

AN INTERACTIVE IMAGE ANALYSIS SYSTEM FOR DENDROCHRONOLOGY

by

Giribalan Gopalan

A Thesis Submitted to the Faculty of the

DEPARTMENT OF ELECTRICAL & COMPUTER ENGINEERING

In Partial Fulfillment of the Requirements
For the Degree of

MASTER OF SCIENCE

In the Graduate College

THE UNIVERSITY OF ARIZONA

2000

STATEMENT BY AUTHOR

This thesis has been submitted in partial fulfillment of requirements for an advanced degree at The University of Arizona and is deposited in the University Library to be made available to borrowers under rules of the Library.

Brief quotations from this thesis are allowable without special permission, provided that accurate acknowledgement of source is made. Requests for permission for extended quotation from or reproduction of this manuscript in whole or in part may be granted by the head of the major department or the Dean of the Graduate College when in his or her judgment the proposed use of the material is in the interests of scholarship. In all other instances, however, permission must be obtained from the author.

SIGNED: _____

APPROVAL BY THESIS DIRECTOR

This thesis has been approved on the date shown below:

Robert A. Schowengerdt
Professor of Electrical &
Computer Engineering

Date

ACKNOWLEDGEMENTS

I would like to thank Dr. Malcolm Hughes and Dr. Robert Schowengerdt for devising the concept for TREES and providing invaluable ideas to make the system user-friendly and practical. I am grateful to Dr. Robert Schowengerdt for giving me an opportunity to be a part of and work at the Digital Image Analysis Laboratory.

I wish to thank Martin Munro for his support and assistance throughout the project. I want to thank Steven Conner for having helped me in getting started on the project. I thank Dr. Paul Sheppard for his feedback on the system and his assistance with the computerized skeleton plot generation algorithms. I would like to thank James Engle for his tips on graphical user interface design, Jim Burns and Gary Funkhouser for testing the software, and Ramzi Touchan for his recommendations on how to make the system more useful for dendrochronologists.

This work was supported by the National Science Foundation, *Grant SBR9601867*, and the University of Arizona, Office of the Vice President for Research.

DEDICATION

I dedicate this thesis to two wonderful people in my life, my parents, whose love, support, and encouragement have made this possible.

TABLE OF CONTENTS

LIST OF TABLES	7
LIST OF FIGURES.....	8
ABSTRACT	10
CHAPTER 1 INTRODUCTION	11
1.1 PROBLEM STATEMENT	11
1.2 OVERVIEW OF DENDROCHRONOLOGY	11
1.2.1 The Douglass Method of Cross-Dating.....	13
1.2.2 Review of Earlier Image Analysis Systems	15
1.3 OVERVIEW OF THE TREES SYSTEM.....	15
1.3.1 System Hardware	16
1.3.2 Software Architecture	17
1.3.3 Sample Analysis.....	18
1.3.4 Features	19
1.4 OVERVIEW OF THESIS	21
CHAPTER 2 TREES SYSTEM VERSION 1.0	22
2.1 REQUIREMENTS FOR TREE RING ANALYSIS.....	22
2.2 REVIEW OF EDGE DETECTION ALGORITHMS	23
2.2.1 Canny Edge Detector	24
2.2.2 Modified Canny Edge Detector.....	25
2.3 ALGORITHM DESCRIPTION	26
2.3.1 Outline of Edge Detection Algorithm	26
2.3.2 Gradient Computation.....	26
2.3.3 Double threshold Linking.....	29
2.3.4 Final Linking Stage	31
2.4 LIMITATIONS OF TREES v1.0.....	33
2.4.1 Ring Identification.....	33
2.4.1.1 Edge Fragmentation.....	34
2.4.1.2 Cross-connected Rings.....	35
2.4.2 Disk Usage	37
2.4.3 Wood-Centered Analysis	37
CHAPTER 3 TREES SYSTEM VERSION 2.0	38
3.1 OVERVIEW	38
3.2 EDGE DIRECTION	41

TABLE OF CONTENTS - Continued

3.2.1	Local Dominant Ring Direction.....	42
3.2.2	Global Dominant Ring Direction.....	46
3.2.3	Consistency Check for Local Ring Directions.....	48
3.3	RING IDENTIFICATION BY EDGE TRACKING.....	50
3.3.1	Isolating Flow Patterns.....	50
3.3.2	Edge Tracking.....	52
3.4	TEST RESULTS.....	54
CHAPTER 4 EDGE MAPS.....		57
4.1	MEMORY AND DISK REQUIREMENTS.....	57
4.2	CHAIN CODES.....	58
4.2.1	Data Structure.....	59
4.2.2	File Format.....	61
4.3	OPERATIONS ON CHAIN CODES.....	62
4.3.1	Down Sampling Edge Maps.....	62
4.3.2	Width Measurement from Chain Codes.....	65
4.4	AVERAGE GRAYSCALE PROFILE.....	70
4.4.1	Extracting Missed-Ring Boundaries.....	71
4.4.2	Extracting Earlywood-Latewood boundaries.....	76
CHAPTER 5 ANALYST INTERACTION.....		79
5.1	REGION-OF-INTEREST SELECTION AND PROCESSING.....	79
5.1.1	Processing Modes.....	80
5.1.2	Data Structures for Boundary Marking.....	82
5.2	RING BOUNDARY DETECTION.....	84
5.3	WOOD-CENTERED ANALYSIS.....	87
CHAPTER 6 SUMMARY.....		93
6.1	SUMMARY.....	93
6.2	CONCLUSIONS.....	95
6.3	SCOPE FOR FUTURE CONTRIBUTIONS.....	95
REFERENCES.....		97

LIST OF TABLES

TABLE 4-1. Elements of the chain code data structure.....	60
---	----

LIST OF FIGURES

Figure 1-1. Transverse view of a tree sample.....	12
Figure 1-2. Growth ring width representation.....	14
Figure 1-3. TREES system hardware.	17
Figure 1-4. Software architecture in the TREES system.....	18
Figure 1-5. TREES data processing flowchart.....	20
Figure 2-1. Cross-section of a tree sample with resin ducts.....	23
Figure 2-2. Edges found by the Canny algorithm for the image in Figure 2-1.	25
Figure 2-3. Image analysis flowchart of the TREES system v1.0.....	26
Figure 2-4. Sobel edge masks used to approximate $\frac{\partial}{\partial y}$ and $\frac{\partial}{\partial x}$	27
Figure 2-5. Histogram of gradient magnitude for the image shown in Figure 2-1.....	29
Figure 2-6. Double threshold linking in TREES system v1.0.....	31
Figure 2-7. Fragment linking in the TREES system v1.0.	32
Figure 2-8. Gradual shift in the ring orientation within a Juniper sample.	33
Figure 2-9. Ring fragmentation caused by prominent rays.	34
Figure 2-10. Cross-connections introduced by double threshold linking.	36
Figure 2-11. Wild fragment connection produced by final linking process.....	36
Figure 3-1. Ring boundary characteristics.	39
Figure 3-2. Image analysis flowchart of the TREES system v2.0.	41
Figure 3-3. Gradient direction image for the sample shown in Figure 3-1.....	42
Figure 3-4. Distribution of gradient direction in region R_2	45
Figure 3-5. Distribution of gradient direction in region R_4	45
Figure 3-6. Local gradient direction determined for various regions in the image.....	46
Figure 3-7. Dominant gradient and ring directions.	47
Figure 3-8. Global distribution of gradient direction.	48
Figure 3-9. Gradient magnitude image for the sample in Figure 3-1.....	49
Figure 3-10. Flow patterns isolated from gradient magnitude image.	51
Figure 3-11. Flow patterns with discontinuities.....	52
Figure 3-12. Search operator used for edge tracking.	53
Figure 3-13. Ring detection in the presence of resin ducts.	55
Figure 3-14. Ring detection in the presence of prominent rays.	55

LIST OF FIGURES - Continued

Figure 3-15. Ring detection in the presence of narrow rings.	56
Figure 3-16. Ring detection in the presence of low-contrast narrow rings.	56
Figure 4-1. Edge pixels in a raster image and the chain code operator.	58
Figure 4-2. Components of the chain code data structure.	59
Figure 4-3. Sample chain code structure.	61
Figure 4-4. Binary file format for archiving chain codes.	62
Figure 4-5. Sample tree-ring image at full resolution.	63
Figure 4-6. Edge map for the sample in Figure 4-5.	63
Figure 4-7. Down-sampled image and edge map for the sample in Figure 4-5.	64
Figure 4-8. Edge map obtained by down sampling along the tree-ring direction.	65
Figure 4-9. Width lines overlaid on a sample edge map.	69
Figure 4-10. Average grayscale profile for a sample cross-section.	71
Figure 4-11. Average grayscale profile and its first derivative.	73
Figure 4-12. Cross-section of a Juniper sample with narrow rings.	75
Figure 4-13. Missed rings extracted from average grayscale profile.	76
Figure 4-14. Earlywood-latewood width measurement from average grayscale profile. .	78
Figure 5-1. Single frame captured from a core sample.	80
Figure 5-2. Region-of-interest selection.	81
Figure 5-3. Pseudo-code for a typical raster scan.	82
Figure 5-4. Data structure for boundary marking.	83
Figure 5-5. Pseudo-code for scanning an irregular region.	83
Figure 5-6. Pseudo-code for convolution on an irregular region.	84
Figure 5-7. Noise features obscuring ring boundaries.	85
Figure 5-8. Gradient magnitude image for the sample in Figure 5-7.	85
Figure 5-9. A partly detected tree-ring in the zoom window.	86
Figure 5-10. Fully detected tree-rings in the zoom window.	86
Figure 5-11. TREES graphical interface with control points marked on the browse image.	89
Figure 5-12. Stage Motion graphical interface with control points marked on the live video image.	90

ABSTRACT

Edge detection forms the key component in an image analysis system for application in dendrochronology. The nature and diversity of tree-ring patterns make it necessary to employ human intervention and intelligence in tree-ring detection. Numerous edge detection algorithms have been developed for various applications, but not all of them are amenable to analyst interaction. This thesis describes the design and implementation of an interactive image analysis system for dendrochronology. The primary focus is on the development of a flow-based edge detection algorithm that exploits the edge characteristics in a tree-ring analysis framework. Other issues addressed in the thesis are region-of-interest processing, interactive edge linking, tree-ring boundary extraction from averaged grayscale profile, and tree-ring width measurement from chain-code representation.

Chapter 1 INTRODUCTION

1.1 Problem Statement

Edge detection forms the key component in a computer vision-based tree-ring analysis system for use in dendrochronology. The nature and diversity of tree-ring patterns make it necessary to employ human intervention and intelligence in tree-ring boundary identification. Any attempt to completely automate the process would mandate unacceptable levels of computation and could cause the system to become unpredictable when handling tree-ring samples with anomalies. A number of edge detection algorithms have been developed for various applications, but not all of them are amenable to analyst interaction. The work presented in this thesis is a part of a research project being carried out jointly by the Digital Image Analysis Laboratory of the Electrical and Computer Engineering Department and the Laboratory of Tree-Ring Research at the University of Arizona. This research was supported by the National Science Foundation, *Grant SBR9601867*, and the University of Arizona, Office of the Vice President for Research.

1.2 Overview of Dendrochronology

Dendrochronology is the science of dating events and variations in environment in former periods by comparative study of growth rings in trees and aged wood. Each year a tree adds a layer of wood to its trunk and branches, thus creating the annual rings that are seen when viewing a cross-section. Earlywood develops during the rainy season when there is abundant water for cellular growth. Latewood develops during the drier season

and consists of densely packed cells that cause the ring to appear darker compared to the earlywood. Thus a typical cross-section of a tree trunk exhibits alternating annular regions of earlywood and latewood. The transverse view of a tree ring sample is shown in Figure 1-1. The growth ring is the annular region consisting of the earlywood and the subsequent latewood, moving away from the wood center (*pith*). Transitions from the latewood to an adjoining outer earlywood mark the boundaries between growth rings.

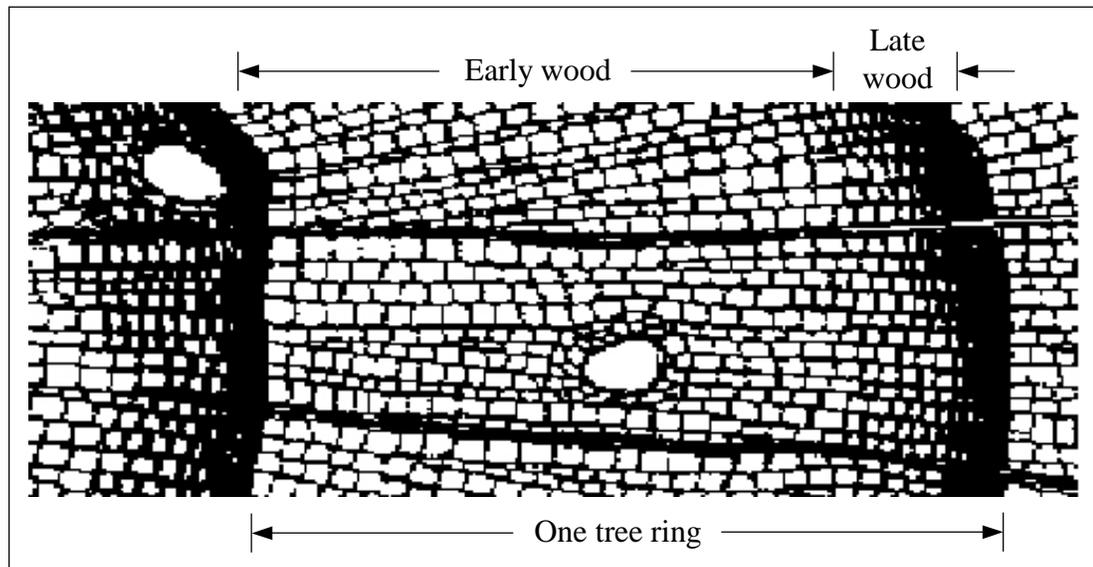


Figure 1-1. Transverse view of a tree sample.

(Courtesy of the Laboratory of Tree-Ring Research, University of Arizona,
<http://www.ltrr.arizona.edu/dendrochronology.html>)

The age of a tree is estimated by counting the number of growth rings from the pith to the bark. The outermost growth ring at the bark corresponds to the current year in the growth of a living tree or the last year in the span of a dead tree. The widths of growth rings vary across species, across geographical region and from year to year. Samples from trees of the same species in a geographical region exhibit similar patterns of

variation in the width of growth rings. This is because the local climatic fluctuation tends to be fairly consistent over a given geographical region. These climatic changes or patterns in specific geographical regions can be traced by the study of old living trees. The samples taken from trees of unknown growth period can then be studied for matches with the samples from trees of known age and growth period. Using this process, known as *cross-dating*, when the widths of growth rings from different samples match or are found to be overlapping in age, a master chronology is built for the geographical region.

1.2.1 The Douglass Method of Cross-Dating

There are two popular techniques for cross-dating that vary in the method of representation used for the widths of growth rings. These representations are *time-series plots* and *skeleton plots*. Figure 1-2 shows an example of a time-series and a skeleton plot. A time-series plot is a graphical representation of the sequence of ring widths obtained from tree-ring width measurement. Methods of cross-dating based on time-series plots usually employ a combination of computational and visual correlation techniques. A skeleton plot is a discrete-time graphical representation of the strength of outliers in the sequence of ring widths obtained from tree-ring width measurement. A ring is considered to be an outlier if its width is significantly different from that of the adjacent rings. Vertical bars mark the locations of outliers in the plot, and the length of the bar itself represents the outlier magnitude. Skeleton plots enable samples to be matched based on the similarity in ring width variation rather than the absolute ring widths. In practice, additional ring attributes are symbolically marked on the plot and play a crucial

role in the cross-dating process. These attributes, which represent features such as micro rings, false rings, frost rings, missing rings, etc., could be used to visually validate a match. This technique of dating tree-ring samples is known as the Douglass method[5].

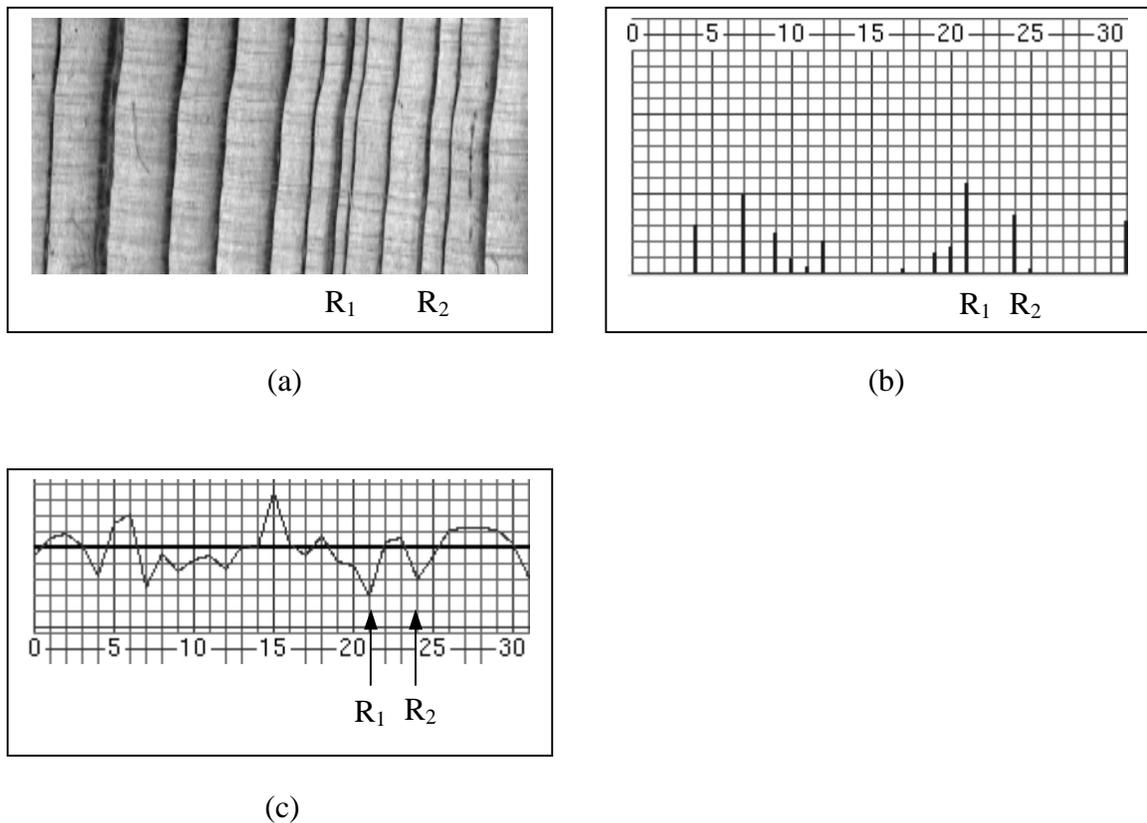


Figure 1-2. Growth ring width representation.

(a) Cross-section of a tree-ring sample marked with two outliers R_1 and R_2 . (b) Skeleton plot for the sample in (a) shown for a 31 year span. (c) Time-series plot for the sample in (a) shown for a 31 year span (Individual ring points are connected for display).

A successful match depends on the reliability of ring width measurements. The absence of width measurement for one or more rings in a sample offsets the corresponding skeleton plot and could render it undateable even though the samples have a chronological overlap. To resolve such issues, the Douglass method of cross-dating

makes it imperative for the wood sample to be available during the entire matching process. Termed *wood-centered analysis*, this is the distinguishing feature of the Douglass method of cross-dating.

1.2.2 Review of Earlier Image Analysis Systems

Image analysis techniques have been applied earlier to the problem of ring boundary detection in dendrochronology. Some of the commercially available software systems include DENDRO[7],[8],[16], and MacDRUID[20]. Reflected-Light Image Analysis system, developed at the University of Arizona, is another software system that estimates the ring density by a careful calibration of the light reflected from the sample[17],[18],[19]. The three systems primarily differ in the type and amount of image data that is analyzed. Absent in these systems is the ability to return to the wood sample that is critical to the Douglass method of cross-dating. The following section provides an overview of the TREES system and its features that make it suitable for the Douglass method of dendrochronology.

1.3 Overview of the TREES System

TREES is a computer-assisted tree-ring analysis system developed specifically to aid in the Douglass method of cross-dating. It uses a mosaic format to store the images captured from the wood to limit the amount of memory required to store and process a sample. The dependency of the TREES system on the hardware used for image capture limits the system from processing frames captured outside the TREES environment.

1.3.1 System Hardware

A schematic diagram of the hardware setup is shown in Figure 1-3. The image capture device in the TREES system is a Kodak Megaplug 1.4i CCD monochrome digital camera capable of resolutions up to 1317x1035 pixels. The resolution of each frame captured by the TREES system is 1280x1024 pixels and the mosaic is composed from many such frames. The camera is coupled to the eyepiece of a Nikon SMZ-U microscope for imaging a cross section of the sample mounted on a positioning table under the microscope. The microscope magnification is set to 1.1X during any given sample acquisition. In order to increase the field of view of the captured frames, a video coupler lens with magnification set to 0.63X is used. This optical setup results in a combined magnification of 0.693X and a diagonal field of view of approximately 16.4mm for each frame. The resolution obtained from the combination is approximately 10 μ m, which is sufficient to resolve most small tree rings. A fiber-optic illuminator of approximately 75mm in diameter is mounted on the objective lens of the microscope to provide the necessary illumination. Two stepper motors drive the positioning table, one in each of the X and Y directions. The stepper motors are under computer control and are capable of moving the positioning table by a span of up to 75cm in each direction, with a resolution of about 1mm. A third stepper motor, coupled to the focus knob of the microscope, is controlled by the TREES system for automatic focus adjustments. The entire hardware setup is under the control of a Sun Ultra-1 workstation running the Solaris operating system with 256 MB of RAM and a 4 MB video frame buffer.

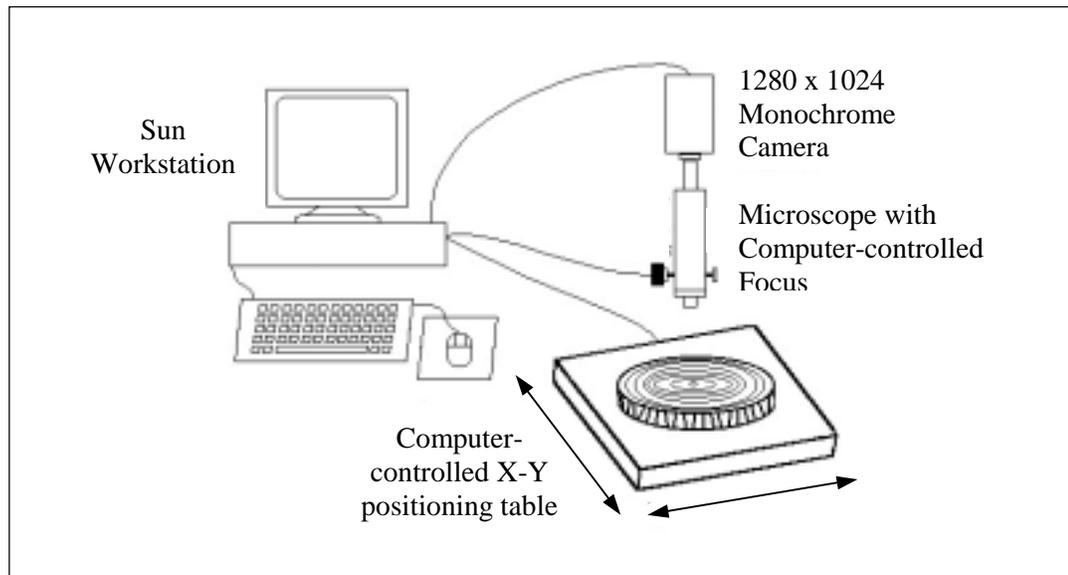


Figure 1-3. TREES system hardware.

