Automated Planning and Optimization of Lumber Production Using Machine Vision and Computed Tomography

Suchendra M. Bhandarkar, *Member, IEEE,* Xingzhi Luo, *Student Member, IEEE,*
Richard F. Daniels and E. William Tollner

***Abstract*—An automated system for planning and optimization of lumber production using Machine Vision and Computed Tomography (CT) is proposed. Cross-sectional CT images of hardwood logs are analyzed using Machine Vision algorithms. Internal defects in the hardwood logs pockets are identified and localized. A virtual *in silico* 3-D reconstruction of the hardwood log and its internal defects is generated using Kalman filter-based tracking algorithms. Various sawing operations are simulated on the virtual 3-D reconstruction of the log and the resulting virtual lumber products automatically graded using rules stipulated by the National Hardwood Lumber Association (NHLA). Knowledge of the internal log defects is suitably exploited to formulate sawing strategies that optimize the value yield recovery of the resulting lumber products. A prototype implementation shows significant gains in value yield recovery when compared to lumber processing strategies that use only the information derived from the external log structure. The system is intended as a decision aid for lumber production planning and an interactive training tool for novice sawyers and machinists in the lumber industry.**

***Index Terms*—Automated Lumber Production, Lumber Production Optimization, Non-destructive Evaluation, Automated Lumber Grading, Computed Tomography.**

I. Introduction

The value of hardwood lumber is determined by the quantity, size and types of internal log defects such as knots, cracks, decay and other anomalies of tree growth that eventually appear on the lumber surfaces. Depending on the nature of the end utilization, each log is sawed to minimize the presence of these internal defects on the resulting lumber surfaces. In order to achieve this goal, the internal defects within the log must be accurately identified and localized prior to the sawing of the log. The knowledge of the nature and positions of the internal log defects must then be exploited to determine a lumber production strategy that maximizes the value and yield of the resulting lumber product(s). In most sawmills, however, logs are processed into lumber based solely on external log inspection and knowledge of lumber grades with little or no information about the internal log defects and with inaccurate or incomplete geometric log data. This adversely affects the accuracy of the lumber processing, resulting in suboptimal lumber production where the potential value of logs is wasted.

Production of lumber is essentially a destructive and hence irreversible process; any loss in value yield due to incorrect or suboptimal sawing is irrevocable. Given the low conversion efficiency of about 35% for conventional sawmills [16] and with the rising costs of hardwood logs accounting for over 80% of total production costs [17], improving the lumber value yield from hardwood logs has become imperative for many sawmills. Given the inherent limitations of external log inspection, it is reasonable to assume that future gains in lumber value yield will be achieved only by internal log scanning [7], [13], [17], [34].

Identification and localization of internal log defects are estimated to lead to potential gains of about 15%–18% in lumber value [7]. This represents a savings of over $2 billion for the hardwood lumber industry in the United States [17], [25]. Forest products–based economies are increasingly dependent on getting the highest-value wood products from a declining forest resource base. This results in disproportionate harvesting pressure on high-demand hardwood species such as Hard Maple, Black Walnut, White Ash and Red Oak that exhibit large differences in value between the highest and lowest lumber grades [33]. Environmental concerns and the ecological need for maintaining biodiversity in forest ecosystems underscores the need to utilize as many hardwood species for wood products as possible, to improve the efficiency in converting low-grade logs into high-value lumber products, to reduce unnecessary wastage and to conserve valuable hardwood forest resources. One way of achieving these goals is by identification and localization of internal defects in hardwood logs and using this information to optimize the processing of the resulting lumber.

Studies of computed axial tomography (CAT or CT) and magnetic resonance imaging (MRI) (also known as nuclear magnetic resonance (NMR) imaging) for internal log defects [1], [2], [6], [13], [34] have demonstrated that the CT and MRI technologies available today can be used successfully to image the internal features of logs. On account of their inherent sensitivity to the water content of the imaged sample, MRI techniques are particularly well suited for detecting internal features of logs, such as knots, reaction wood, wetwood, and gum spots, that are characterized by varying moisture

This work was supported in part by the US Department of Agriculture through an NRICGP grant (Award Number: 2001-35103-10049).

S.M. Bhandarkar and X. Luo are with the Department of Computer Science, The University of Georgia, Athens, GA 30602-7404, USA (e-mails: suchi@cs.uga.edu, xingzhi@cs.uga.edu).

R.F. Daniels is with the Warnell School of Forest Resources, The University of Georgia, Athens, GA 30602-2152, USA (e-mail: ddaniels@smokey.forestry.uga.edu).

