Report Research Paper NE-646, Northeastern Forest Experiment Station, US Department of Agriculture, Forest Service, 1990.

[69] P. J. Rousseeuw. Least-median of squares regression. *J. Amer. Statist. Assoc.*, 79:871–880, 1984.

[70] P. J. Rousseeuw and A. M. Leroy. *Robust Regression and Outlier Detection.* John Wiley, 1987.

[71] P. J. Rousseeuw and B.C. Van Zomeren. Robust distances: Simulations and cutoff values. *Directions in Robust Statistics and Diagnostics*, 34(II):195–203, 1991.

[72] M. Samson. Method for assessing the effect of knots in the conversion of logs into structural lumber. *Wood and Fiber Science*, 25(3):298–304, 1993.

[73] E. Sarigul. *Interactive Machine Learning for Refinement and Analysis of Segmented CT/MRI Images.* PhD thesis, Virginia Polytechnic Institute and State University, Blacksburg, Virginia, 2004.

[74] E. Sarigul, A. L. Abbott, and D. L. Schmoldt. Rule-driven defect detection in ct images of hardwood logs. In *Proceedings of the 4th International Conference on Image Processing and Scanning of Wood*, pages 37–49, 2000.

[75] A. L. Shigo. A tree hurts, too. Technical Report NE-INF-16-73, U.S. Department of Agriculture, Forest Service, Northeastern Forest Experiment Station, Upper Darby, PA, 1973.

[76] A. L. Shigo. Tree decay: An expanded concept. Technical report, U.S. Department of Agriculture, Forest Service, Northeastern Forest Experiment Station, Bromall, PA, 1979. AGRICULTURAL Information Bulletin 419.

[77] G. W. Snedecor and W. G Cochran. *Statistical Methods*. Iowa State University Press, Ames, Iowa, eighth edition, 1989.

[78] W. A. Stahel. *Robuste Schätzungen von Kovarianzmatrizen*. PhD thesis, ETH Zurich, Switzerland, 1981.

[79] P. H. Steele, T. E. G. Harless, F. G. Wagner, L. Kumar, and F. W. Taylor. Increased lumber value from optimum orientation of internal defects with respect to sawing pattern in hardwood logs. *Forest Products Journal*, 44(3):69–72, 1994.

[80] *The Source for Java Developers*, 2006. AVAILABLE: `http://java.sun.com/`.

[81] P. Tan, M. Steinbach, and V. Kumar. *Introduction to data mining*. Pearson Addison Wesley, Boston, 2006.

[82] G. Taubin. Estimation of planar curves, surfaces and nonplanar space curves defined by implicit equations with applications to edge and range image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(11):1115–1138, November 1991.

[83] D. Terzopoulos and D. Metaxas. Dynamic 3-d models with local and global deformations: Deformable superquadrics. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(7):703–714, 1991.

[84] D. Terzopoulos, A. Witkin, and M. Kass. Symmetry-seeking models for 3-d object reconstruction. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 259–268, London, UK, 1987.

[85] S. Theodoridis and K. Koutroumbas. *Pattern recognition*. Academic Press, San Diego, 1999.

[86] L. Thomas and L. M. Mili. A robust gm-estimator for the automated detection of external defects on barked hardwood logs and stems. *IEEE Transactions on Signal Processing*, 2006. Revised.

[87] L. Thomas, L. M. Mili, C. A. Shaffer, and R. E. Thomas. Defect detection on hardwood logs using high resolution three-dimensional laser scan data. In *IEEE International Conference on Image Processing*, pages 243–246, Singapore, October 2004.

[88] L. Thomas, L. M. Mili, R. E. Thomas, and C. A. Shaffer. Automated detection of severe external defects on hardwood logs using robust estimation with high-resolution 3-d laser-scan data. *Wood Fiber Science Journal*, 2006. To appear.

[89] L. Thomas, C. A. Shaffer, L. M. Mili, and R. E. Thomas. Algorithm for the automated detection of severe surface defects on barked hardwood logs and stems. *Forest Product Journal*, 2006. Submitted.

[90] R. E. Thomas. *Predicting Internal Yellow-Poplar Defect Features Using Surface Indicators*. Manuscript.

[91] R. E. Thomas, L. Thomas, L. M. Mili, R. W. Ehrich, A.L. Abbott, and C. A. Shaffer. Primary detection of hardwood log defects using laser surface scanning. In *IS&T/SPIE Electronic Imaging*, volume 5011, pages 39–49, Santa Clara, CA, USA, January 2003.

[92] R. E. Thomas, L. Thomas, C. A. Shaffer, and L. M. Mili. Using external high-resolution log scanning to determine internal defect characteristics. In *the 15th Central Hardwood Forest Conference*. USDA Forest, SE Research Station, 2006. TO be published.

[93] X. Tian and G. E. Murphy. Automated feature extraction and defect recognition from digital images of tree stems using texture analysis. In *Proceedings of the first joint Australia & New Zealand biennial conference on Digital Image & Vision Computing - Techniques and Applications*, 1997.

[94] X. Tian and G. E. Murphy. Detection of trimmed and occluded branches on harvested tree stems using texture analysis. *International Journal of Forest Engineering*, 8(2):65–78, 1997.

[95] A. Tirumalai and B. G. Schunck. A robust method for surface reconstruction. In *Proceedings of the International Workshop on Robust Computer Vision*, pages 183–199, Seattle, WA, 1989.

[96] F. G. Wagner, F. W. Taylor, D. S. Ladd, C. W. McMillin, and F. L. Roder. Ultrafast ct scanning of an oak log for internal defects. *Forest Products Journal*, 39(11/12):62–64, 1989.

[97] R. E. Walpole and R. H. Myers. *Probability and Statistics for Engineers and Scientists*. Collier Macmillan Publishers, thrid edition, 1985.

[98] *A Tutorial on Clustering Algorithms*, 2006. AVAILABLE: `http://www.elet.polimi.it/upload/matteucc/Clustering/tutorial_html/kmeans.html`.

[99] *Decision Tree Software for Classification*, 2006. AVAILABLE: `http://www.kdnuggets.com/software/classification-decision-tree.html`.

[100] D. Zhu and A. A. Beex. Robust spatial autoregressive modeling for hardwood log inspection. *Journal of Visual Communication and Image Representation*, 5(1):41–51, 1994.

[101] D. Zhu, R. W. Conners, F. M. Lamb, and P. A. Araman. A computer vision system for locating and identifying internal log defects using ct imagery. In *Proceedings of the 4th International Conference on Scanning Technology in the Wood Industry*, pages 1–13, San Francisco, CA, 1991. Miller Freeman Publishing, Inc.

[102] D. Zhu, R. W. Conners, D. L. Schmoldt, and P. A. Araman. A prototype vision system for analyzing ct imagery of hardwood logs. *IEEE Transactions on Systems, Man, and Cybernetics Part B: Cybernetics*, 26(4):522–532, 1996.

Vita

Liya Thomas received a B.S. degree in Computer Science from Fudan University, Shanghai, China in 1988, and a M.S. degree in Computer Science from West Virginia University, Morgantown, West Virginia, in 1993. She is presently a doctoral candidate in Computer Science at Virginia Tech. Her research interests are robust statistics theories, computer vision technology and algorithms, image processing theories, and log external defect detection technology.