1.3.2 Software Architecture

The software architecture of the TREES system is shown in Figure 1-4. The topmost layer in the TREES system is the Tcl/Tk layer that creates the graphical user interface and handles all the user interaction. A few of the Tcl/Tk scripts used are modifications of the ones presented in [15] and [22]. Custom Tcl/Tk commands required to carry out image analysis tasks are defined in the command layer. The command layer serves to translate custom commands to the actual C code, in the algorithm layer, that implements them. Although most of the computation is done in the algorithm layer, a small percentage of it is handled in the command layer. The *SADIE* (System at Arizona for Digital Image Experimentation) library, developed at the Digital Image Analysis Laboratory, University of Arizona, is a set of image processing software routines written

in the C language. The algorithm layer interfaces extensively with the SADIE library to make use of the provided functions. The low-level routines to interface with the hardware are written in C++. The Tcl scripts communicate with the camera and positioning table through routines in the hardware interface layer.

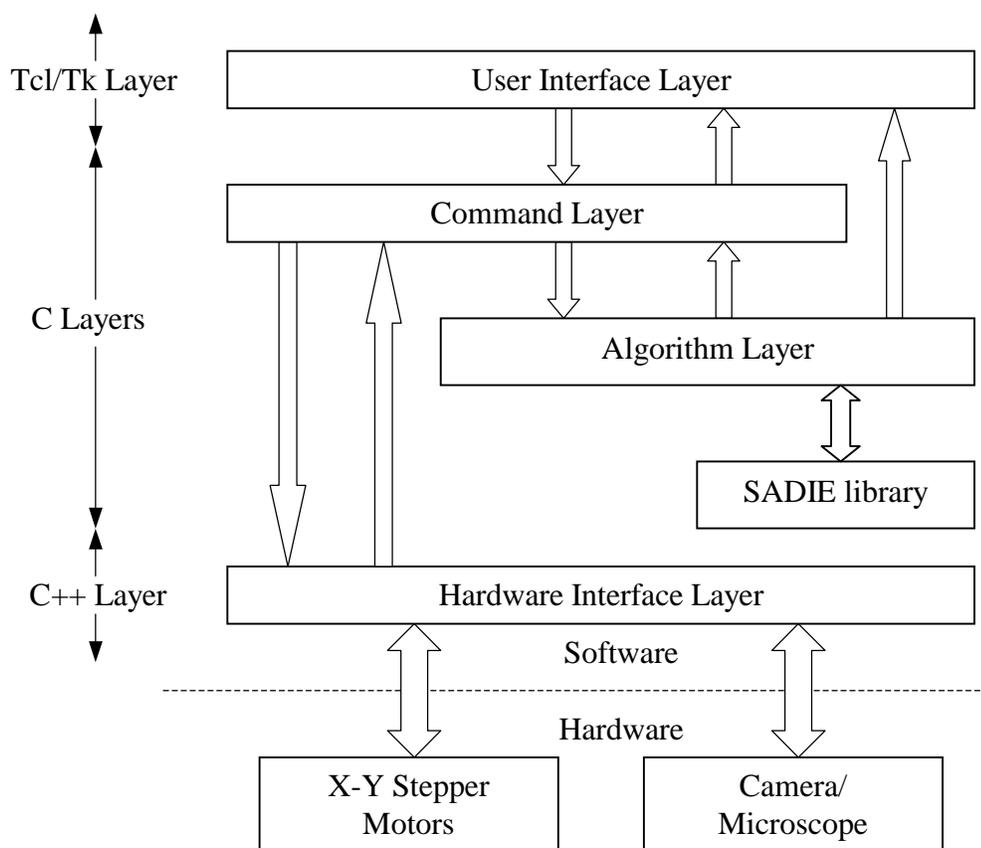


Figure 1-4. Software architecture in the TREES system.

1.3.3 Sample Analysis

A flowchart of the processing steps in the TREES system is shown in Figure 1-5. The following steps outline the procedure involved in analyzing a sample with the TREES system:

- The sample to be analyzed is mounted on the X-Y positioning table.
- The analyst is prompted to select a cross-section within the sample to be dated.
- The system computes the number of frames required to cover the cross-section.
- Automatic focusing is done prior to the capture of each frame[4].
- Gain and bias adjustments are performed on each frame to ensure a smooth gray level transition across frame boundaries in the mosaic[4].
- The frames are combined after spatial registration to form a mosaic image.
- Computer vision algorithms are used to detect rings in the mosaic image.
- Tree-ring widths are then measured to generate the skeleton plot.
- Skeleton plot data from the samples can be exported to the *CROSSDATE* program, which assists in cross-dating and master chronology composition[6].

1.3.4 Features

The following is a list of the TREES system features that make it suitable for the Douglass method of cross-dating:

- The TREES system allows the analyst to select a region-of-interest within the wood sample. This enables the analyst to visually inspect the sample and select a region free of wood anomalies.
- The amount of data analyzed by the TREES system is significantly greater than that analyzed by other systems. The individual width measurements are averaged over the region-of-interest, resulting in better accuracy.

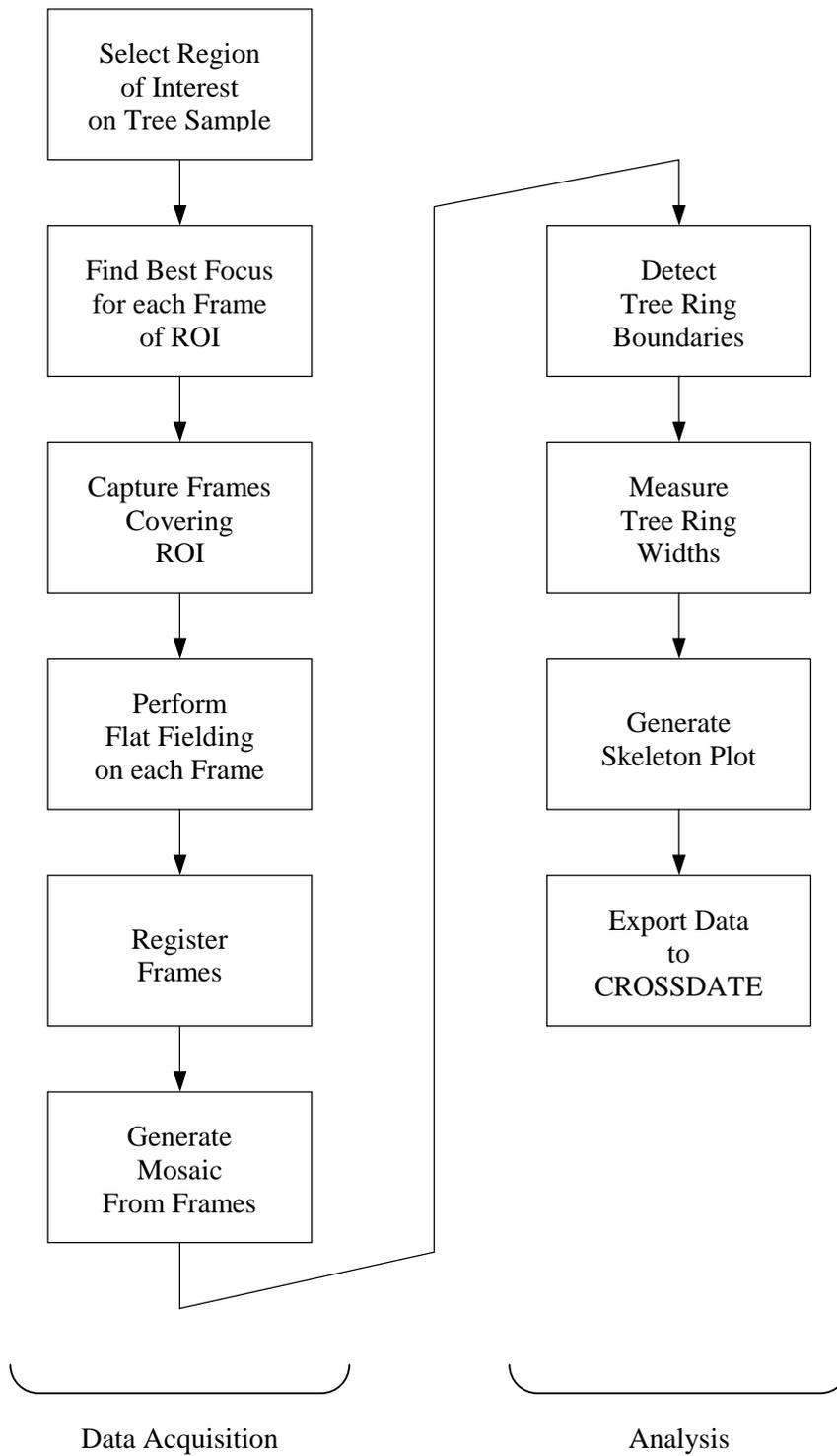


Figure 1-5. TREES data processing flowchart.

- The TREES system exports width measurements in a format suitable for skeleton plots, which form the basis for the Douglass method of cross-dating.

1.4 Overview of Thesis

This thesis describes the design and implementation of an interactive image analysis system for dendrochronology based on an edge detection algorithm that adapts itself to changing tree-ring orientation. This Chapter provided an introduction to dendrochronology and an overview of the TREES system. Chapter 2 begins with a description of TREES system v1.0 , the previous version of the TREES system. The new techniques and algorithms presented in this thesis have been tested as replacements, and some as enhancements, to the algorithms in the TREES system v1.0 and constitute TREES system v2.0, the current version of the TREES system described in Chapter 3. The data structure design and methods for archival, editing, and processing of edge maps are discussed in Chapter 4. A scheme for measuring tree-ring widths from the chain codes will be presented at the end of Chapter 4. Chapter 5 describes analyst interaction during various stages of the tree ring identification process and a technique for wood-centered analysis in the TREES system. Finally, the work presented in the thesis is summarized in Chapter 6.

Chapter 2 TREES SYSTEM VERSION 1.0

The main objective of the tree ring analysis system is to compute ring widths from the edge map. The edge map represents transitions from one ring to another, or more specifically, transitions from the latewood of one ring to the earlywood of an adjacent ring, moving away from the pith. Earlywood and latewood can be distinguished from one another by the gray scale intensity in the captured image. Latewood to earlywood transitions, representing ring boundaries, can be identified by abrupt changes in gray scale intensity. In an image analysis system, these are found by the process of edge detection. The following section will discuss the requirements of edge detection for tree ring analysis.

2.1 Requirements for Tree Ring Analysis

A primary requirement of edge detection for tree ring analysis is that the rings must be fully connected. Ring identification should not result in rings that are spatially broken. This plays an important role in the ring width measurement process. The problem posed by rings that are not fully connected is that fragments of the same ring appear as distinct rings to the algorithms that process the edge maps. Although methods exist for linking fragments, and are implemented in TREES system v1.0, it would be desirable to avoid fragmentation in the first place. Moreover, a mosaic formed from 25 frames could reach a size of 21,760x17,408 pixels, assuming one-third overlap between frames, and

connected-components labeling on the mosaic image during edge linking would require an unacceptable amount of memory for storing intermediate labels.

The second requirement is that the edges representing ring boundaries must be singly connected. This requirement arises from the fact that the edge maps are to be chain coded before archiving on the disk. Edges that are not singly connected cause ambiguity during chain coding. A final, but important, requirement is that the edge detection algorithm must lend itself to analyst interaction. The presence of anomalies, such as resin ducts, shown in Figure 2-1, branch scars, cracks, prominent rays, etc. would mandate analyst intervention to guide in ring boundary identification. For this purpose, algorithms that operate on a ring-by-ring basis are more suitable than those that operate on a pixel-by-pixel basis.

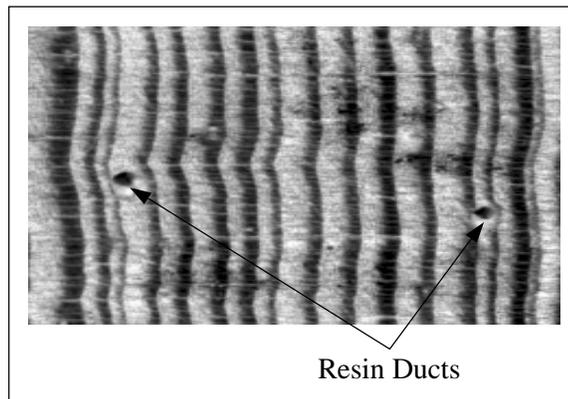


Figure 2-1. Cross-section of a tree sample with resin ducts.

2.2 Review of Edge Detection Algorithms

The TREES system v1.0 has features that are beneficial in the Douglass method of cross-dating. However, the edge detection algorithms used in this initial version of the TREES

system have limitations. This section presents an overview of the computer vision techniques employed in the TREES system v1.0. The focus will be on the edge detection algorithm for ring boundary identification.

2.2.1 Canny Edge Detector

The Canny algorithm[1] has been applied to the problem of edge detection in various images. The primary advantage of this algorithm over the other edge detection techniques is that it is guaranteed to produce fully connected, one pixel wide boundaries, which is a requirement in the tree-ring analysis framework. The first step in implementing the Canny edge detector is to smooth the image to suppress noise edges. A Gaussian filter with a standard deviation of 3 pixels has been found to be appropriate for most tree ring samples [2]. The gradient magnitude and direction are computed at each pixel in the resulting smoothed image by applying a gradient operator, such as the Sobel edge mask. Non-maxima suppression in the Canny edge detector uses both the gradient magnitude and direction to suppress pixels along the gradient direction at each pixel. An implementation of non-maxima suppression is described in [10]. The gradient direction at each pixel is quantized into one of four sectors, and pixels that have a lower magnitude than their two directional-neighbors are suppressed. This guarantees the edges to be one pixel wide and fully connected.

As evident from the non-maxima suppression, the algorithm relies on local pixel data to detect edges. Therefore, a straightforward implementation of the Canny edge detection algorithm for tree-ring analysis has the following drawbacks (Figure 2-2):

- [1] Wood features like prominent rays that run perpendicular to the ring boundaries, cause connections between detected tree-ring edges.
- [2] Intra-ring earlywood-latewood boundaries are also detected.

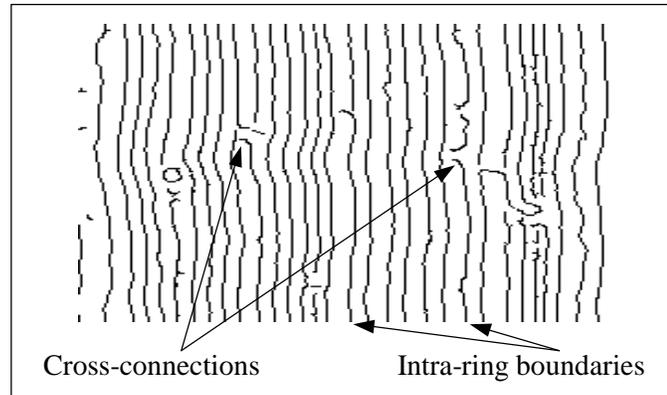


Figure 2-2. Edges found by the Canny algorithm for the image in Figure 2-1.

2.2.2 Modified Canny Edge Detector

The drawbacks of the Canny algorithm operating in a local scope are overcome by the modified Canny edge detection algorithm [2]. The modified Canny edge detection algorithm, employed in the TREES system v1.0, differs from the original Canny algorithm in the way non-maxima suppression is implemented. *If the orientation of tree rings within a region of interest were fairly constant*, then non-maxima suppression in a direction perpendicular to the ring direction would suppress the edges along prominent rays. This forms the basis for the modified Canny edge detection algorithm. The second drawback of the Canny algorithm is overcome by discarding pixels where the gradient direction is opposite to that of the *assumed* latewood-earlywood gradient direction. A

detailed discussion of the modified Canny edge detection algorithm as implemented in the TREES system v1.0 is provided in the following section.

2.3 Algorithm Description

2.3.1 Outline of Edge Detection Algorithm

The image analysis flowchart in the TREES system v1.0 is shown in Figure 2-3. This is an expanded view of the first analysis block in the TREES data processing flowchart shown in Figure 1-5.

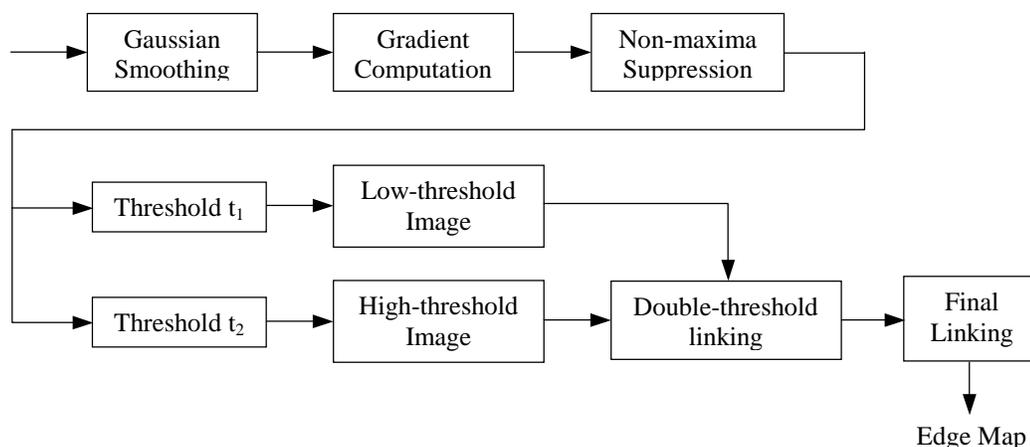


Figure 2-3. Image analysis flowchart of the TREES system v1.0.

2.3.2 Gradient Computation

An edge in an image is a boundary or contour where there is appreciable variation in some physical aspect of the image that reflects as a change in intensity, color, or texture. Given that the ring boundaries are transitions from latewood to earlywood, edges of interest in the image are identified by abrupt changes in grayscale intensity values. The edge location is determined by computing the first derivative of the grayscale intensity

image. The generalization of first derivative to a two-dimensional intensity image, $f(x, y)$, is the gradient $\nabla f(x, y)$ defined as

$$\nabla f(x, y) = \frac{\partial f(x, y)}{\partial x} \hat{i}_x + \frac{\partial f(x, y)}{\partial y} \hat{i}_y,$$

where \hat{i}_x is the unit vector in the x-direction and \hat{i}_y is the unit vector in the y-direction[13]. The gradient magnitude is computed by a discrete approximation to

$$|\nabla f(x, y)| = \sqrt{\left(\frac{\partial f(x, y)}{\partial x}\right)^2 + \left(\frac{\partial f(x, y)}{\partial y}\right)^2}.$$

A method developed by Sobel[13] computes the approximation to $|\nabla f(x, y)|$ as

$$|\nabla f(n_1, n_2)| \rightarrow \sqrt{f_x^2(n_1, n_2) + f_y^2(n_1, n_2)},$$

where

$$f_x(n_1, n_2) = f(n_1, n_2) * h_x(n_1, n_2),$$

$$f_y(n_1, n_2) = f(n_1, n_2) * h_y(n_1, n_2),$$

with $h_x(n_1, n_2)$ and $h_y(n_1, n_2)$ as shown in Figure 2-4.

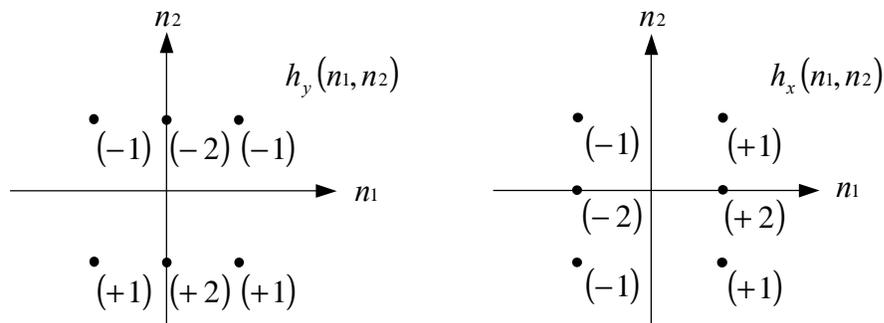


Figure 2-4. Sobel edge masks used to approximate $\frac{\partial}{\partial y}$ and $\frac{\partial}{\partial x}$.

Ring boundary identification in the TREES system v1.0 is based on the modified Canny edge detection algorithm. The algorithm requires that the image be smoothed prior to gradient computation. A 11 x 11 Gaussian smoothing filter, $g(x, y)$, with a standard deviation of 3 pixels has been found to be suitable for this purpose[2]. The gradient magnitude, $|\nabla s(x, y)|$, is then computed for the resulting smoothed image given by

$$s(x, y) = f(x, y) * g(x, y).$$

Since the rings are assumed to be vertical within each frame, directional non-maxima suppression evaluates to comparing the gradient magnitude of each pixel with that of its left and right neighbors and retaining only those pixels that have a gradient magnitude greater than that of both the neighbors. The image obtained from directional non-maxima suppression is

$$p(x, y) = \begin{cases} |\nabla s(x, y)| & \text{if } |\nabla s(x, y)| > |\nabla s(x-1, y)| \text{ and } |\nabla s(x, y)| > |\nabla s(x+1, y)|, \\ 0 & \text{else.} \end{cases}$$

In order to suppress earlywood-latewood transitions, pixels in $p(x, y)$ that have a left neighbor of higher grayscale intensity than the right neighbor in the smoothed image, $s(x, y)$, are reset to zero,

$$q(x, y) = \begin{cases} p(x, y) & \text{if } s(x-1, y) < s(x+1, y), \\ 0 & \text{else.} \end{cases}$$

2.3.3 Double threshold Linking

The resulting image consists of noise edges and ring fragments in addition to the fully-connected rings. This poses the problem of having to connect the ring fragments while discarding the noise edges. To solve this, TREES v1.0 employs a double-threshold approach. An example histogram of the gradient magnitude image of a tree sample is shown in Figure 2-5.

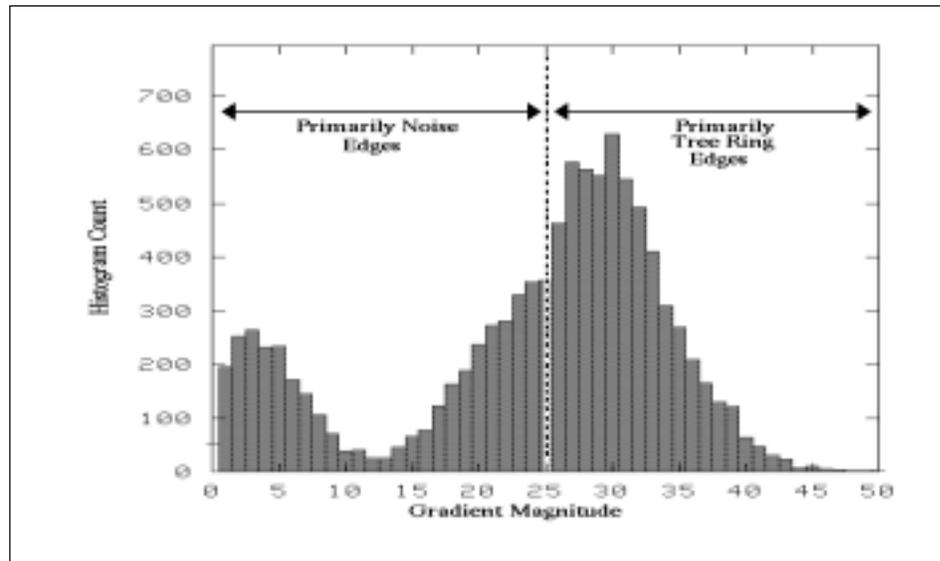


Figure 2-5. Histogram of gradient magnitude for the image shown in Figure 2-1.

A low threshold image, $l(x, y)$, is obtained from $q(x, y)$ by applying a threshold t_L that is sufficiently small to retain all the ring fragments,

$$l(x, y) = \begin{cases} 1 & \text{if } q(x, y) > t_L, \\ 0 & \text{else.} \end{cases}$$

Owing to the low threshold, high contrast noise edges would be present in $l(x, y)$ as shown in Figure 2-6 (b). The high threshold image, $h(x, y)$, is obtained from $q(x, y)$ by applying a threshold t_H that is sufficiently large to suppress the noise edges,

$$h(x, y) = \begin{cases} 1 & \text{if } q(x, y) > t_H, \\ 0 & \text{else.} \end{cases}$$

In effect, the high threshold image consists primarily of edges that represent tree rings. Fragments in the high threshold image that are smaller than a predefined size are discarded, and the size-filtered image is as shown in Figure 2-6 (c). Since the two thresholded images have pixels in common between them, retaining only those pixels that are absent in the high threshold image forms a mutually exclusive version of the low threshold image,

$$l_e(x, y) = \begin{cases} l(x, y) & \text{if } h(x, y) = 0, \\ 0 & \text{else.} \end{cases}$$

Double-threshold linking is accomplished by retaining all the fragments from the high threshold image $h(x, y)$, but only those from the low threshold image $l_e(x, y)$ that are in a τ -pixel neighborhood of the high threshold fragments. In effect, this approach serves to bridge the gaps between high contrast edge fragments of a ring with the low contrast edge fragments belonging to the same ring, as shown in Figure 2-6 (d). The value, τ , that defines the neighborhood is kept small to avoid connecting the fragments of adjacent rings together.

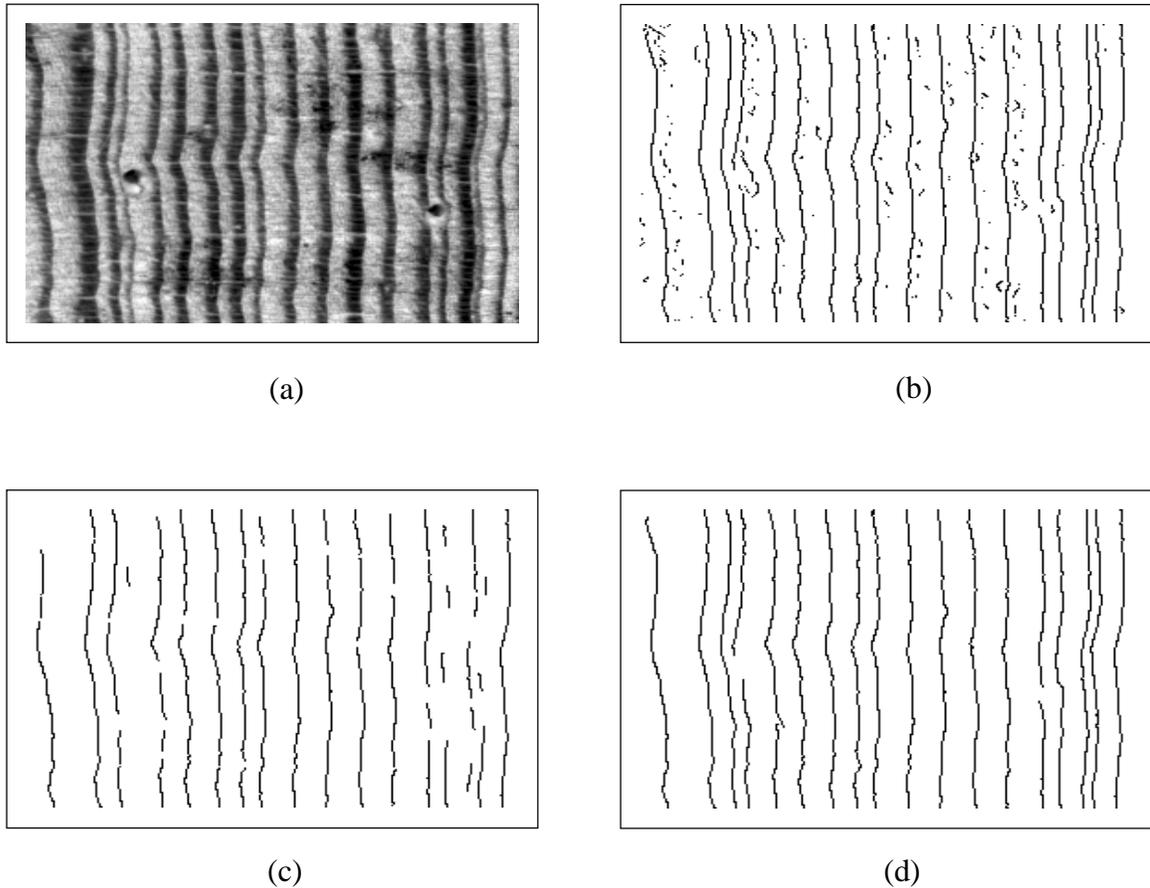


Figure 2-6. Double threshold linking in TREES system v1.0.

(a) Cross-section of a tree sample with resin ducts. (b) Low threshold image of the gradient magnitude. (c) High threshold image of the gradient magnitude. (d) Double threshold linked rings.

2.3.4 Final Linking Stage

Due to the constraint imposed by the τ value on the neighborhood, some ring fragments still remain disconnected after double-threshold linking. A second and final stage of linking is used to connect ring fragments that occur between fully connected rings. The width of a ring relative to that of an adjacent ring usually remains fairly constant along its length due to the physical characteristics of growth rings. Therefore, fragments of a

broken ring would occur at the same fraction of width between the two enclosing fully connected rings, as shown in Figure 2-7. The second linking stage is based on this idea.

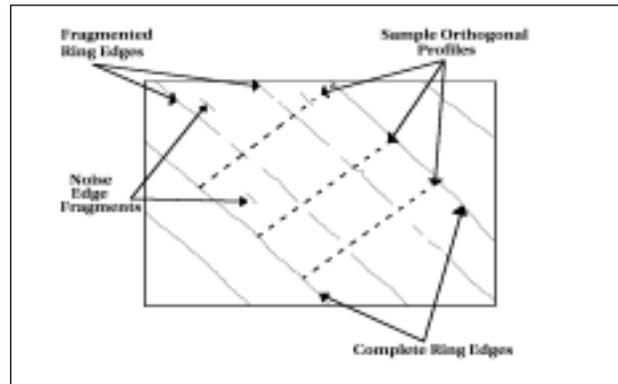


Figure 2-7. Fragment linking in the TREES system v1.0.

The number of broken rings, n , between two complete rings is estimated to be the most frequent fragment count that occurs at orthogonal profiles between the fully connected rings. To determine their positions, a histogram of the number of fragments that occur at each fraction of width, quantized into M bins between 0 and 100%, along the orthogonal profiles is computed. The n largest bins in the histogram indicate the position of the broken rings relative to the width between the two enclosing fully connected rings, as shown in Figure 2-7. Fragments that occur at these positions are assigned labels to associate them with the broken rings. The remaining fragments are discarded as noise edges. Straight lines are then synthesized to connect the endpoints of fragments with the same label.

2.4 Limitations of TREES v1.0

2.4.1 Ring Identification

In TREES v1.0, the computer vision algorithms used in the ring identification process are based on the assumption that *the tree rings are almost vertical and that they extend from the frame top to its bottom*. Owing to the high magnification at which the frames are captured, this assumption usually holds for well-behaved samples. However, there exist samples that exhibit appreciable variation in the ring orientation within the mosaic image as shown in Figure 2-8. This would cause the edge detection and linking algorithms in TREES v1.0 to fail. Moreover, the system does not allow for analyst interaction during the ring identification process to resolve problems that the system may not be able to handle on its own.

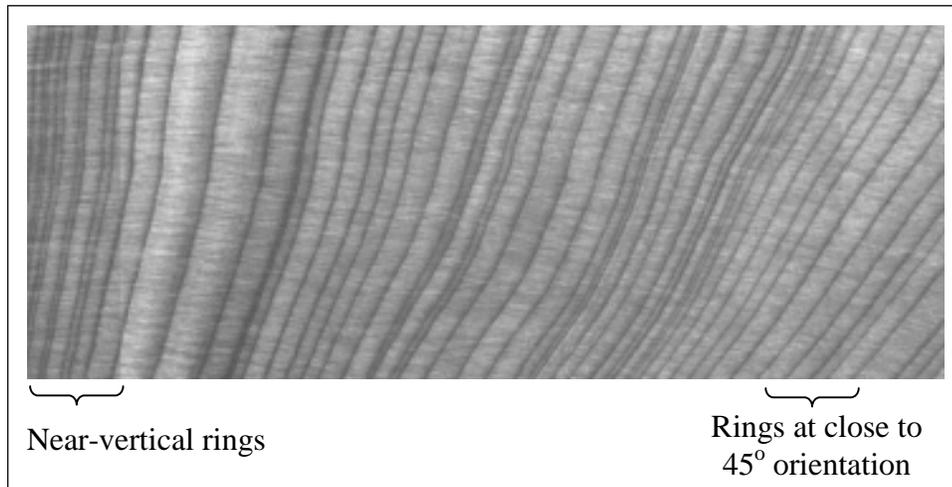


Figure 2-8. Gradual shift in the ring orientation within a Juniper sample.

2.4.1.1 Edge Fragmentation

Edge fragmentation in the TREES system is a result of the modification to the Canny algorithm. Non-maxima suppression in the Canny algorithm relies on local gradient direction and assures one pixel wide, fully connected rings. Directional non-maxima suppression implemented in the modified Canny edge detector improves noise filtering but also fragments the rings in the presence of features like resin ducts and prominent rays. This is illustrated in Figure 2-6 (d). Note how the change in gradient direction because of the resin duct causes directional non-maxima suppression to fragment the ring. The Canny algorithm would have followed the gradient contour around the resin duct thus producing a false protrusion in the detected ring, as shown in Figure 2-2.

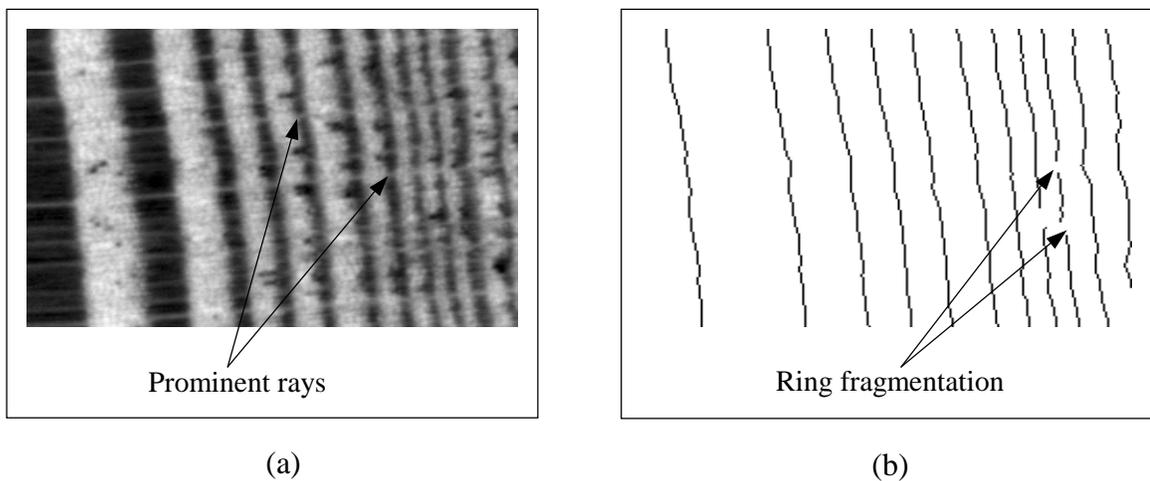


Figure 2-9. Ring fragmentation caused by prominent rays.

Figure 2-9 illustrates ring fragmentation in the presence of a prominent ray. Recall that the Canny algorithm tends to follow the gradient along prominent rays and cross-connects with the adjacent rings (Figure 2-2). The modified Canny algorithm suppresses

pixels along the noise edge gradient thus breaking continuity in the detected ring. A two-stage edge linking process in the TREES system v1.0 attempts to link the fragments.

2.4.1.2 Cross-connected Rings

Cross-connection occurs as a result of linking fragments of different rings together. The high threshold applied to suppress noise edges in the double-threshold approach breaks the ring at places of low contrast. The linking phase attempts to connect the ring fragments by searching for low contrast ring fragments in a τ -pixel neighborhood of high contrast edge fragments. τ defines the radius of a circle centered at the end point of a high contrast fragment. In case multiple low contrast fragments are encountered in the circular search region, the algorithm picks the nearest fragment. Since the algorithm does not incorporate the gradient direction at the point of discontinuity, there is a tendency to link with the fragment of an adjacent ring that happens to be in the search region as shown in Figure 2-10. The τ chosen is a compromise between the number of missed rings and the number of cross-connections tolerable. In the TREES system v1.0, the search radius is kept relatively small to minimize cross-connections. This, however, degrades the edge linking performance and necessitates a second linking stage to ensure fully connected rings.

In the second linking stage, the locations of broken rings are determined and fragments that occur at these locations are assigned a unique label to associate them with the broken ring. In regions of narrow rings obscured by resin ducts, fragmentation is severe and the algorithm occasionally assigns the same label to fragments of two adjacent

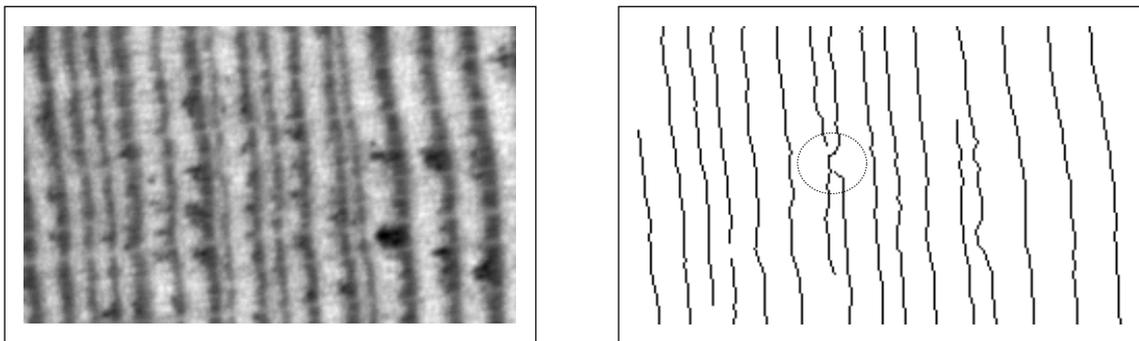


Figure 2-10. Cross-connections introduced by double threshold linking.

rings due to its inability to distinguish between them. An example of wild fragment connection is shown in Figure 2-11. This causes wild fragment linking between fragments of different rings. In addition, the second stage of linking requires fully connected rings to be present on both sides of a fragmented ring for it to be able to connect the fragments. Therefore, ring fragments that lie outside the outermost connected rings cannot be linked together.

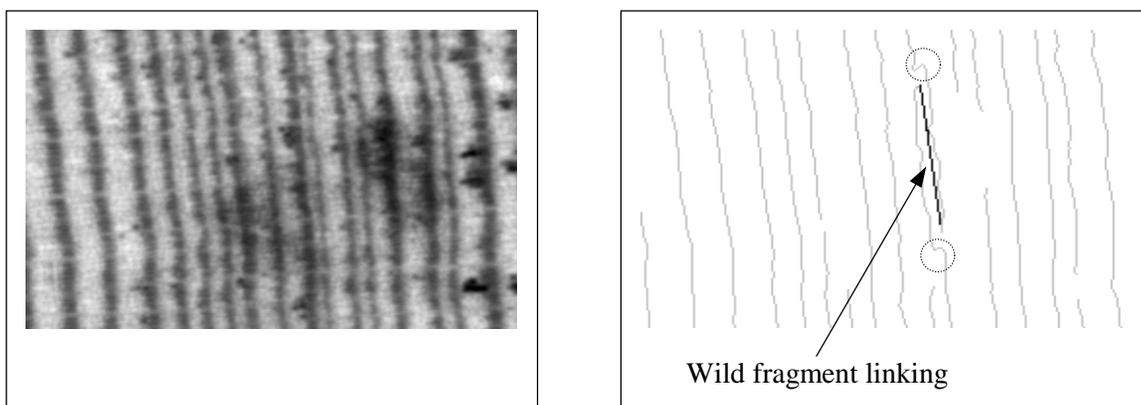


Figure 2-11. Wild fragment connection produced by final linking process.

Note: Dotted circles highlight cross-connections due to double threshold linking.

2.4.2 Disk Usage

Edge maps obtained during the ring identification process are stored as image files in the TREES system v1.0. Saving this image on the disk at 8 bits/pixel requires approximately 1.25MB of disk space per frame. For a sample with 15 frames, this would translate to over 18MB of disk space. As more and more samples are archived, the disk space requirements grow rapidly. Also, this format of storing edge maps does not lend itself well to ring boundary editing involving insertion and deletion of rings from the edge map.

2.4.3 Wood-Centered Analysis

The Douglass method of cross-dating requires that the wood be available for reference during the *entire* process of dating. The ability to go back to the wood poses two requirements. First, there needs to be a stage co-ordinate system with an origin that remains constant across samples. And second, the mapping between stage and image co-ordinates has to be recorded for each sample to be dated. Due to hardware limitations, the stage coordinates are reset each time the stage is initialized. Stage re-initialization affects the coordinate mapping established during a previous session, thereby hindering our ability to return to the wood sample.

Chapter 3 TREES SYSTEM VERSION 2.0

3.1 Overview

The performance of edge detection in the TREES system v1.0 is effected by the shortcomings of fragment linking. Fragmentation occurs as a result of thresholding the gradient magnitude and necessitates linking to ensure fully connected rings. The objective of double-thresholding in the TREES system v1.0 is to suppress noise edges. In a tree ring analysis framework, ring boundaries form the edges of interest and wood anomalies such as resin ducts and prominent rays contribute to the noise edges. Tree-rings exhibit certain characteristics that distinguish them from other edges in an image. First, in most samples, there is very little change in the boundary profile from one ring to another (Figure 3-1), i.e., adjacent rings are almost parallel. Second, the change in ring direction in a given cross-section is gradual. Tree-ring detection in the TREES system v2.0 is based on these two properties.

Kasaei et al., present a method for fingerprint feature extraction by considering fingerprints as sample images from non-stationary processes with flow patterns[11]. In this method ridge areas are treated as foreground and noise regions as background. The foreground-background segmentation is based on the assumption that clear regions flow in a particular direction while noise regions have no dominant direction. They present a block-direction approach, using 16x16 blocks, for determining the dominant ridge direction. A similar technique is used in the TREES system v2.0 for ring detection.

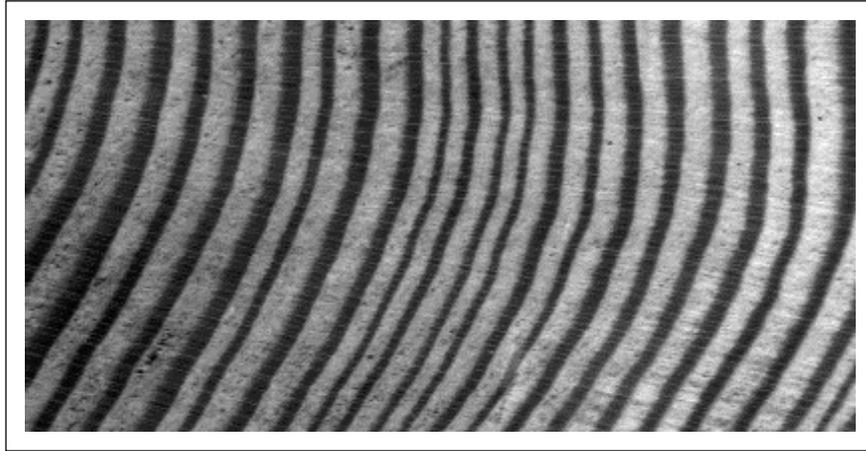


Figure 3-1. Ring boundary characteristics.

The gradient is computed as in the modified Canny algorithm and the *global dominant ring direction (GDRD)* is determined. In addition, since the tree-ring profiles vary from region to region within a sample, the *local dominant ring direction (LDRD)* is determined for each region in the image. The presence of high contrast noise edges would corrupt the *LDRD* determined for a region. In order to identify these regions, the *LDRD* of each region is compared with the *GDRD*. When the *LDRD* in a region deviates from the *GDRD* in excess of a pre-defined threshold, a majority vote is taken among the neighboring regions. If there is a clear majority, the region under consideration is assigned the winning *LDRD*, else the region is assigned the average *LDRD* computed from its compliant neighbors. Once the local dominant ring direction is resolved for all regions in the image, the gradient direction at each pixel is compared with the local gradient direction (perpendicular to *LDRD*) of the region that contains the pixel. An algorithm that operates at a pixel level employing visual tests is described in [21]. Pixels where the gradient direction deviates from the local gradient direction are reset in the

gradient magnitude image to prevent them from contributing to an edge. In practice, a certain degree of flexibility is allowed to account for any variation in the ring direction within the region. Eliminating pixels this way results in a gradient magnitude image that consists of narrow bands of pixels, or ridge areas, around ring boundaries. The precise edge location could then be tracked by following the path of highest gradient within the ridge areas. A technique that incorporates test for consistency is described in [14]. Thus, the likely problem of ring fragmentation in curved sections of the ring is overcome by incorporating the local ring direction. In addition, since edges are identified on a ring-by-ring basis, this approach lends itself well to analyst interaction.

The image analysis flowchart of the TREES system v2.0 is shown in Figure 3-2. The algorithm development is based on the edge characteristics in a tree ring analysis framework. The image is divided into non-overlapping regions and the gradient direction histogram is computed for each individual region. The local dominant ring direction is determined from the observed histogram probabilities for each region. The global dominant ring direction is determined by merging the local histograms. After resolving the *LDRDs*, pixels that disagree with the local edge direction are eliminated. An edge tracking approach similar to ridge detection is then employed to track the ring boundaries. In view of the processing time constraints imposed by the high resolution of images, this approach is computationally efficient, allowing the edges to be chain-coded on the fly to eliminate the need for labeling and, thereby, easing the requirement on memory resources.

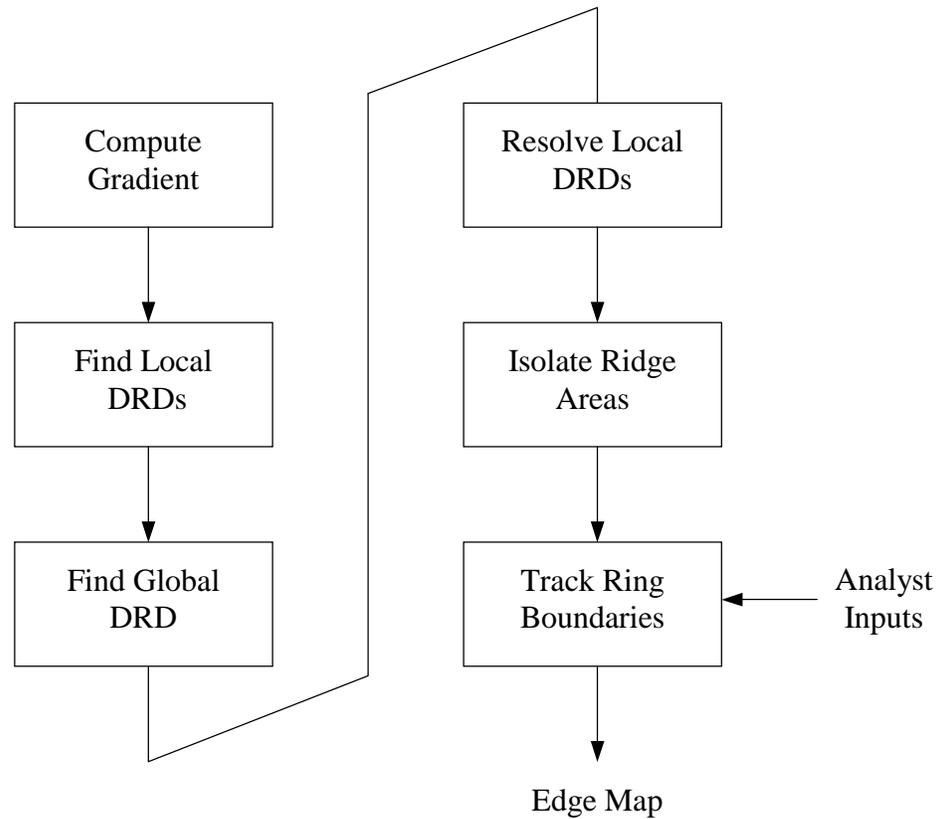


Figure 3-2. Image analysis flowchart of the TREES system v2.0.

3.2 Edge Direction

The Canny algorithm performs non-maxima suppression based on the gradient direction at each pixel. In effect, it relies on the gradient direction in a highly-localized region. This is a pixel-level estimate of the edge direction and often reflects the direction of non-ring features running across ring boundaries. The gradient direction image for the sample in Figure 3-1 is shown in Figure 3-3. On the other hand, the modified Canny algorithm uses global ring orientation alone to implement directional non-maxima suppression. The global ring orientation is a coarse estimate of the ring direction within the image. The improved algorithm incorporates the following in identifying tree-rings in a region:

1. Global orientation of tree rings in the image as a whole ($GDRD$).
2. Local tree-ring orientation in the region ($LDRD_i$).
3. Orientation of tree rings in the neighboring regions ($\{LDRD_j \mid j \in N_g(i)\}$).
4. Gradient direction at each pixel in the region ($\theta(r, c)$).

A discussion of the technique employed in the TREES system v2.0 to determine global and local dominant ring directions is presented in this section.

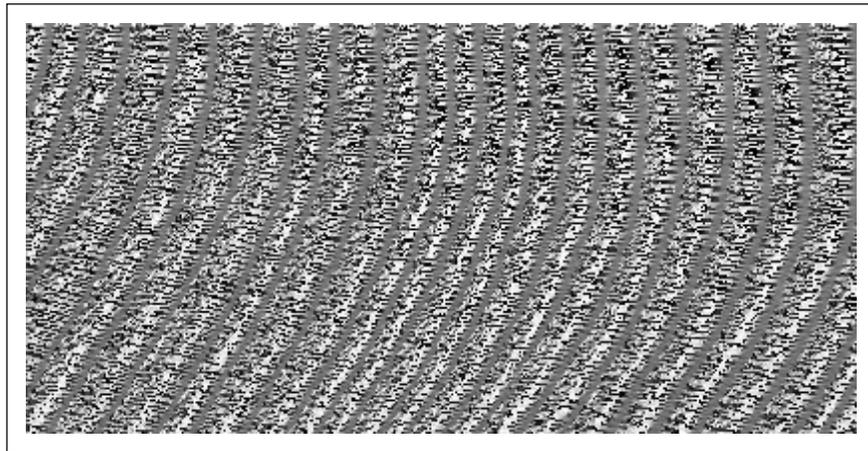


Figure 3-3. Gradient direction image for the sample shown in Figure 3-1.

3.2.1 Local Dominant Ring Direction

The gradient direction, $\theta(r, c)$, and magnitude, $G(r, c)$, are computed at each pixel in the image as in the Canny edge detection algorithm. The gradient direction values are quantized into m bins in the range $[-\pi, +\pi]$, resulting in a quantization step of

$$\Delta = \frac{2\pi}{m} \text{ radians}$$

The gradient magnitude and direction images are divided into non-overlapping regions such that

$$\bigcup_i R_i = R,$$

where R_i is the region of support (ROS) for region i , and R is the region of support for the whole image. The region of support, R_i , could be selected depending on the spacing between ring boundaries in the image and the ring curvature. However, in the TREES system v2.0, the ROS is a $n \times n$ square, with a constant n both within an image sample and across samples. The n value used in the tests is 64 pixels, i.e. the ROS is a 64x64 pixel square. This value has been found to be suitable for a wide range of samples. The grid overlay on the sample is shown in Figure 3-6. Once the image is partitioned, histograms of observed gray levels in the gradient magnitude image are determined for all regions in the image as:

$$H_i^G(j) = \#\{(r, c) \in R_i \mid G(r, c) = j\},$$

where $H_i^G(j)$ represents the histogram count for gray level j in region i . The histogram probabilities of the observed gray levels are computed as:

$$P_i^G(j) = \frac{H_i^G(j)}{\#\{(r, c) \mid (r, c) \in R_i\}},$$

where $P_i^G(j)$ represents the histogram probability of gray level j in region i . From the regional gradient magnitude distributions, the q^{th} percentile, $\pi_i[q]$, is determined for each region as:

$$\pi_i[q] = \min \left\{ j \mid \sum_{k=1}^j P_i^G(k) \geq \frac{q}{100} \right\}.$$

Using the value of $\pi_i[q]$ for each region, the histogram probabilities of gray levels in the gradient direction image are computed, taking into account only those pixels that have a gradient magnitude equal to or greater than $\pi_i[q]$. This ensures that only the pixels that constitute high-contrast edges contribute to the gradient direction histogram, and hence the role of any low-contrast noise edges is reduced. In the TREES system v2.0, the $\pi_i[q]$ value chosen corresponds to the upper quartile, i.e. $q = 75$. The histogram of observed gray levels in the gradient direction image for region i is given by:

$$H_i^\theta(j) = \#\{(r, c) \in R_i \mid G(r, c) \geq \pi_i[75] \text{ and } \theta(r, c) = j\}.$$

The corresponding histogram probabilities for region i are computed as:

$$P_i^\theta(j) = \frac{H_i^\theta(j)}{\#\{(r, c) \in R_i \mid G(r, c) \geq \pi_i[75]\}}.$$

The largest bin in P_i^θ corresponds to the dominant gradient direction in region i ($LDGD_i$), i.e.

$$P_i^\theta(LDGD_i) = \max_j \{P_i^\theta(j)\}.$$

Assuming that *the local dominant gradient direction is an indication of the local ring direction*, the local dominant ring direction for region i is determined as:

$$LDRD_i = LDGD_i \pm \frac{\pi}{2} \text{ radians.}$$

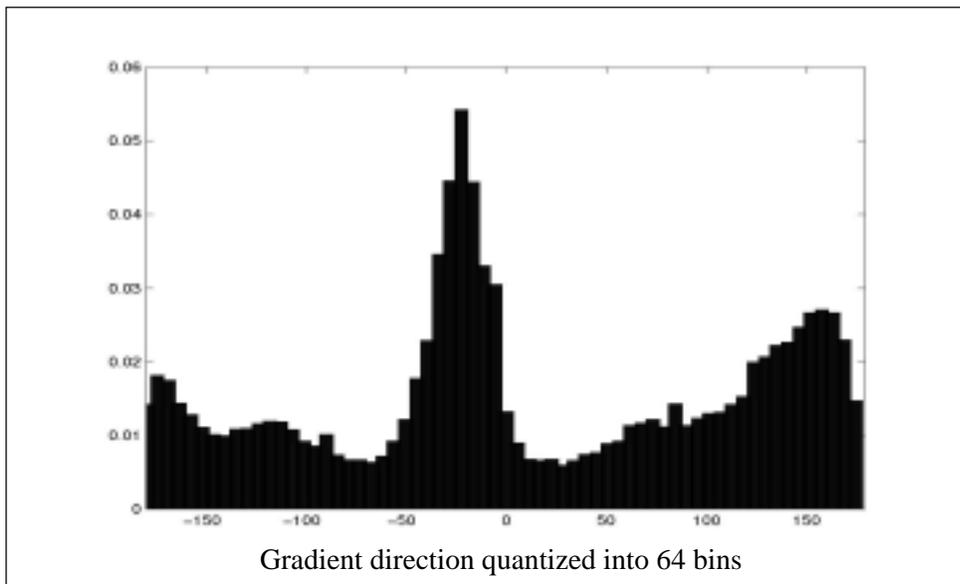


Figure 3-4. Distribution of gradient direction in region R_2 .

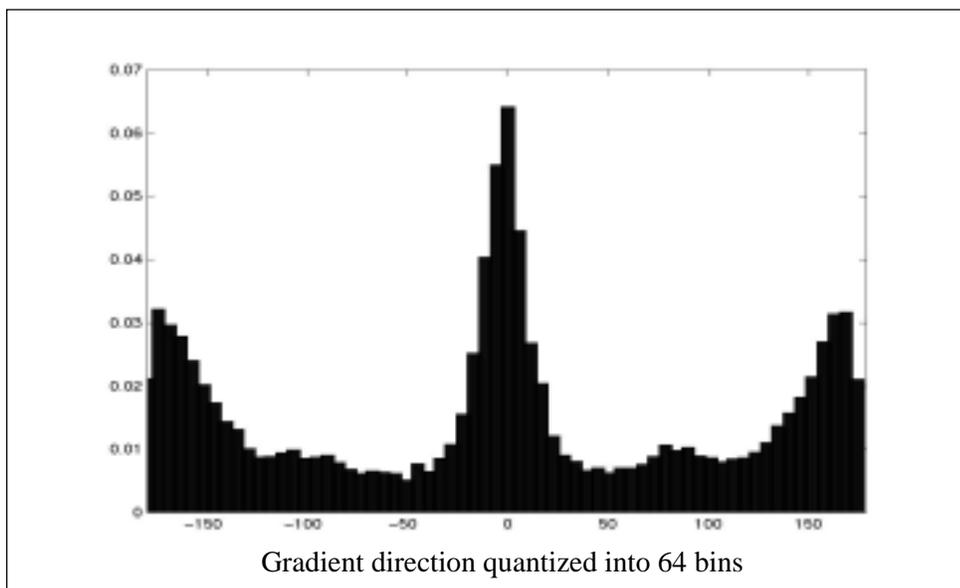


Figure 3-5. Distribution of gradient direction in region R_4 .

The gradient direction distributions for two of the regions are shown in Figure 3-4 and Figure 3-5. Figure 3-6 shows the image obtained from a tree ring sample and the local gradient direction determined for various regions in the image.

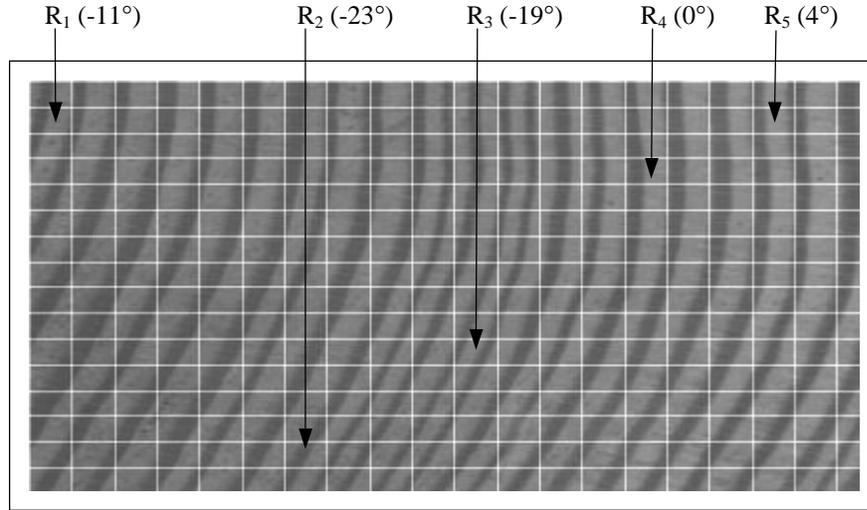


Figure 3-6. Local gradient direction determined for various regions in the image.

3.2.2 Global Dominant Ring Direction

The global dominant gradient direction (*GDRD*) is determined by combining the local magnitude histograms. The global histogram of observed gray levels in the gradient magnitude image is computed as:

$$H^G(j) = \sum_i H_i^G(j).$$

The histogram probabilities of observed gray levels are computed as:

$$P^G(j) = \frac{H^G(j)}{\#\{(r,c) \in R\}}.$$

From the global magnitude distribution, the upper quartile, $\pi[75]$, is determined as:

$$\pi[q] = \min \left\{ j \mid \sum_{k=1}^j P^G(k) \geq \frac{q}{100} \right\},$$

with $q = 75$. Similar to the technique for determining local ring direction, the upper quartile reduces the effect of pixels constituting low-contrast noise edges. Once the upper quartile is found, the histogram of observed gray levels in the gradient direction image is computed by counting pixels where the gradient magnitude exceeds $\pi[75]$.

$$H^\theta(j) = \#\{(r, c) \in R \mid G(r, c) \geq \pi[75] \text{ and } \theta(r, c) = j\}.$$

The global histogram probabilities of gradient direction are computed as:

$$P^\theta(j) = \frac{H^\theta(j)}{\#\{(r, c) \in R \mid G(r, c) \geq \pi[75]\}}.$$

The largest bin in P^θ is an indication of the global dominant gradient direction (Figure 3-8). Assuming that *the global dominant gradient direction is an indication of the global dominant ring direction*, the dominant ring direction (Figure 3-7) is determined as:

$$GDRD = GDGD \pm \frac{\pi}{2} \text{ radians.}$$

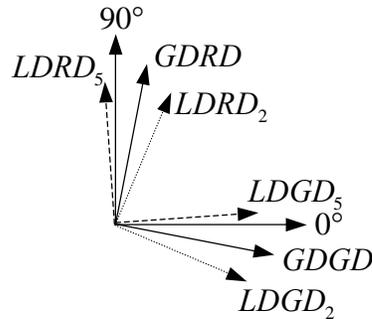


Figure 3-7. Dominant gradient and ring directions.

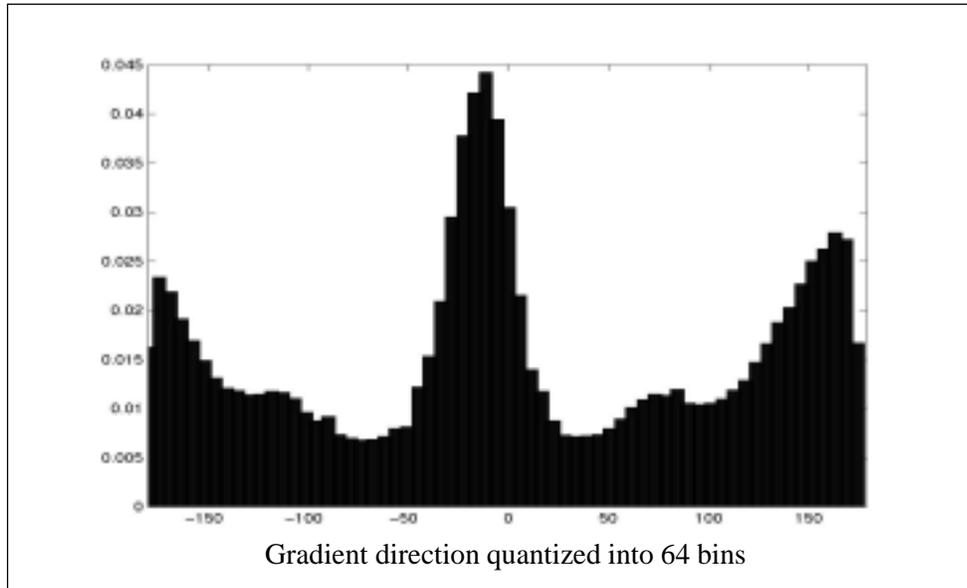


Figure 3-8. Global distribution of gradient direction.

3.2.3 Consistency Check for Local Ring Directions

Recall that the upper quartile was incorporated to prevent low-contrast noise edges from contributing to the dominant gradient direction. However, there are regions in the image where the ratio of the number of pixels constituting ring boundaries to the total number of pixels in the region is relatively small. There might also be regions that do not contain any boundary pixels. The direction histograms of these regions are noisy. The dominant gradient direction determined in these regions is an indication of the direction of noise edges that are present. For the purpose of isolating ridge areas (flow patterns) in the gradient magnitude image, shown in Figure 3-9, a region in which the dominant ring direction deviates from the global ring direction in excess of a threshold will be termed as

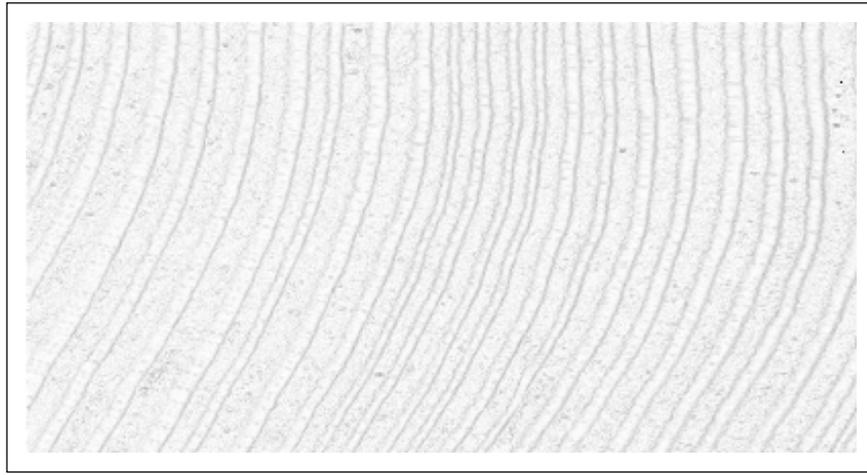


Figure 3-9. Gradient magnitude image for the sample in Figure 3-1

a *non-compliant region*. In order to effectively isolate the flow patterns without breaking them, it becomes necessary to resolve non-compliant regions. Retaining noise flow patterns would cause the edge-tracking algorithm to follow the noise edges and deviate from the actual ring boundary. A region is non-compliant if

$$|LDRD_i - GDRD| > \varepsilon .$$

The threshold (or tolerance), ε , could be selected based on the curvature of tree-rings in the cross-section. However, in the TREES system v2.0 the value of ε is chosen to be $\pi/12$ radians, which translates to a tolerance band of $\pi/6$ radians. The set of all non-compliant regions is given by:

$$NC = \{i \mid |LDRD_i - GDRD| > \varepsilon\}$$

To resolve non-compliant regions, a majority vote is taken among its compliant neighbors as:

$$MV_i(j) = \#\{k \mid k \in N_8^i \text{ and } |LDRD_k - GDRD| \leq \varepsilon \text{ and } LDRD_k = j\} \forall i \in NC,$$

where N_8^i is the set of 8-connected neighbors of region i . If there is an absolute majority, the non-compliant region under consideration is assigned the winning $LDRD$, i.e.

$$LDRD_i = k \mid MV_i(k) = \max_j(MV_i(j)).$$

In the absence of a clear majority, the region is assigned the average $LDRD$ of its compliant neighbors, i.e.

$$LDRD_i = \frac{\sum_{\{k \mid k \in N_8^i \text{ and } |LDRD_k - GDRD| \leq \varepsilon\}} LDRD_k}{\#\{k \mid k \in N_8^i \text{ and } |LDRD_k - GDRD| \leq \varepsilon\}}.$$

3.3 Ring Identification by Edge Tracking

3.3.1 Isolating Flow Patterns

Figure 3-9 shows the gradient magnitude image for the sample in Figure 3-1. Note the presence of narrow bands of pixels (flow patterns) that stand out distinctly around each ring boundary. In addition to these, earlywood-latewood boundaries and prominent rays are also present. For the edge-tracking algorithm to work, the flow patterns corresponding to latewood-earlywood boundaries have to be isolated from the rest of the image. The pixels in the narrow bands fall within a small tolerance region centered at the local dominant gradient direction. Thus by filtering out pixels that lie outside the tolerance band in each region, the narrow bands could be isolated.

The gradient direction, $\theta(r,c)$, at each pixel is checked to see if it lies within the tolerance band around the local gradient direction, $LDGD_i$. The flow pattern in the image is obtained by retaining the gradient magnitudes of pixels that lie in the tolerance band of each region, while discarding the rest of the pixels, i.e.

$$F(r,c) = \begin{cases} G(r,c) & \text{if } |LDGD_i - \theta(r,c)| \leq \frac{\epsilon}{2} \\ 0 & \text{otherwise} \end{cases}$$

where i is the region containing the pixel (r,c) . The threshold used is half of that used in identifying non-compliant regions. This follows from the fact that, in general, there is more variation in ring direction in the image as a whole than within a small region. The resulting ridge-isolated gradient magnitude image, $F(r,c)$, is shown in Figure 3-10. For comparison, the flow pattern isolated from the gradient magnitude image before resolving non-compliant regions is shown in Figure 3-11. Note the discontinuities in flow patterns in unresolved regions.

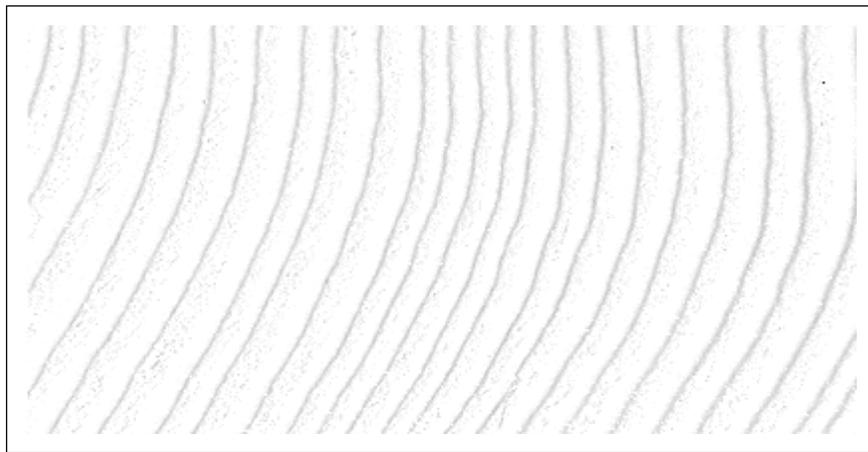


Figure 3-10. Flow patterns isolated from gradient magnitude image.

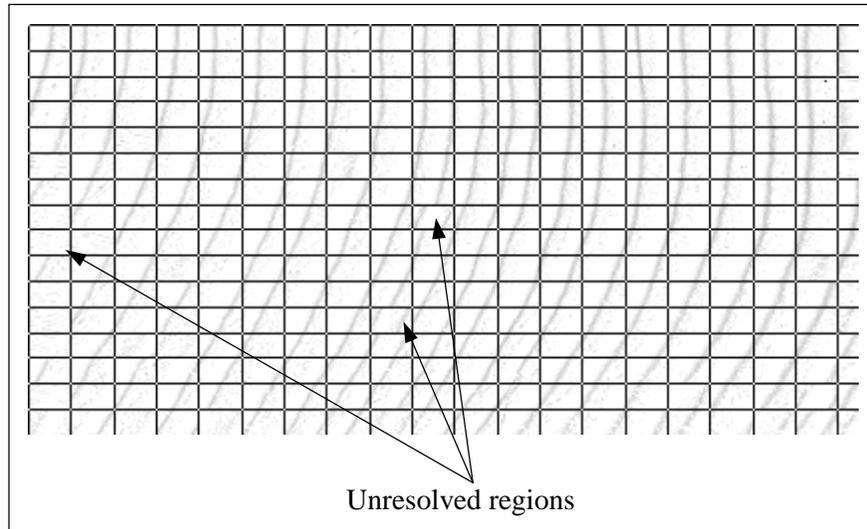


Figure 3-11. Flow patterns with discontinuities.

3.3.2 Edge Tracking

The exact location of ring boundaries within the ridge areas could be determined by following the path of highest gradient magnitude in the flow-pattern image, $F(r, c)$. The algorithm starts by horizontally scanning the image searching for non-zero pixels to identify the location of ridges. A pixel, p_1 , with the highest gradient magnitude, along the scan direction, before encountering a zero pixel is taken to be the starting point of a ring boundary. The test for consistency incorporated during flow-pattern isolation ensures that the gradient direction at p_1 , $\theta(p_1)$, is consistent with the local dominant gradient direction, i.e. p_1 is consistent since:

$$|\theta(p_1) - LDGD_i| \leq \frac{\varepsilon}{2},$$

where i is the region containing p_1 . With p_1 as the starting point, the next pixel in the ring being tracked is determined to be a consistent neighboring pixel with the highest gradient magnitude using the search operator shown in Figure 3-12.

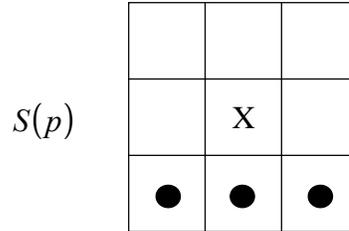


Figure 3-12. Search operator used for edge tracking.

This is a modification of the tracking algorithm presented in [14] since the cross-section of samples selected for analysis usually comprise rings that run top-down, rather than sideward. Under this assumption, it is reasonable to reduce the neighborhood operators used in [14] to $S(p)$. The next pixel is found as:

$$p_{i+1} = p_j \in S(p_i) \mid G(p_j) = \max_{\{k \mid p_k \in S(p_i)\}} (G(p_k)) \text{ and } |\theta(p_j) - \theta(p_i)| \leq \frac{\epsilon}{2},$$

where p_i is the previous pixel that was tracked on the tree-ring boundary. In certain regions of the image, wood features such as prominent rays run perpendicular to the ring boundaries. In such cases, the flow pattern breaks up at the point where the rays intersect with the ring boundary. Thus, at the point of break, the tracking algorithm fails to find any consistent neighbors. To handle these issues, the TREES system v2.0 has two modes of operation.

The first is an automatic, non-interactive mode. In this mode, the average gradient direction of a 20-pixel long segment of the ring preceding the point of break is computed. The algorithm proceeds by synthesizing a virtual straight-line segment in the direction of the ring segment at the point of break. The length of this virtual line segment is increased in steps of one pixel, until a limiting value is reached, and at each of the resulting end points of the virtual line, the neighborhood operator is applied to search for non-zero pixels. If a pixel is found within the threshold, the virtual line segment is used to connect the ring identified so far with the newly found pixel. From here on, the algorithm proceeds as before. If, however, the line segment reaches the limiting value and no pixels are found, then the algorithm discards the ring identified so far. In addition to the limiting value of the search region, the algorithm incorporates a limit on the number of breaks that a ring could have and still be connected using synthesized line segments. This is helpful if the algorithm happens to track a non-existent edge by tracking and connecting stray clusters of pixels. In such events, the limiting value forces the algorithm to discard the edge being tracked and proceed to the next ring boundary. Since a complete ring boundary is identified before proceeding to the next ring, this algorithm easily lends itself to chain coding on the fly. The second mode, involving analyst interaction is described in Chapter 5.

3.4 Test Results

In this section, a set of edge maps obtained using the improved algorithm on the samples used to discuss the limitations of the modified Canny edge detection algorithm will be presented. Figure 3-13(a) shows the image obtained from a tree sample consisting of

resin ducts. Recall that the Canny algorithm has a tendency to follow the edges around resin ducts and connect with the adjacent rings (Figure 2-2). The modified Canny algorithm causes the ring to fragment at the resin dust locations (Figure 2-6(d)). The ring boundaries identified by the new algorithm are shown in Figure 3-13(b).

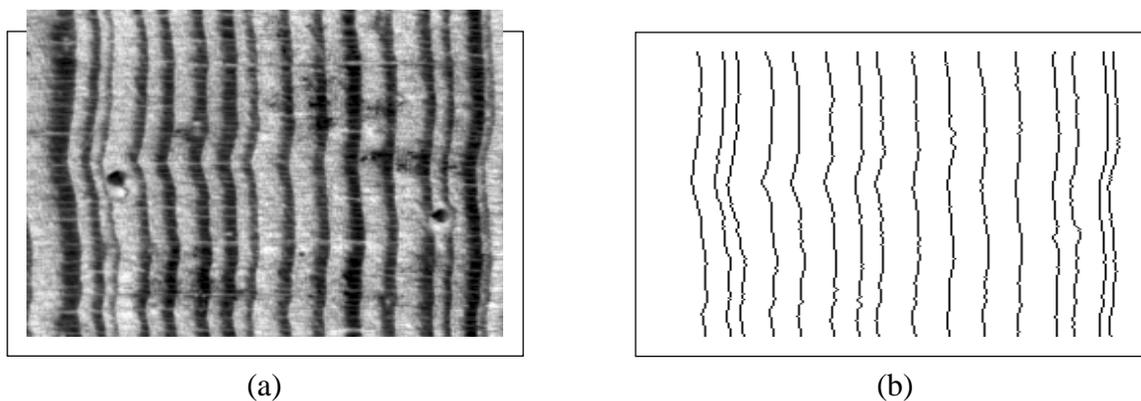


Figure 3-13. Ring detection in the presence of resin ducts.

Figure 3-14(a) shows the sample obtained from a tree sample with prominent rays. The Canny algorithm follows the gradient along prominent rays and cross-connects with adjacent rings. The modified Canny algorithm fragments the ring boundaries at points of intersection with the prominent rays (Figure 2-9(b)). The latewood-earlywood boundaries identified by the new algorithm are shown in Figure 3-14(b).

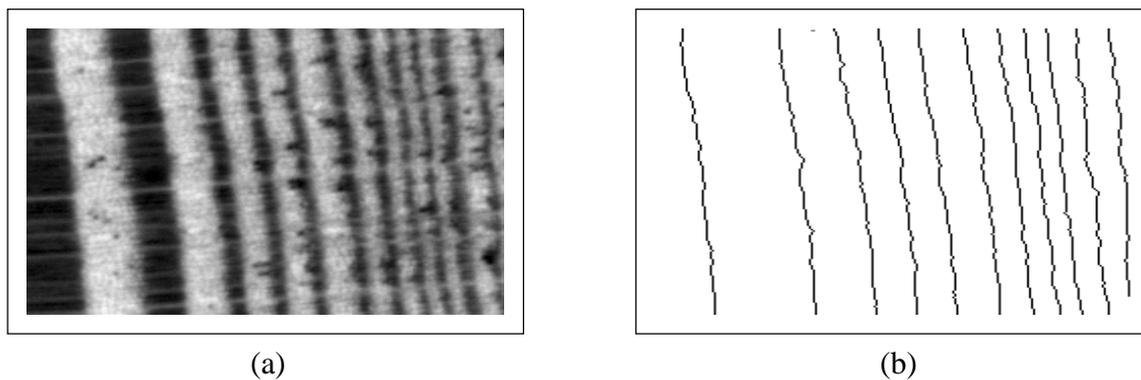


Figure 3-14. Ring detection in the presence of prominent rays.

Figure 3-15(a) shows the image obtained from a sample consisting of narrow rings. This image was used to illustrate the limitations of the double-threshold linking process (Figure 2-10) in the modified Canny algorithm. The tree-rings detected by the new algorithm are shown in Figure 3-15(b).

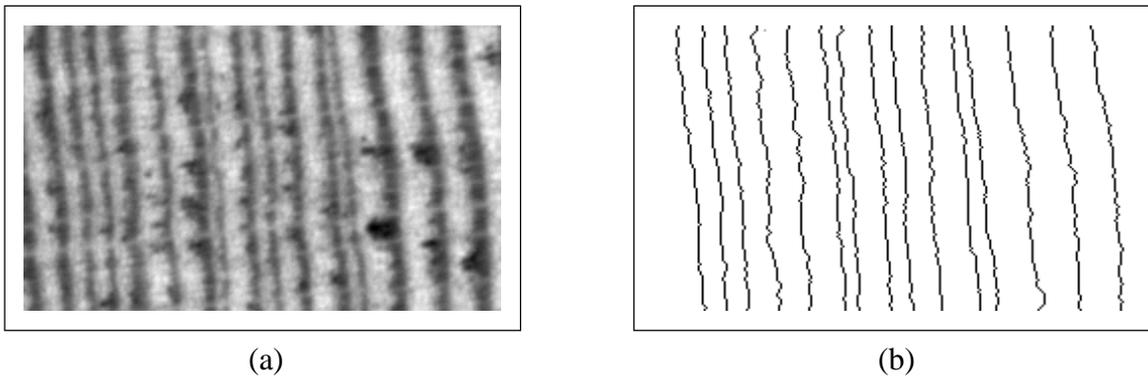


Figure 3-15. Ring detection in the presence of narrow rings.

Figure 3-16(a) shows the image used to illustrate the limitations in the final linking process of the modified Canny algorithm. Recall that the modified Canny algorithm performed a wild-fragment linking (Figure 2-11). The edge map obtained from the new algorithm is shown in Figure 3-16(b).

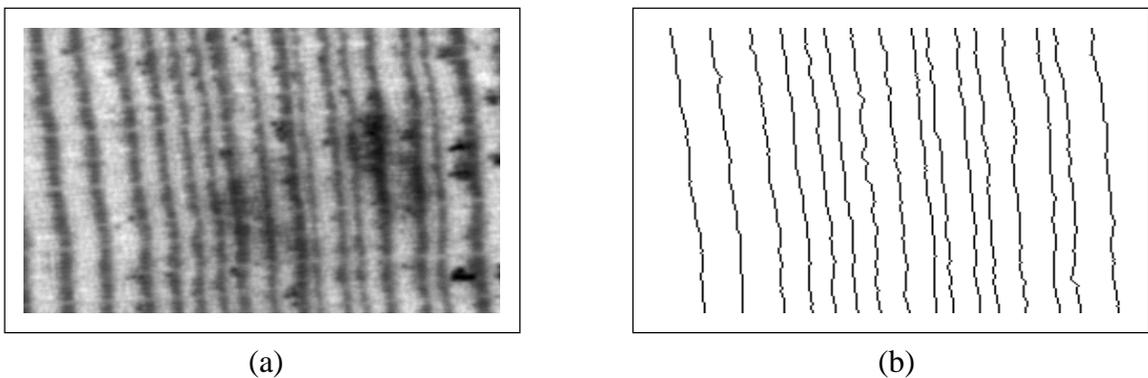


Figure 3-16. Ring detection in the presence of low-contrast narrow rings.

Chapter 4 EDGE MAPS

4.1 Memory and Disk Requirements

Edge maps obtained in the TREES system indicate the location of latewood-earlywood boundaries in the image. The non-zero pixels in an edge map represent edges while the zero-pixels represent background. In the TREES system v1.0, the edge maps are represented as raster images both in the main memory and on the disk. Due to the high resolution of image capture in the system ($10\mu\text{m}/\text{pixel}$) and the large scan areas, the raster images representing edge maps tend to be large in size. For example, to cover a 25cm section of a sample, a total of 29 frames would be required. The resulting edge map would be approximately $1024 \times 25,000$ pixels, assuming a horizontal scan. This translates to over 25MB of memory to hold the edge map. Since multiple samples will have to be archived on the disk, the raster image representation for edge maps causes the disk space requirements to grow rapidly.

In addition, during the process of ring width measurement, edge map scaling, boundary insertion, etc., there is a need to identify the components in the edge map. A connected-components labeling on this image would require approximately 102MB of memory, initially, using the iterative algorithm presented in [9]. Considering that the TREES application resides in the system memory, this requirement poses an additional burden on the resources. Moreover, each connected-components labeling operation would mandate at least two scans over the entire edge map. This significantly increases

the time required to analyze a sample. To reduce both the memory required and the analysis time, a chain-coding scheme is presented in the following section.

4.2 Chain Codes

Chain codes provide a compact format for representing one-pixel wide edges in an image. Recall that a binary raster image would suffice for archiving edge maps, using a 1-bit to mark an edge pixel and a 0-bit otherwise. However, a raster image is just a block of pixels as compared to the chain-code notation, which allows object handling while keeping the memory requirement low. An example chain-code operator for 8-connected edges is shown in Figure 4-1. The starting pixel for chain coding is usually an end point of the edge and all the other edge pixels are coded with reference to this pixel. In the example below, the co-ordinates of the reference pixel, (0,2), constitutes the chain initializer and the code segment is the sequence 3, 3, 4, 5, 3, 5, 5.

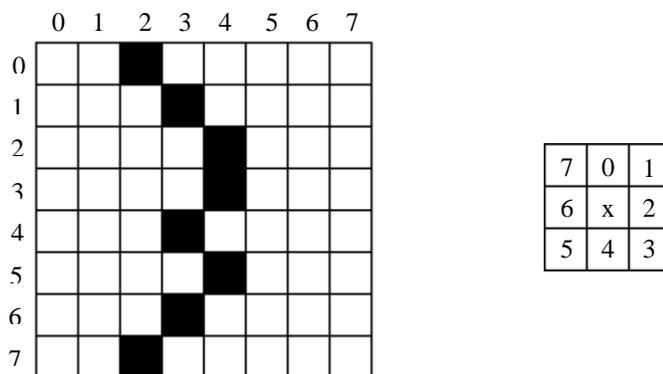


Figure 4-1. Edge pixels in a raster image and the chain code operator.

As evident from the example, for an edge with p pixels, the number of bytes required,

n , is

$$n = (p - 1) * size(code) + size(initializer)$$

Thus for a 1024 x 15,000 pixel edge map having 100 rings, each of length 1024 pixels, the size of the chain code map would be approximately 100KB compared to 15MB for a raster map. This evaluates to a compression factor of approximately 150. The actual compression achieved would vary according to the ratio of edge to non-edge pixels. In general, the compression achieved for a sample with narrow tree-rings is lower than that for a sample with wide tree-rings.

4.2.1 Data Structure

An efficient data structure design allows for easy manipulation of ring boundaries. The data structure components, as implemented in the TREES system v2.0, for representing edge maps with chain codes are shown in Figure 4-2. The variables used are described in

TABLE 4-1

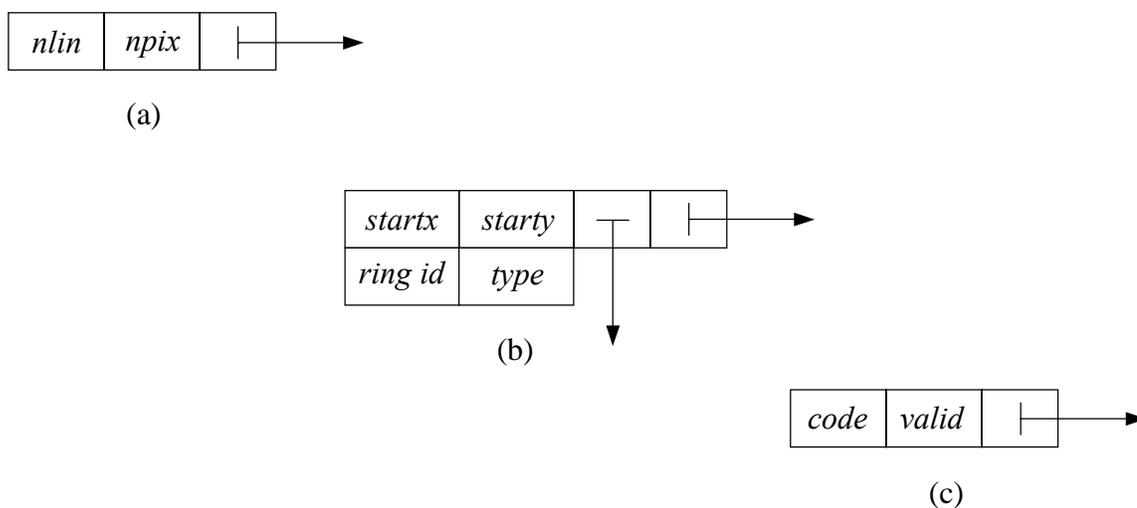


Figure 4-2. Components of the chain code data structure.

(a) Chain header. (b) List node. (c) Chain node.

Parameter	Variable Name	Data Type (bytes)
Number of Lines	<i>nlin</i>	unsigned integer (4)
Number of Columns	<i>npix</i>	unsigned integer (4)
X Co-ordinate of Starting Pixel	<i>startx</i>	unsigned integer (4)
Y Co-ordinate of Starting Pixel	<i>starty</i>	unsigned integer (4)
Ring Index	<i>ring_id (ring id)</i>	signed integer (4)
Ring Type	<i>ring_type (type)</i>	unsigned character (1)
Chain Code	<i>code</i>	unsigned character (1)
Pixel Validity	<i>valid</i>	unsigned character (1)

TABLE 4-1. Elements of the chain code data structure.

The chain header is the starting point in the data structure. It stores the number of lines in the image, *nlin*, and the number of pixels per line in the image, *npix*. In addition, the header points to the list of rings in the edge map. The list node is the starting point for a ring boundary. It contains the chain initializer, i.e. the co-ordinate pair formed by *startx* and *starty*, as the reference for the rest of the chain codes to follow. The list node also stores the ring index, *ring id*, and the ring type. The ring index is a number that is used to uniquely identify a ring in the list. The ring type is one of *normal*, *added*, or *removed*. Rings that are found by the system are tagged as normal rings. Ring boundaries that are inserted by the analyst after the edge detection process are tagged as added rings. In case the system identifies false rings, the dendrochronologist is allowed to remove them from analysis by tagging them as removed rings. The chain node contains the actual chain code, in addition to a *valid* field that indicates whether or not the pixel is part of a link. The chain code data structure is composed from the chain header, list nodes, and chain nodes as shown in Figure 4-3.

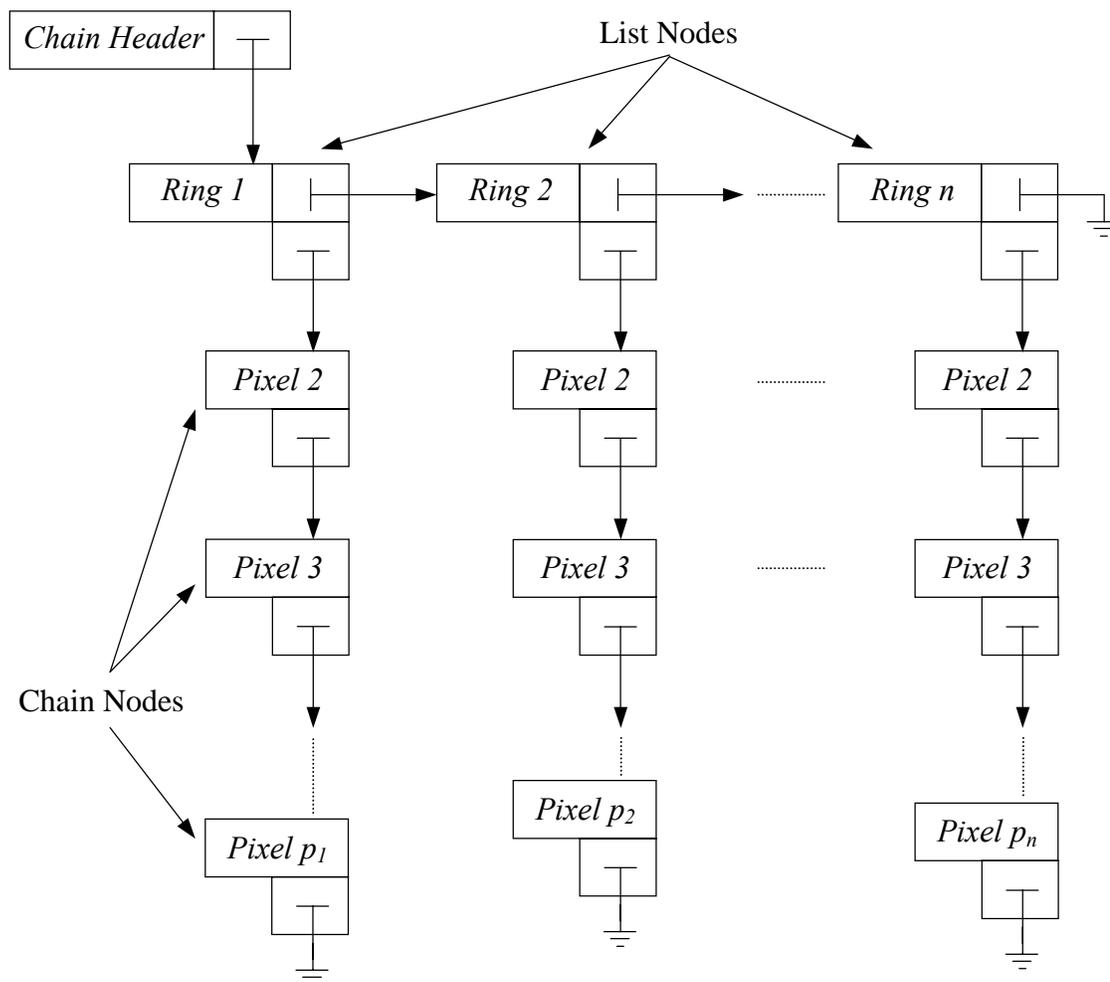


Figure 4-3. Sample chain code structure.

4.2.2 File Format

The binary file format for archiving edge map chain codes to the disk is shown in Figure 4-4. This format simplifies the reading of chain codes from the disk into the data structure. Though, at present, the header consists only of $nlin$ and $npix$, a 512-byte space allows for future expansion. The termination sequence 9,0 marks the byte boundary between the code segments of adjacent rings. Considering the overheads of the chain

header, *ring id*, *ring type*, and *valid* fields, the compression achieved, on an average, drops to approximately 75. While this format requires much more disk space than a raw bitmap, the advantages of object handling justify the additional overhead.

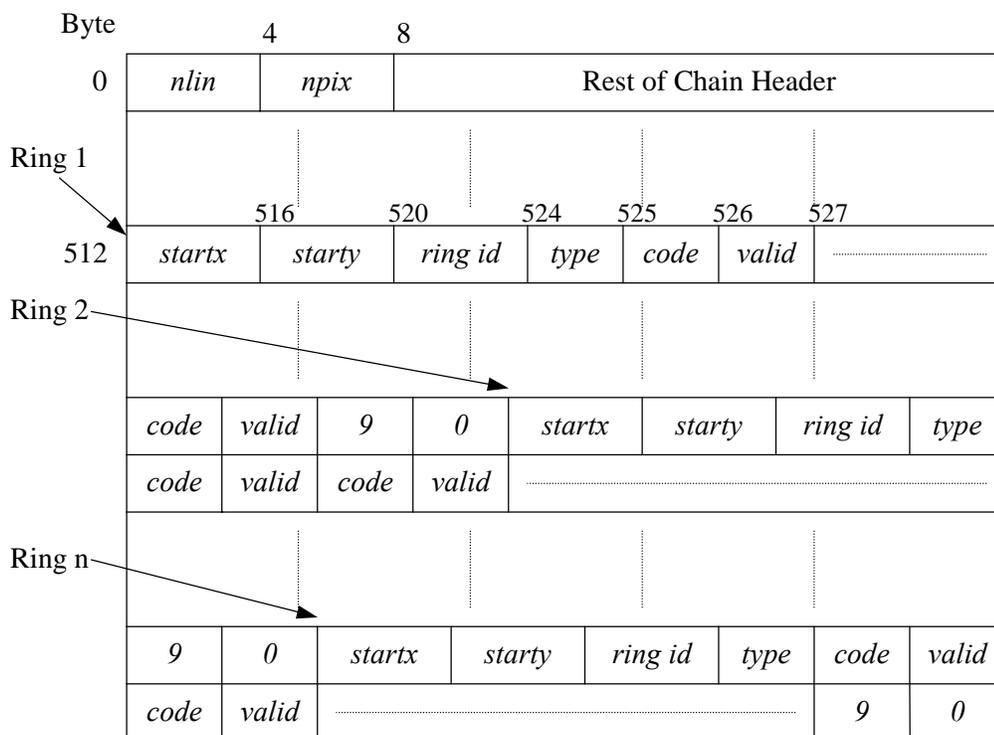


Figure 4-4. Binary file format for archiving chain codes.

4.3 Operations on Chain Codes

4.3.1 Down Sampling Edge Maps

Due to the high magnification of image capture, the mosaic image tends to be too large to be displayed at full resolution even for small samples. To overcome this problem, the mosaic is down-sampled and displayed at a lower resolution as a browse image. In the TREES system, the browse image corresponds to down sampling by a factor of six. At

this resolution, the analyst could scroll through the browse image viewing four frames at a time. Figure 4-5 shows a sample tree-ring image at full resolution.

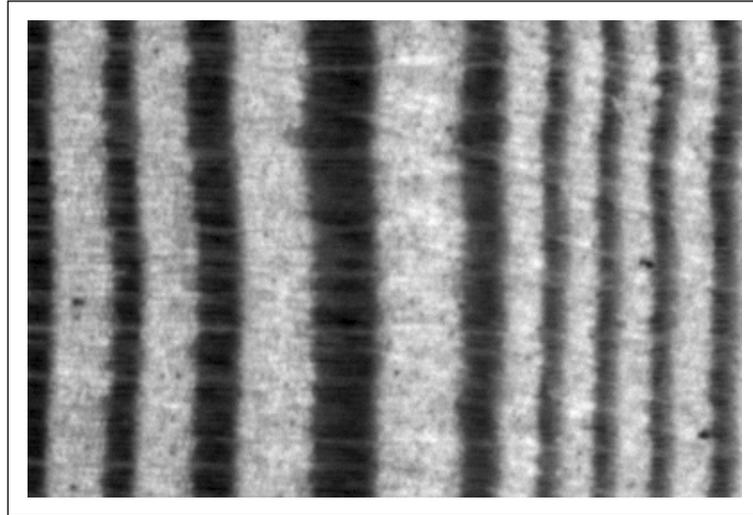


Figure 4-5. Sample tree-ring image at full resolution.

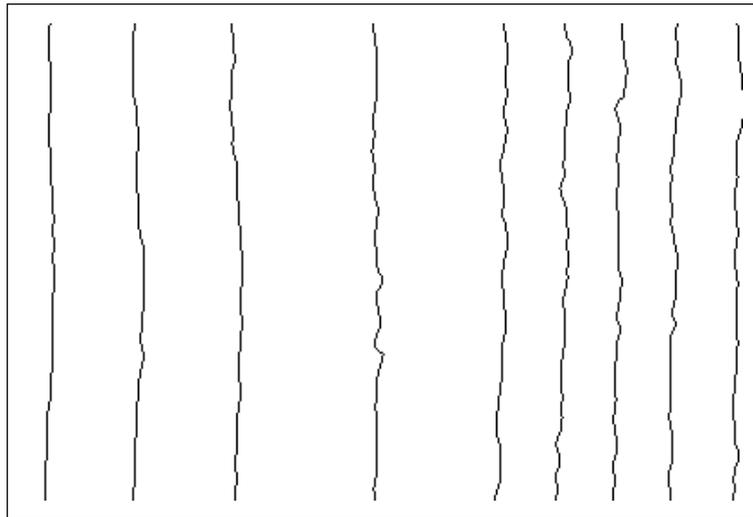


Figure 4-6. Edge map for the sample in Figure 4-5.

The ring boundaries identified by the system will also have to be displayed on the browse image to allow the analyst to identify missed or false rings. Figure 4-6 shows the edge map obtained for the image in Figure 4-5 at full resolution. Since the rings are detected at full resolution, it becomes necessary to down-sample the edge map for an overlay on the browse image. A low resolution edge map, $e_l(n_1, n_2)$, obtained from the full resolution edge map, $e_f(n_1, n_2)$, as

$$e_l(n_1, n_2) = e_f(n_1/6, n_2/6)$$

would break the continuity in ring boundaries as shown in Figure 4-7.

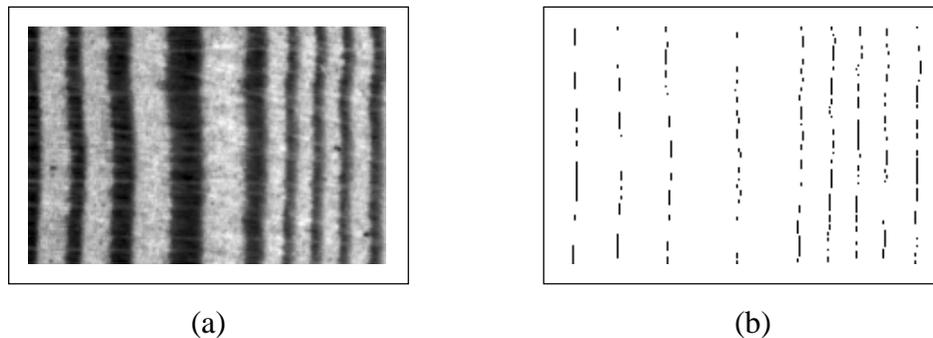


Figure 4-7. Down-sampled image and edge map for the sample in Figure 4-5.

To overcome this problem and to avoid recreating the raster edge map, the chain-coded ring boundaries are down-sampled along their length as outlined in the algorithm below:

```
procedure Subsample
for R = 1 to NRINGS
  [Lstart_x, Lstart_y] = [Hstart_x/6, Hstart_y/6]
  [Hprev_x, Hprev_y] = [Hstart_x, Hstart_y]
  [Lprev_x, Lprev_y] = [Lstart_x, Lstart_y]
  skip = 0;
  for P = 2 to NPIXELS
    [HCurr_x, HCurr_y] = DECODE(HPrev_x, HPrev_y, HCode(R, P))
    if skip = 5
```

```

    then
      LCode(R, P/6+1) = ENCODE(LPrev_x, LPrev_y, HCurr_x/6, HCurr_y/6)
      LPrev_x = HCurr_x/6
      LPrev_y = HCurr_y/6
      skip = 0
    else
      increment skip
  end for
end for

```

The prefix H (eg. Hstart_x) implies that the co-ordinates refer to the full-resolution image. An L prefix (eg. Lstart_x) is used for low-resolution co-ordinates. The down-sampled edge map obtained by this approach is shown in Figure 4-8. Since the down-sampled edge map is required only for display and no measurements are obtained from it, this approach is adequate for representing ring boundaries on the browse image.

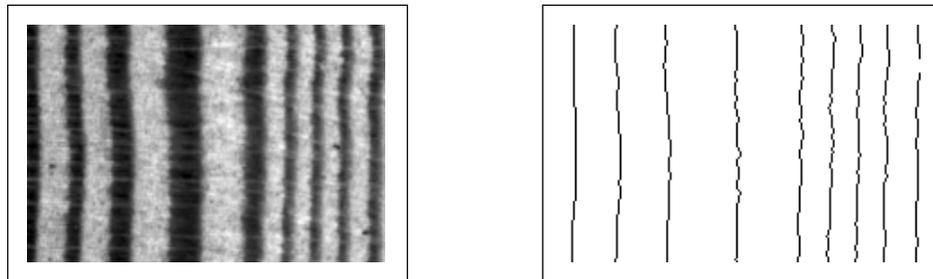


Figure 4-8. Edge map obtained by down sampling along the tree-ring direction.

4.3.2 Width Measurement from Chain Codes

For the purpose of cross-dating tree ring samples, the most commonly used features are the ring widths. As stated earlier, it is possible to match different samples based on the pattern of variation in ring widths rather than comparing absolute widths. In the manual measurement method, ring widths are often measured along a single line. However, since

the TREES system analyzes much more data, i.e. the entire region of interest, more accurate width measurements could be obtained by averaging the widths obtained at different points between two rings. In other words, ring width in the TREES system refers to the average distance between a pair of adjacent rings. This method of width measurement is less prone to errors arising due to the presence of wood anomalies along the ring boundaries.

Since tree rings exhibit appreciable variation in their orientation at different regions within a cross-section, it becomes necessary to define the direction of width measurement. If the tree rings were known to be concentric, it would be possible to define ring width as the radial distance between a pair of ring boundaries. However, this is rarely the case and a more practical approach would be to measure width orthogonal to the ring at the point of measurement. But since the ring boundaries are often not smooth curves, the normal vector used for measurement should be orthogonal to the low frequency component of the tree ring curve rather than the high frequency components, which contain small dips and bumps in the ring [2].

The gradient direction $\theta(r, c)$ computed at each pixel is an indication of the maximum grayscale intensity change at that pixel. For pixels that constitute ring boundaries, it is an indication of the orthogonal to the latewood-earlywood boundary at those points. However, the local gradient direction is extremely sensitive to wood anomalies along the ring boundary such as resin ducts, prominent rays, etc. Since the ring boundaries identified by the system are reliable representations of the location of

latewood-earlywood transitions, a better approach is to model the ring boundaries to be piecewise linear. The segment length for the model should be large enough to smooth the effects of small twists and curves. However, the segment length should be kept small enough to follow large curves in the rings. At the magnification of image capture used in the TREES system, a segment length of twenty to thirty pixels has been found to be sufficient, with the point of interest at the center of the segment [2].

A least squares fit algorithm is used in the TREES system to approximate the orientation of tree ring boundaries. To determine the orthogonal to the ring boundary at each pixel, a segment length of N pixels, centered at the point of interest, is used. If the co-ordinates (x_i, y_i) of the pixels in the data set are known, the center of the set is at:

$$\bar{x} = \frac{\sum_{i=1}^N x_i}{N}$$

$$\bar{y} = \frac{\sum_{i=1}^N y_i}{N}$$

The angle of orientation of the edge points that minimizes the sum of squared errors for the points in the segment can be computed as [10]:

$$\alpha = \frac{1}{2} \tan^{-1} \left(\frac{b}{a-c} \right),$$

with parameters a , b , and c computed as:

$$a = \sum_{i=1}^N (x_i - \bar{x})^2,$$

$$b = 2 \sum_{i=1}^N (x_i - \bar{x})(y_i - \bar{y}),$$

$$c = \sum_{i=1}^N (y_i - \bar{y})^2$$

Since it is common to analyze a sample going from the pith outwards to the bark, with the pith to the left, the orthogonal angle to the tree ring boundary is obtained by subtracting $\pi/2$ radians from the computed tangent angle α .

Once the orthogonal directions are computed for all pixels on the ring boundaries, the average distance between two adjacent rings is computed along the orthogonal vector. To measure the distance at a pixel on the left ring, pixel-by-pixel steps are taken along the orthogonal vector at that pixel, until a pixel on the right ring is encountered. At this point, the angle between the orthogonal at the pixel on the left ring and the orthogonal at the ending pixel on the right ring is compared. If the two orthogonal vectors are within ten degrees of each other, then the distance measured is considered valid. These distances are then averaged over the number of valid measurements that were made between the two rings. Since the edge maps are chain coded, the technique used in the TREES system v2.0 is to compute the normal, α , at each pixel on the left ring. With the chain initializer as the starting point, the co-ordinates of the remaining pixels on the left ring could be decoded serially from the chain codes for the left ring. Then, the normal at each pixel on the right ring is computed similar to the left ring but using the chain codes for the right ring. Once the normal is computed at all pixels for a pair of adjacent rings, a series of width measurements are made, one for every pixel on the left ring along the width line (orthogonal vector). For each measurement, pixel-by-pixel steps are taken along the orthogonal at the pixel under consideration (starting point). Upon encountering a pixel on

the right ring (ending point), the normal at the ending point is checked with the normal at the starting point for that measurement. If they lie within ten degrees of each other, the measurement is considered valid. A counter that keeps track of the number of valid measurements is incremented for that pair of ring boundaries. Once measurements are obtained for all pixels on the left ring, the average ring width for a ring-boundary pair is computed.

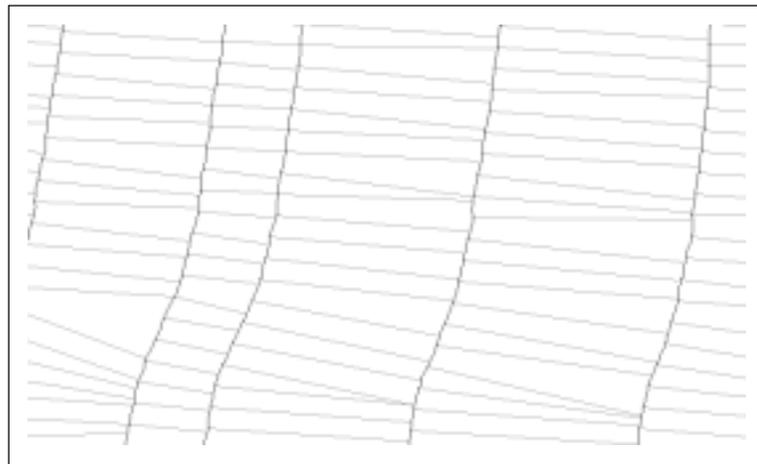


Figure 4-9. Width lines overlaid on a sample edge map.

This approach is computationally efficient since the edge pixels of a ring boundary could be serially decoded from a chain code representation for calculating the normal. This technique also eliminates the need to label the raster edge map, as in the TREES system v1.0, to identify edge pixels of a ring boundary. In addition to reducing the time required to analyze a sample, width measurement from chain code representation significantly lowers the memory required.

4.4 Average Grayscale Profile

In addition to latewood-earlywood boundaries, other features such as the average grayscale profile could be extracted from the image. Once the ring width measurements have been made, it is possible to compute the average grayscale profile. The average grayscale profile is a one-dimensional sequence of numbers that represents the average variation in the grayscale intensity along a profile orthogonal to the ring boundaries. It is important to note that the average grayscale profile is determined along the width lines used for measurement and not along the rows in the image [2]. To create the average grayscale profile, the width lines between two adjacent ring boundaries are traversed pixel-by-pixel, and at each pixel, the grayscale intensity is added to an accumulator array at an index that is determined by the location of the pixel on the width line expressed as a fraction of ring width. A counter array is incremented at the same index. Once this procedure is repeated for the other width lines, the average grayscale profile between the pair of ring boundaries is computed by dividing the grayscale accumulator array by the counter array at the corresponding indices. The average grayscale profile so obtained has a length equal to the average width of the ring.

The grayscale accumulator and counter arrays are initialized to be of length \overline{W} , i.e. the average width for the ring under consideration. For a pixel that occurs at the j^{th} position along the i^{th} width line, its grayscale value is added to the accumulator at the index:

$$index_{avg}(i, j) = \left(\frac{j}{W_i} \right) \overline{W}$$

This procedure is repeated for all the rings, and the average grayscale profile is determined for each adjacent pair of ring boundaries. Concatenating the individual average grayscale profiles then forms the average grayscale profile for the cross-section of the image being analyzed. The average grayscale profile obtained for a sample image is shown in Figure 4-10.

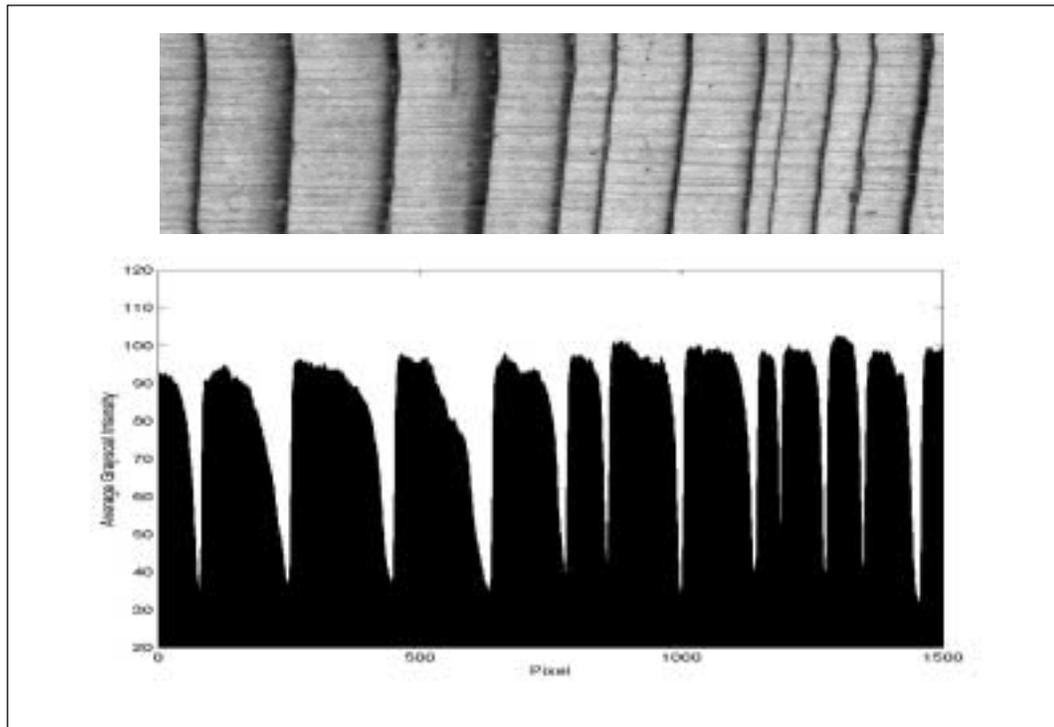


Figure 4-10. Average grayscale profile for a sample cross-section.

4.4.1 Extracting Missed-Ring Boundaries

Though the edge detection algorithm implemented in the TREES system v2.0 is more robust than the modified Canny algorithm used in the TREES system v1.0, there could

still be circumstances under which the system might fail to identify one or more ring boundaries. These rings are termed *missed rings*, implying that the system failed to find them, as opposed to *missing rings* that refer to the rings that are actually absent in the sample. Recall that the edge-tracking algorithm incorporates a threshold that limits the number of break points a ring boundary could have and still be connected. The TREES system v2.0 could fail to detect a ring boundary because of the limiting value of the threshold. For example, if the number of prominent rays cutting across a ring boundary were to exceed this limit, the algorithm discards the edge, treating it as a non-ring feature. However, the average grayscale profile determined for a sample could be used to identify missed rings, as shown in Figure 4-11. Since the average grayscale profile is computed by summing and averaging the grayscale intensity along the orthogonal width lines, the average locations of missed rings obtained from the average grayscale profile is fairly accurate. The location of missed rings could be determined as a percentage of the distance between the enclosing ring boundaries. By knowing the fraction of width at which the missed ring boundary lies between the two enclosing ring boundaries, the missed-ring profile could be extracted with reasonable accuracy for the purpose of display.

To extract the ring profile, first the missed-ring location is determined from the average grayscale profile as shown in Figure 4-11. The first derivative of the average grayscale profile is computed. Notice the peaks in the first derivative at the locations of ring boundaries. In the TREES system v2.0, a threshold is defined for each pair of ring boundaries. This is computed as $1/3$ the average of the first derivative of the

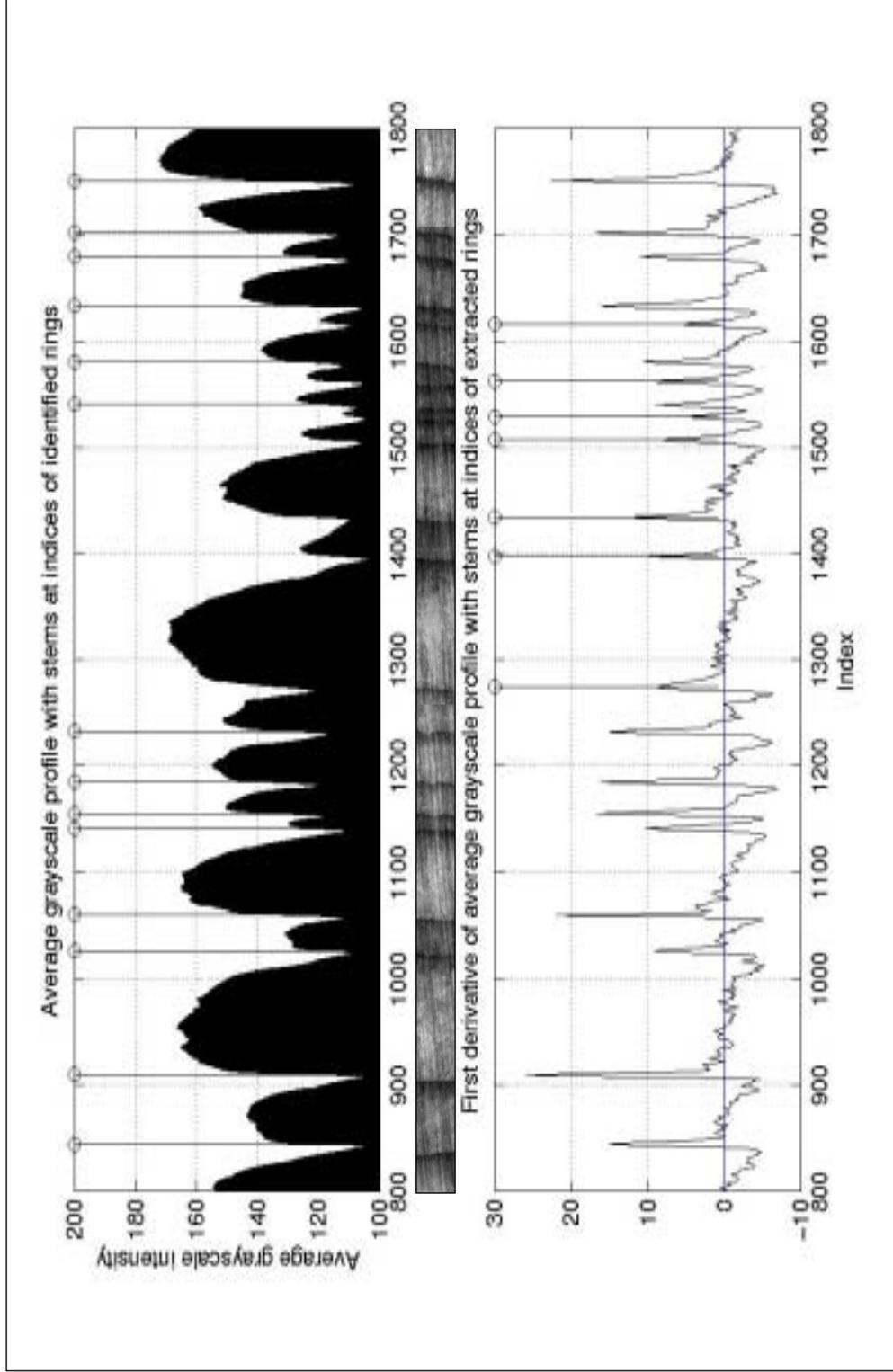


Figure 4-11. Average grayscale profile and its first derivative.

average grayscale profile at the locations of the enclosing ring boundaries. Locations of peak overshoots of the first derivative above the computed threshold are taken as an indication of the location of missed rings. The fractional location of the missed ring is determined as:

$$f = \frac{index_{missed} - index_{left}}{index_{right} - index_{left}},$$

where $index$ refers to the pixel position in the one-dimensional grayscale profile. Let p_{left}^i represent the i^{th} pixel on the left boundary. Similarly p_{right}^i represents the i^{th} pixel on the right boundary. The notation $row(p^i)$ will be used to represent the row co-ordinate of the i^{th} pixel. Similarly $column(p^i)$ will be used to represent the column co-ordinate of the i^{th} pixel. Once the relative location, f , of the missed ring is determined, the next step in extracting the missed ring profile is to determine the vertical overlap between the enclosing rings. The starting point of vertical overlap, p_{left}^{start} , on the left boundary is determined as:

$$p_{left}^{start} = p_{left}^i \mid row(p_{left}^i) = \max(row(p_{left}^1), row(p_{right}^1))$$

Similarly the starting point of vertical overlap, p_{right}^{start} , on the right boundary is determined as:

$$p_{right}^{start} = p_{right}^i \mid row(p_{right}^i) = \max(row(p_{right}^1), row(p_{left}^1))$$

Since the profile of the missed ring boundary could be determined only to the extent of vertical overlap between the enclosing ring boundaries, the end points of vertical overlap on the left and right boundaries are determined:

$$p_{left}^{end} = p_{left}^i \mid row(p_{left}^i) = \min(row(p_{left}^{n_1}), row(p_{right}^{n_2})),$$

$$p_{right}^{end} = p_{right}^i \mid row(p_{right}^i) = \min(row(p_{right}^{n_1}), row(p_{left}^{n_2})),$$

where n_1 and n_2 are the number of pixels constituting the left and right ring boundaries respectively. The starting pixel on the missed boundary is computed as:

$$row(p_{missed}^1) = row(p_{left}^{start})$$

$$column(p_{missed}^1) = column(p_{left}^{start}) + (column(p_{right}^{start}) - column(p_{left}^{start}))f$$

The remaining pixels on the missed boundary are determined as:

$$row(p_{missed}^{1+i}) = row(p_{left}^{start+i})$$

$$column(p_{missed}^{1+i}) = column(p_{left}^{start+i}) + (column(p_{right}^{start+i}) - column(p_{left}^{start+i}))f$$

The average grayscale profile shown in Figure 4-11 corresponds to the Juniper sample shown in Figure 4-12. The missed ring boundaries extracted from average grayscale profile are shown in Figure 4-13.

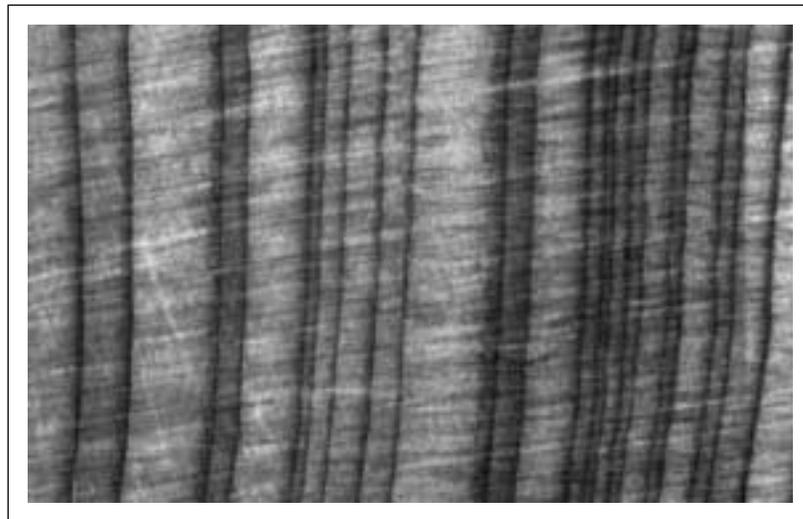


Figure 4-12. Cross-section of a Juniper sample with narrow rings.

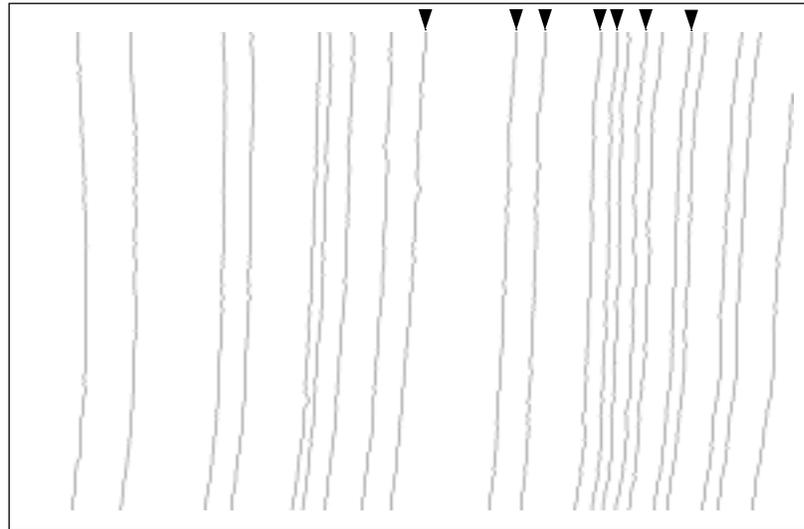


Figure 4-13. Missed rings extracted from average grayscale profile.

4.4.2 Extracting Earlywood-Latewood boundaries

The primary tree ring attribute used for cross-dating samples is the ring width variation. Ring width refers to the distance between latewood-earlywood boundaries of adjacent rings. Other ring attributes are often used to validate the chronological match established between samples. One example is the width between earlywood-latewood boundaries of adjacent rings. In order to measure earlywood-latewood widths, the exact locations of these boundaries have to be established. Recall that the earlywood-latewood boundaries are transitions from light-colored to dark-colored annular regions in the wood. This is in contrast to the latewood-earlywood boundaries that have been referred to as ring boundaries in this thesis. The gradient at pixels on the earlywood-latewood boundary is opposite ($\pm\pi$ radians) to the gradient at pixels on the latewood-earlywood boundary. Therefore, it appears logical to locate the earlywood-latewood boundaries by isolating

their flow patterns using phase reversed dominant gradient directions. However, unlike the latewood-earlywood boundaries that are well defined, the earlywood-latewood boundaries tend to be ill-defined. To address this problem, the earlywood-latewood widths are extracted directly from the average grayscale profile. This ensures that the width measurements are less prone to errors arising due to the noisy nature of earlywood-latewood boundary profiles.

The average grayscale profile obtained for a sample image is shown in Figure. From the first derivative of the average grayscale profile, it is seen that negative peaks mark the locations of earlywood-latewood boundaries. These negative peaks, which correspond to the zero-crossings in the second derivative of the average grayscale profile, indicate the points of steepest fall in the gradient magnitude along the width lines. Thus by identifying negative peaks in the first derivative, the earlywood-latewood widths could be determined. The earlywood-latewood boundary profiles are extracted similar to the technique used to find missed rings.

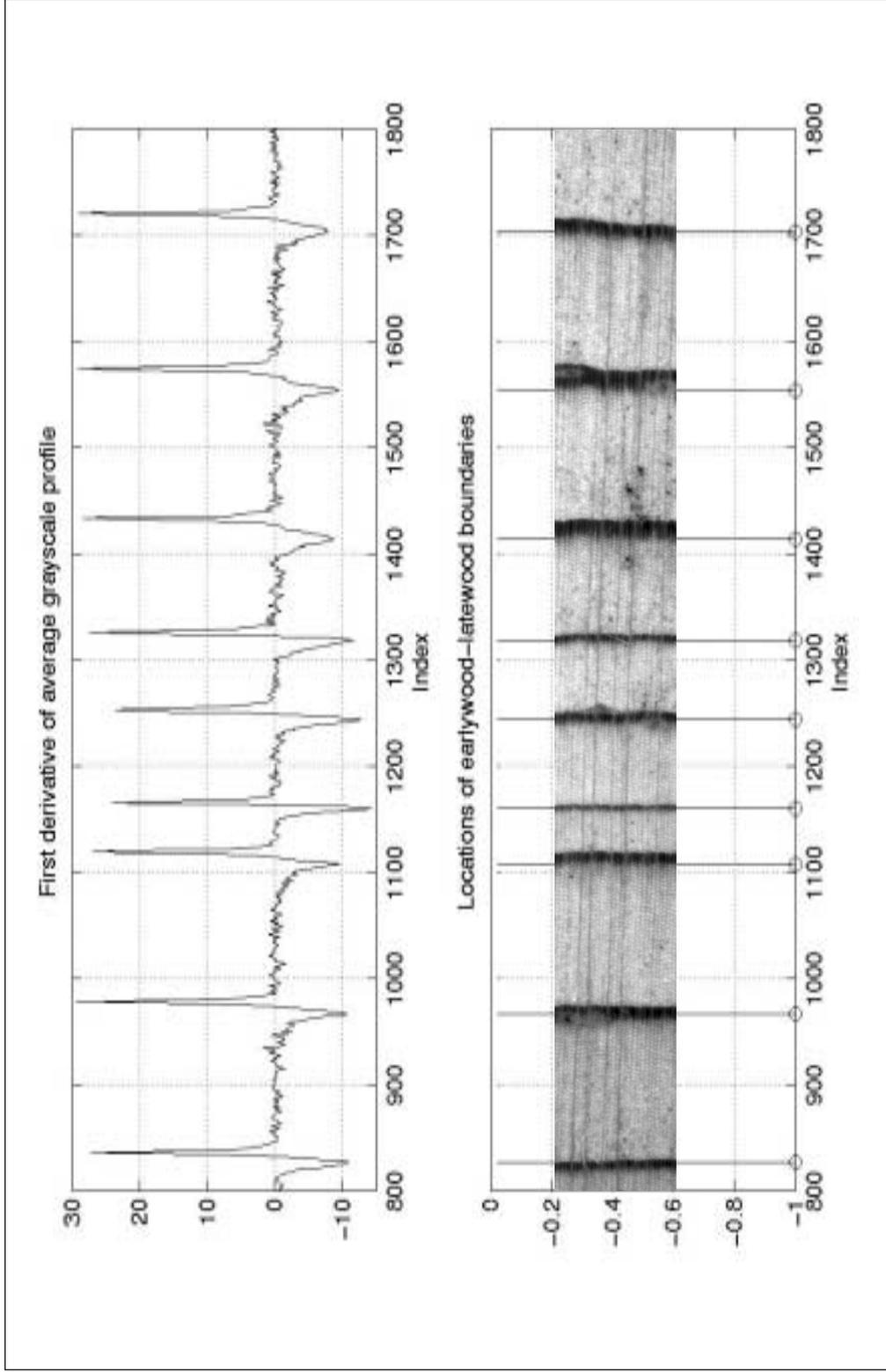


Figure 4-14. Earlywood-latewood width measurement from average grayscale profile.

Chapter 5 ANALYST INTERACTION

An important design philosophy in the TREES system is to incorporate analyst interaction at every stage during sample analysis. In view of this philosophy, the TREES system has been developed to be a computer-assisted, rather than a fully automated system. However, the initial version of the TREES system is limited in its ability to process analyst inputs. The improvements in the TREES system v2.0 allow analyst interaction before, during, and after the ring identification process. A discussion of the interactive features in the TREES system v2.0 is presented in this Chapter.

5.1 Region-of-Interest Selection and Processing

Recall that the TREES system v1.0 allows the analyst to select a section within the wood sample for analysis. The system then captures a linear sequence of partly overlapping 1024 x 1280 pixel frames to compose the mosaic image. In the TREES system v1.0, ring identification is carried out on the entire mosaic image. However, there could be regions in the mosaic image that the analyst might want to avoid. The TREES system v1.0 lacks the capability to allow the analyst to select a region of interest within the mosaic image. In addition, the TREES system v1.0 cannot analyze core samples, Figure 5-1, since it requires tree rings to extend from the frame top to the bottom. To overcome this problem, TREES system v2.0 incorporates region-of-interest selection and processing using a boundary-marking approach.

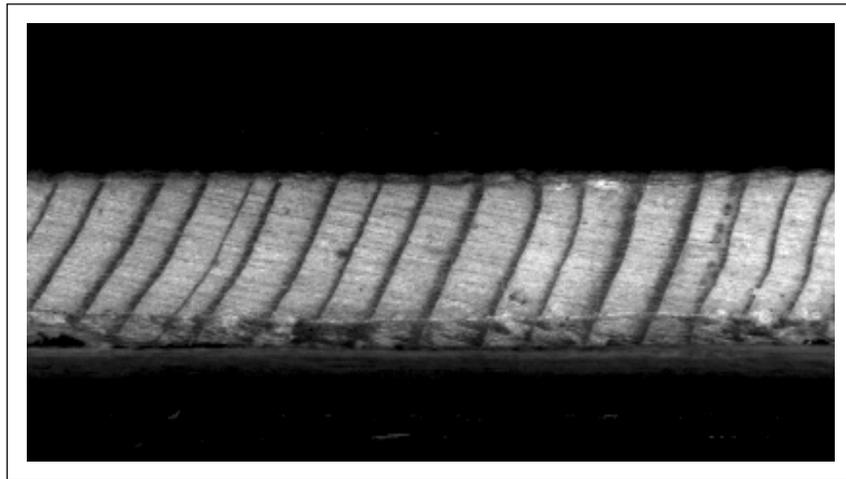


Figure 5-1. Single frame captured from a core sample.

5.1.1 Processing Modes

Processing mode in the TREES system v2.0 refers to the combination of processing resolution, region-of-interest, and ring detection mode. Each of these parameters could be selected independent of the others. The processing resolution defaults to 1:1 (image capture resolution), but the system provides an option to process the mosaic at a lower resolution (1:2). The lower processing resolution is useful for well-behaved samples with complacent rings. In all samples, the analyst is required to visually inspect the mosaic image for narrow rings before choosing the processing resolution. The low-resolution mosaic image is obtained by averaging and sub-sampling the mosaic image. Therefore, in samples with narrow rings, the system would be unable to resolve two closely spaced ring boundaries if they were to merge on sub-sampling. Although there is a loss in accuracy at lower resolution, the time required to analyze a sample is drastically reduced.

The TREES system v1.0 lacks the capability to process a region-of-interest within the mosaic image. The presence of wood anomalies in the mosaic image would mandate selection of a sub-section within the mosaic image. The TREES system v2.0 processes the entire mosaic by default, but also allows the analyst to select a region-of-interest within the mosaic. The region-of-interest is restricted to a rectangular region, oriented at any angle, as shown in Figure 5-2. In addition, since the core samples do not cover the entire frame along its vertical dimension, it becomes necessary to select just the core and leave out the blank pixels, as shown in Figure 5-1. This enables the system to analyze core samples.

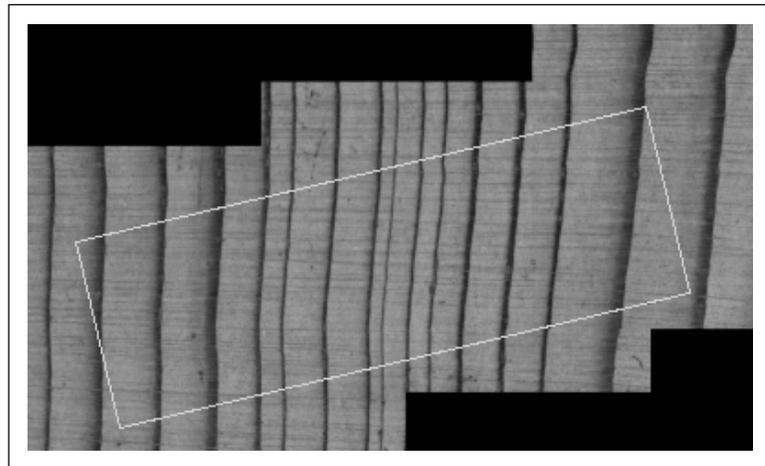


Figure 5-2. Region-of-interest selection.

The TREES system v2.0 can operate in one of the two modes during the ring detection process. The first is an automated mode suitable for well-behaved samples that are fairly easy to handle. In this mode, the system operates without analyst intervention. The second is an interactive mode that is suited for ill-behaved samples and samples with

wood anomalies. In this mode, the analyst is prompted to assist the system in ring identification when the algorithm encounters a break in the ring boundary. A detailed discussion of the interactive mode is presented in section.

5.1.2 Data Structures for Boundary Marking

The pseudo-code for a typical raster scan is shown in Figure 5-3. Irregular regions require sophisticated data structures to mark the image boundaries. The data structure adopted in the TREES system v2.0 for region-of-interest processing is shown in Figure 5-4. Although the region of interest in the TREES system is restricted to a rectangle, this data structure could be used for any arbitrary region. The raster scan method employed in conjunction with the data structure is shown in Figure 5-5. Note that unlike a regular raster image, both the number of columns per row and the number of rows per column vary within an irregular region.

```

procedure scan
for row = 1 to numLines
  for col = 1 to numPixels
    .....
    .....
    image[row][col] = .....
    .....
    .....
  end for
end for

```

Figure 5-3. Pseudo-code for a typical raster scan.

Though the scan method described above is fairly simple when used in conjunction with an appropriate data structure, operations on the image such as median filtering,

convolution, etc., involve neighborhood pixels to be taken into account. Each neighborhood pixel has to be checked to see if it lies within the region of interest. The pseudo-code for convolution on an irregular region using an $m \times n$ operator is shown in Figure 5-6.

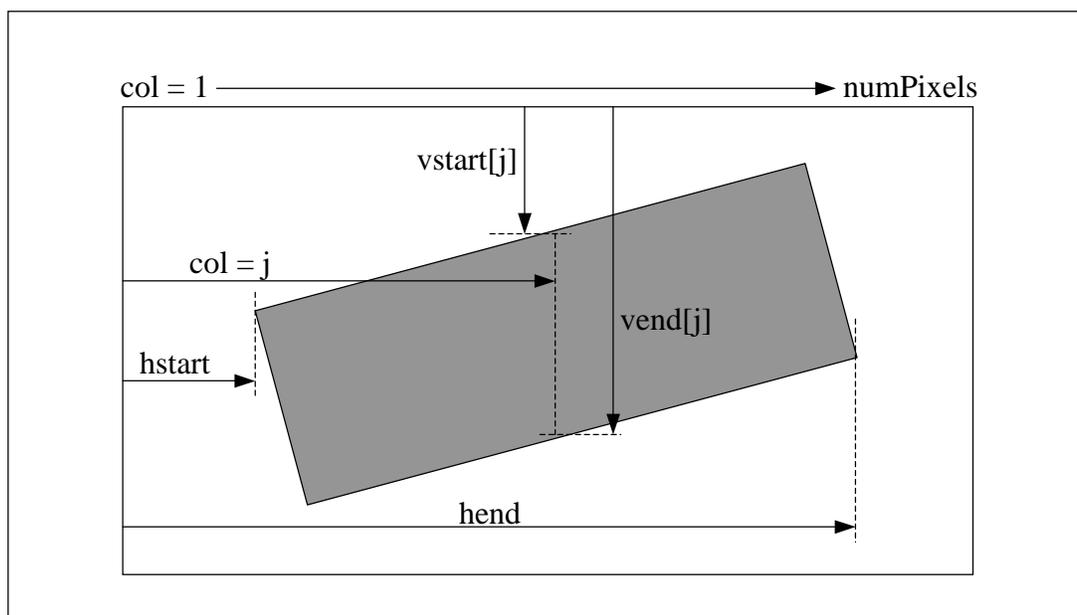


Figure 5-4. Data structure for boundary marking.

```

procedure scan
for col = hstart to hend
  for row = vstart[col] to vend[col]
    .....
    .....
    image[row][col] = .....
    .....
    .....
  end for
end for

```

Figure 5-5. Pseudo-code for scanning an irregular region.

```

procedure convolve
for col = hstart to hend
  for row = vstart[col] to vend[col]
    sum = 0
    for j = -n/2 to n/2
      if col + j < hstart or col + j > hend
        skip j
      for i = -n/2 to n/2
        if row + i < vstart[col + j] or row + i > vend[col + j]
          skip i
        sum = sum + mask[i][j].image[row + i][col + j]
      end for
    end for
    outImage[row + i][col + j] = sum
  end for
end for

```

Figure 5-6. Pseudo-code for convolution on an irregular region

5.2 Ring Boundary Detection

The interactive mode in the TREES system v2.0 allows analyst intervention during the ring identification process to assist the edge-tracking algorithm. Recall that ring boundaries are identified by tracking the path of highest gradient magnitude in the ridge areas. The presence of high contrast noise edges running across ring boundaries, shown in Figure 5-7, breaks the tree-ring flow pattern at the point of intersection, as shown in Figure 5-8. The automatic mode in the TREES system v2.0 attempts to bridge the discontinuities by fitting a polynomial to a 20-pixel wide segment of the ring just before the break point. It then extends the edge by using a piecewise-linear model until the discontinuity is bridged. This method is suited for regions that contain wide rings. In the presence of narrow rings, this approach could lead to cross-connections with adjacent

rings, similar to the fragment linking technique in TREES system v1.0. To address this problem, the TREES system v2.0 provides for an interactive mode that allows the analyst to manually bridge discontinuities in tree-ring boundaries.

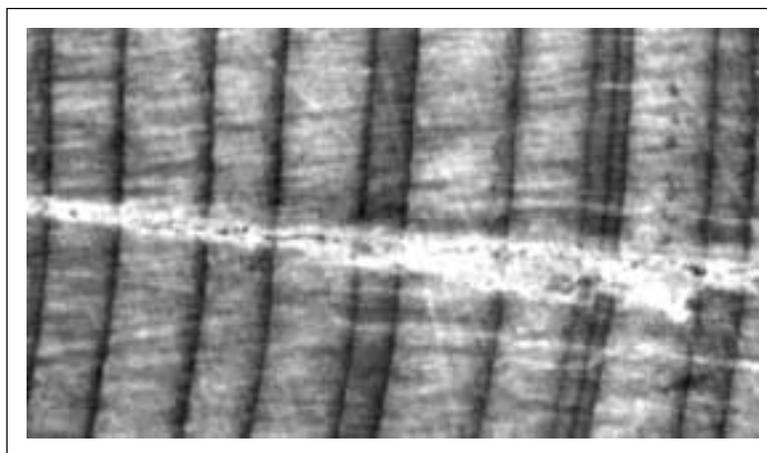


Figure 5-7. Noise features obscuring ring boundaries.

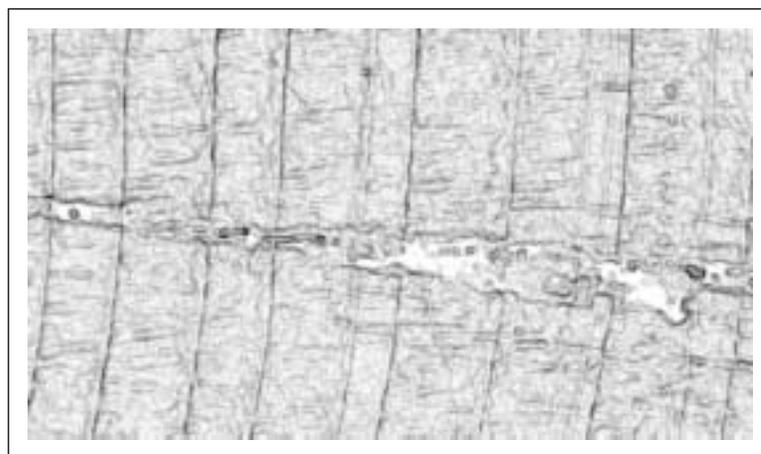


Figure 5-8. Gradient magnitude image for the sample in Figure 5-7

In the interactive mode, the TREES system v2.0 prompts the dendrochronologist to manually link tree-ring boundaries. When there is a break in the tree-ring flow pattern,

the absence of a consistent neighborhood pixel prompts the edge-tracking algorithm to seek analyst intervention. At this point, the system displays the partly tracked ring boundary in zoom window, as shown in Figure 5-9, and prompts the analyst to either manually link the discontinuity or discard the edge. Tree-ring boundaries identified by the system, with the help of analyst inputs, are shown in Figure 5-10.

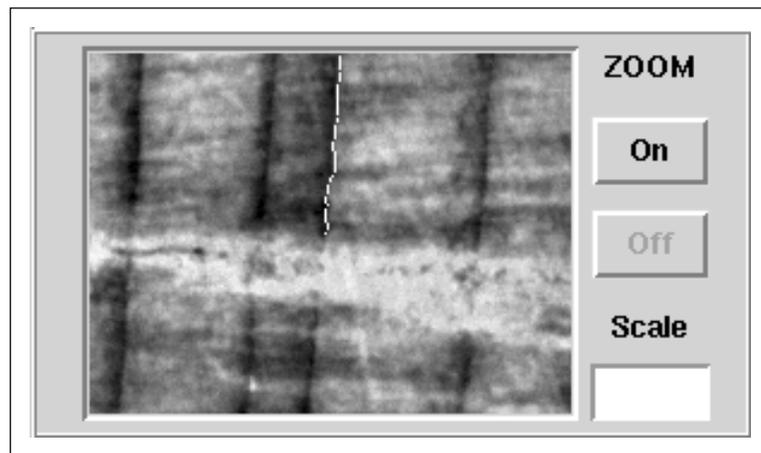


Figure 5-9. A partly detected tree-ring in the zoom window.

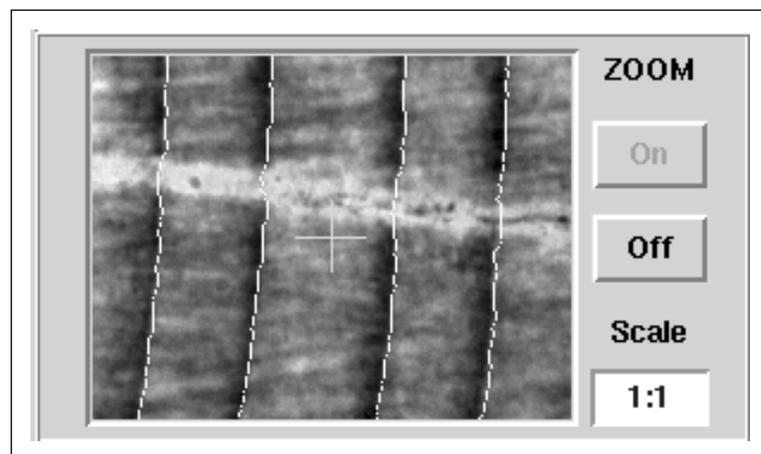


Figure 5-10. Fully detected tree-rings in the zoom window.

5.3 Wood-Centered Analysis

The TREES system has been developed to assist in the Douglass method of dating. The Douglass method of matching samples requires that the wood be available for reference during the entire process of cross-dating. Thus the Douglass method is a wood-centered, rather than image-centered, approach to tree ring analysis. However, the TREES system v1.0 lacks the ability to return to the wood sample due to hardware limitations in the X-Y stage system. The TREES system records information pertaining to a sample in its data file that allows the analysis to be stopped at any stage and continued later on. As long as the sample remains on the stage, in the same position as that of image acquisition, and the stage motors remain powered, the mapping between the browse image and the stage coordinates is maintained. This enables the system to command the stage motors to move the sample such that the desired region is in the field of view. However, if the sample is removed from the stage and returned later for analysis, or if the analysis session is different from that of acquisition (by restarting the application), then the mapping between the browse image co-ordinates and the stage co-ordinates is broken.

One way to overcome the problem is to have a limit switch for each of the X and Y directions that would communicate to the system when the stage hits the origin in either of the directions. Each time the application is restarted, the system could command the stage to move to the origin. At this point, the stage co-ordinates could be reset to (0,0). The stage co-ordinates, then, have a common reference that is maintained across sessions, samples, and application restarts. Returning to the wood involves replacing the sample, if

it was moved, in exactly the same position as it was during the acquisition session. The mapping established during acquisition could be retrieved from the data file and reused in the current session. There is a one-to-one relation between the browse image and the stage co-ordinates, and mouse clicks on the browse image could be translated to stage motion commands. However, due to the high magnification of image capture in the system, the sample has to be returned to exactly the same position during acquisition. As a more feasible solution, an approach based on co-ordinate registration is employed in the TREES system v2.0.

The TREES graphical interface is shown in Figure 5-11. For linking stage motion with the analysis graphical interface, a live video display is brought up in a window as shown in Figure 5-12. The current stage and pixel coordinates are shown in the window next to the video. The coordinate registration would require three control points to be specified in each of the live video image and the browse image. However, since the magnification remains the same across samples, the coordinate transformation evaluates to just rotation and translation. Thus only two control points would actually be required. In the TREES system v2.0, the control points required for coordinate transformation are specified by the analyst. Unique features on the wood such as resin ducts, pinprick marks, etc, could be used to specify control points on both the browse and live video image. Once control points have been identified, the coordinate transformation is computed using a method described by Wolberg in [12].

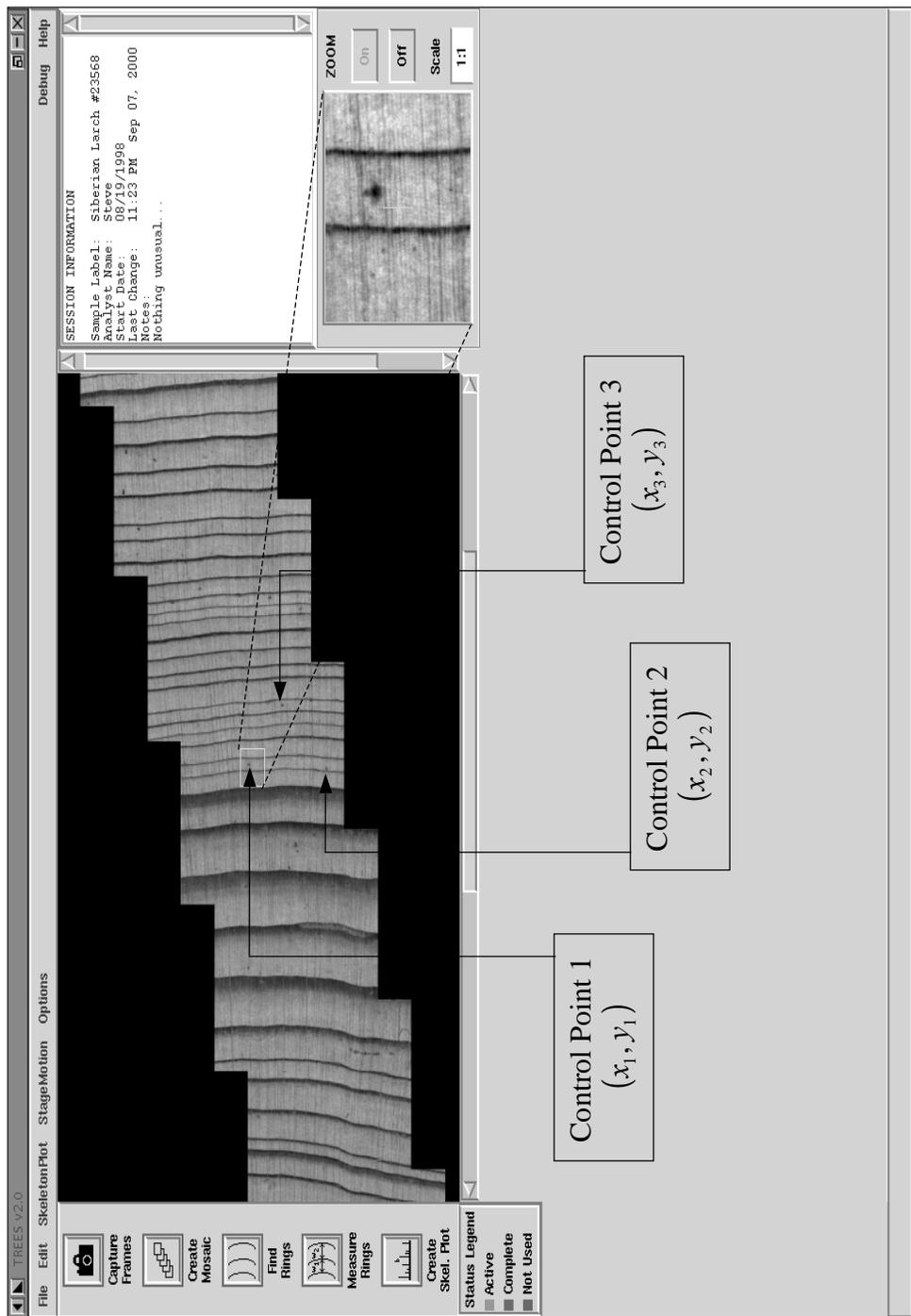


Figure 5-11. TREES graphical interface with control points marked on the browse image.

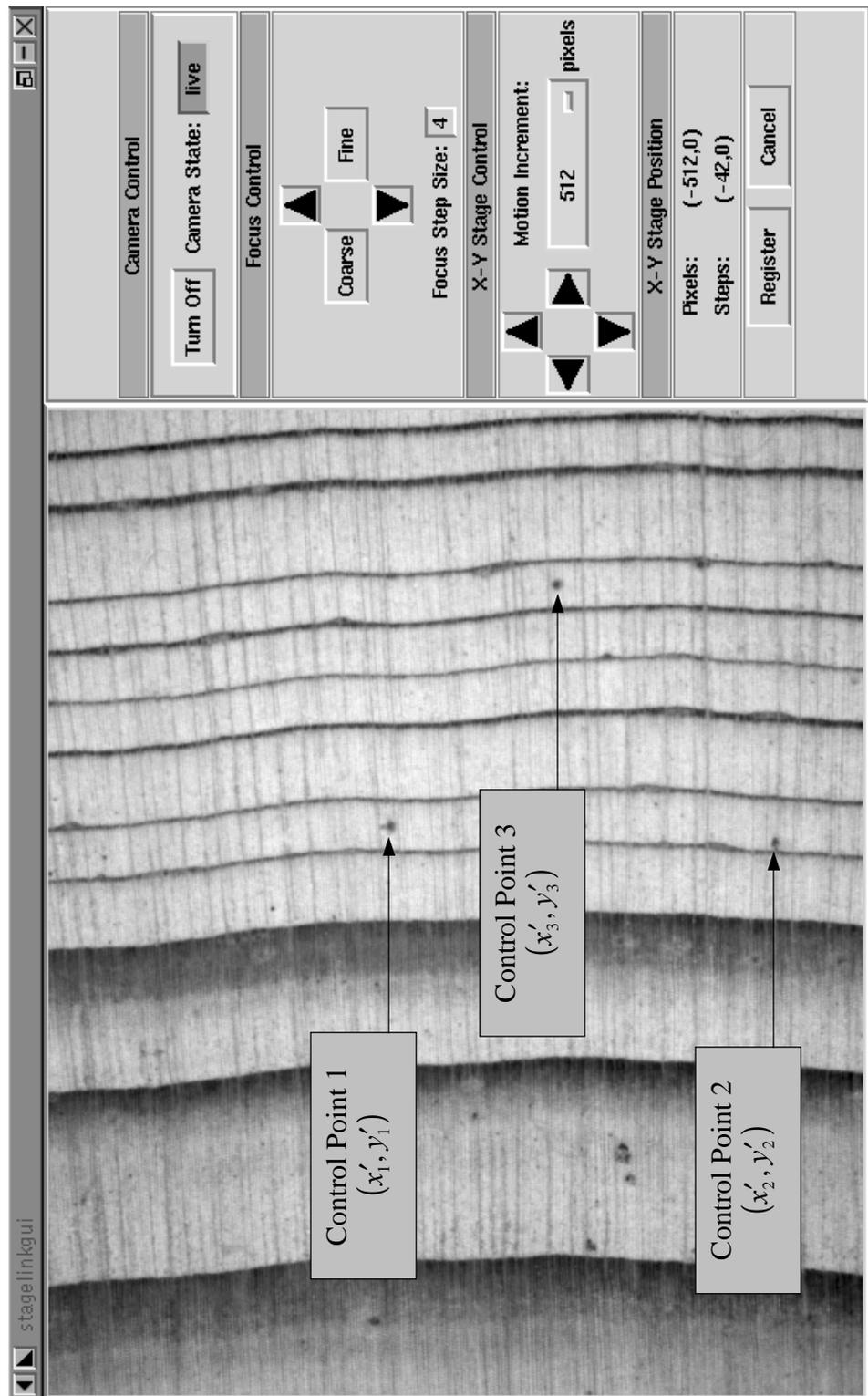


Figure 5-12. Stage Motion graphical interface with control points marked on the live video image.

Let (mx_i, my_i) represent the coordinates of control point i on the mosaic image. The mosaic coordinates of the control points are obtained from the browse image coordinates as:

$$mx_i = x_i(\text{zoomfactor}),$$

$$my_i = y_i(\text{zoomfactor}),$$

where *zoomfactor* is the down-sampling factor for the browse image. The orientation of a line, l_{ij}^m , formed by the (i, j) pair of control points in the mosaic coordinate system is found as:

$$\theta_{ij}^m = \tan^{-1} \left(\frac{my_j - my_i}{mx_j - mx_i} \right).$$

The orientation of a line, l_{ij}^s , formed by the (i, j) pair of control points in the stage coordinate system is found as:

$$\theta_{ij}^s = \tan^{-1} \left(\frac{y'_j - y'_i}{x'_j - x'_i} \right).$$

The angle of rotation between the stage and mosaic coordinate systems is found as:

$$\alpha = \theta_{ij}^s - \theta_{ij}^m.$$

Once the parameters of rotational coordinate transformation have been determined, the linear translation is computed using a specific control point, say i , as:

$$a = x'_i - mx'_i \cos(\alpha) + my'_i \sin(\alpha),$$

$$b = y'_i - mx'_i \sin(\alpha) - my'_i \cos(\alpha).$$

With α , a , and b , the coordinates, (mp, mq) , of a point in the mosaic image could be transformed to stage coordinates as:

$$p = mp \cos(\alpha) + mq \sin(\alpha) + a ,$$

$$q = mp \sin(\alpha) - mq \cos(\alpha) + b .$$

Using this transformation, every pixel on the browse image could be mapped to a corresponding point in the stage coordinate system. It then becomes possible to translate mouse clicks on the browse image to stage motion commands such that the region of interest could be examined in the live image of the wood.

Chapter 6 SUMMARY

6.1 Summary

Tree ring analysis has applications in diverse fields ranging from archaeology to climatology, providing means to establish dates of environmental events. Dendrochronology aids in collecting and analyzing environmental data preserved in the samples of dead and living trees and is one of the very few methods of studying patterns of climatic fluctuation in the past. The present-day techniques of measuring tree ring widths and correlating data from different samples, though accurate enough to be used as a reference for Carbon-14 dating, are primarily manual methods and, hence, time-consuming. The TREES system[2] provides a computer-assisted platform to carry out tree ring analysis and is an outcome of the research being carried out jointly between the Digital Image Analysis Laboratory of the Electrical and Computer Engineering department and the Laboratory of Tree-Ring Research at the University of Arizona.

This thesis describes version 2.0 of the TREES system, with a primary focus on the computer vision issues involved in ring identification. The techniques and algorithms presented in this thesis have been tested as part of the initial TREES system (version 1.0) developed by Conner[2]. The TREES system framework consists of image acquisition and analysis components. The techniques presented here replace only the image analysis component of the TREES system v1.0 while retaining the image acquisition subsystem.

Chapter 2 presented a discussion of edge detection techniques in the TREES system v1.0 and its limitations. Noting that tree ring samples tend to be diverse in nature, we discussed the outcome of ring identification in the presence of wood anomalies and emphasized the need for an interactive environment. We addressed the edge fragmentation problem and showed that the modified Canny edge detection algorithm operating on a pixel-by-pixel basis is less suitable for analyst interaction. In Chapter 3, we discussed in detail the computer vision issues involved in the development of a flow-based edge detection algorithm that is suitable for tree-ring analysis. By considering tree-ring boundaries to be slow-varying features, we presented a method to extract the flow patterns and track the edges within them.

Chapter 4 provided a discussion of edge maps and chain codes. We discussed a technique for scaling edge maps represented by chain codes. Considering that the earlywood-latewood boundaries tend to be ill defined, we presented a technique for extracting them from the average grayscale profile. We also presented a method for extracting missed latewood-earlywood boundaries from average grayscale profile in a post-processing step. Chapter 5 discussed the interactive features in the TREES system with an emphasis on interactive edge linking. The Douglass method of cross-dating requires the ability to return to the wood sample and at the end of Chapter 5, we presented a technique for overcoming the hardware limitation in the TREES system to support wood-centered analysis.

6.2 Conclusions

The TREES system provides a semi-automated platform for tree-ring analysis that drastically reduces the time spent by researchers on tree-ring width acquisition. Version 2.0 of the system adds an interactive dimension to the image analysis system. The techniques presented in this thesis have been tested on a diverse set of conifer samples. The edge detection performance has been compared against the modified Canny edge detection algorithm using samples with wood anomalies. From the results obtained, the accuracy of edge detection has been verified visually, both against the image sample and against the edge map obtained from the modified Canny algorithm. The results obtained are promising despite the presence of resin ducts, prominent rays and other wood features. It is interesting to note that the flow-based edge detection algorithm operates significantly faster than the modified Canny algorithm owing to its ring-by-ring nature of operation. The system memory requirement is also considerably lower due to the elimination of the 5-odd connected-component labeling operations that were required earlier. This allows the TREES system to analyze samples that contain a large number of frames.

6.3 Scope for Future Contributions

The features of the TREES system make it a very promising tool that could find widespread use in the field of dendrochronology. At present, the system is constrained by its inability to process image samples captured outside the TREES environment. For example, an image sample obtained on-site from a digital camera would be unusable with

the TREES system. Since it is not practical to carry the TREES hardware setup to remote sites for image capture, the system has to be modified to accept and process samples captured outside of its environment. The modular design methodology implemented in the TREES system makes it fairly easy to incorporate such changes. The SADIE library implements and provides for functions to create mosaics from multiple individual frames. This feature could be made use of in the TREES system to allow the dendrochronologist to patch together frames obtained from sources such as digital cameras and scanners. In addition to allowing the TREES system to process such mosaics, it would also allow the image samples to be exchanged across multiple research laboratories for analysis.

The algorithms used in the current version of the TREES system are based on the assumption that the tree rings in an image sample run top-down, rather than sideward. While this assumption holds in most cases, there are exceptions. For example, a core sample obtained from a tree could have inclined rings to one end of the core. This happens when the tool used to extract the core misses the pith. In such cases, the current TREES system requires the core sample to be analyzed in sections, with the image capture setup in such a way that in each section, the rings in the mosaic run top-down. A more feasible solution would be to modify the algorithms such that the entire core sample could be analyzed in one mosaic, but with multiple regions of interest specified within the image sample. The analyst could select the regions in a manner that the rings within each region are aligned in the same direction. This would allow the TREES system to patch together the widths obtained from the individual regions to compose a single time series or skeleton plot.

REFERENCES

- [1] Canny, J. F., "A computational approach to edge detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol.8, no.2, pp. 679-698, Nov.1986.
- [2] Conner, W. S., "A Computer Vision based Tree-Ring Analysis and Dating System." Master's Thesis, University of Arizona, 1999.
- [3] Conner, W. S., Schowengerdt, R. A., Munro, M., and Hughes, M. K., "Design of a computer vision based tree ring dating system," *Proceedings of the 1998 IEEE Southwest Symposium on Image Analysis and Interpretation*, pp. 256-261, 5-7 Apr.1998.
- [4] Conner, W. S., Schowengerdt, R. A., Munro, M., and Hughes, M. K., "Engineering design of an image acquisition and analysis system for dendrochronology," *Optical Engineering*, vol.39, no.2, pp. 453-463, Feb.2000.
- [5] Douglass, A. E., "Tree rings and chronology," *University of Arizona Bulletin*, vol.8, no.4, Mar.1937.
- [6] Engle, J. B., "A Computer-Assisted Tree-Ring Chronology Composition System." Master's Thesis, University of Arizona, 2000.
- [7] Guay, R., Gagnon, R., and Morin, H., "MacDENDRO, a new automatic and interactive tree-ring measurement system based on image processing," *TREE Rings and Environment, Proceedings of the International Dendrochronological Symposium (LUNDQUA Report 34)*, Ystad, Sweden, pp. 128-131, 3-9 Sept.1990.
- [8] Guay, R., *WinDENDRO V.6.0 User's Guide*, Regent Instruments, Inc., Quebec, Canada, 1995.
- [9] Haralick, R. M., Shapiro, L. G., *Computer and Robot Vision - Volume I*, Reading, Mass.: Addison-Wesley, 1992.
- [10] Jain, R., Kasturi, R., and Schunck, B. G., *Machine Vision*, New York: McGraw-Hill, 1995
- [11] Kasaei, S., Deriche, M., and Boashash, B., "Fingerprint feature extraction using block-direction on reconstructed images," *TENCON '97, Proceedings of the IEEE*

Region 10 Annual Conference on Speech and Image Technologies for Computing and Telecommunications, vol.1, pp.303-306, 2-4 Dec.1997.

- [12] Wolberg, G., *Digital Image Warping*, IEEE Computer Society Press, 1992, pp. 48-49.
- [13] Lim, J. S., *Two-Dimensional Signal and Image Processing*, Upper Saddle River, New Jersey: Prentice Hall, 1990.
- [14] Morsy, K. A., Kanayama, Y., "A new straight edge detection algorithm using direction-controlled edge tracking and random hitting," *CIRA '97, Proceedings of the 1997 IEEE International Symposium on Computational Intelligence in Robotics and Automation*, pp. 398-405, 10-11 Jul.1997.
- [15] Ousterhout, J., *Tcl and the Tk Toolkit*, Reading, Mass.: Addison-Wesley, Mar.1994.
- [16] Schou J., and Rytter, E., "Dendrochronological dating using scanning and image processing," *TREE Rings and Environment, Proceedings of the International Dendrochronological Symposium (LUNDQUA Report 34)*, Ystad, Sweden, pp. 286-287, 3-9 Sept.1990.
- [17] Sheppard, P. R., Graumlich, L. J., "A reflected-light video imaging system for tree-ring analysis of conifers," *Tree Rings, Environment and Humanity: Proceedings of the International Conference, RADIOCARBON 1996*, Tucson, Arizona, pp. 879-889, 1996.
- [18] Sheppard, P. R., Graumlich, L. J., Conkey, L. E., "Reflected-light image analysis of conifer tree rings for reconstructing climate," *The Holocene*, vol.6, no.1, pp. 62-68, 1996.
- [19] Sheppard, P. R., "Reflected-Light Image Analysis of Conifer Tree Rings for Dendrochronological Research." Dissertation, University of Arizona, 1995.
- [20] Thetford, R. D., D'Arrigo, R. D., and Jacoby, G. C., "An image analysis system for determining densitometric and ring-width time series," *Canadian Journal of Forest Research*, vol.21, pp.1544-1549, 1991.
- [21] Wang, Y., Mitra, S. K., "Edge detection based on orientation distribution of gradient images," *ICASSP '91, Proceedings of the 1991 International Conference on Acoustics, Speech, and Signal Processing*, Ontario, Canada, vol.4, pp. 2569-2572, 14-17 Apr.1991.

- [22] Welch, B. B., *Practical Programming in Tcl and Tk*, 2d ed. Upper Saddle River, New Jersey: Prentice Hall, 1997.