E.W. Tollner is with the Department of Biological and Agricultural Engineering, The University of Georgia, Athens, GA 30602-4435, USA (e-mail: btollner@engr.uga.edu).

content in the underlying wood [6]. In CT images, on the other hand, the grayscale value of a pixel is directly proportional to the x-ray absorption which is then correlated with the material density at the pixel location [13]. Knots and moisture pockets are noted to have higher material density and/or higher moisture content than surrounding clear wood and are often characterized by pixels with very high grayscale values in the CT image. Holes, cracks and decay pockets are void areas filled with air or decayed wood and hence characterized by low material density, resulting in corresponding CT image pixels with very low grayscale values. Holes and decay pockets usually have circular cross-sections and are typically short in length whereas cracks are usually thin and long.

Although MRI is a more recent innovation, solid state CT scanners, capable of scanning rates close to 30 slices per second, are fast approaching the speed necessary for real-time production use in sawmills [17]. However, the computational methods for analyzing the CT images for internal defects reliably and in real time, and exploiting the knowledge of the internal defects to determine optimal lumber processing strategies are a challenging research topic and the subject of this paper.

In this paper we describe the design and implementation of a machine vision system for the automated planning and optimization of lumber production from hardwood logs. The paper makes two significant contributions. First, a Kalman filter-based feature tracking framework is proposed to enable simultaneous detection, localization and 3-D reconstruction of internal log defects in a manner that is computationally much more efficient than existing approaches. Second, detailed mathematical models and algorithms for lumber production optimization are proposed which exploit the knowledge of the internal log defects to maximize the value yield recovery of the resulting lumber products. The system is intended as a decision aid for lumber production planning but could also be used as an interactive training tool for novice sawyers and machinists, allowing them to practice various sawing strategies on *virtual* logs before working on real logs.

The remainder of the paper is organized as follows. Section II provides a brief review of previous work. Section III provides an overview of the proposed system. Section IV provides a detailed scheme for the detection, identification and localization of internal defects such as holes, knots and cracks in a single CT image slice using a combination of structural and spectral features. Section V describes the 3-D reconstruction of these internal defects using Kalman filter-based tracking algorithms. Section VI describes algorithms for determination of optimal lumber production strategies. Section VII presents experimental results on real CT image data from hardwood logs. Section VIII concludes the paper with an outline for future work.

II. Brief Literature Review

Techniques for internal defect identification and classification in cross-sectional CT images of logs include gray-level thresholding and binarization [13], [34], neural network-based classification [29], integration of shape and texture features with pixel grayscale values [1], [2], [5] and Dempster–Schafer theory-based evidential reasoning on the 3–D geometric features of the defects [36]. Bhandarkar *et al.* [1], [2] and Samson [26] present geometrical modeling algorithms to describe the structure of internal defects within the logs and their appearance on the surfaces of the lumber beams sawn from those logs, and to compute the effect of the presence of knots in the conversion of logs into structural lumber [2], [27]. The results of internal defect identification and localization can be used to reconstruct a 3-D model of the log along with its internal defects [1], [2]. Software programs that simulate various machining operations such as sawing and veneering on the virtual 3-D log reconstruction have been described in the literature [1], [2], [8], [15], [23], [28]. Likewise, programs for automated grading of virtually produced lumber (using the aforementioned simulators) and physical lumber produced by sawmills have also been reported in the literature [20], [21]. These programs have been used to estimate the improvements in lumber value yield recovery resulting from internal log scanning which are reported to be in the range of 40%–60% [2], [7], [15]. The above review covers the most relevant developments in the use of CT and MRI technologies for internal defect detection and identification in hardwood logs. The review paper by Pham and Alcock [24] covers the wider area of automated visual inspection of logs and lumber and includes both external scanning using optical sensors and internal scanning using CT and MRI technologies.

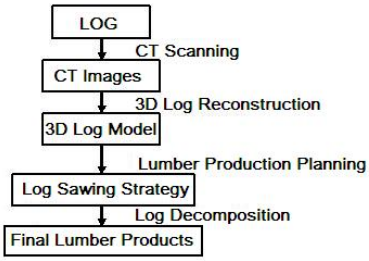


Fig. 1. System overview

III. Overview of the Proposed System

The proposed system (Figure 1) for planning and optimization of lumber production consists of the following subsystems: (a) CT log scanning (b) Internal defect identification and localization, (c) 3-D virtual reconstruction of the log and its internal defects, and, (d) Determination of optimal lumber production strategy.

The CT log scanning subsystem consists of a Toshiba TCT 20AX CT scanner with a pixel resolution of 0.75mm by 0.75mm, an intensity resolution of 8 bits per pixel (i.e., 256 gray levels) and an image size of 316×316 pixels. The scanning of a 4 meter log typically results in 224 cross-sectional CT images. Although the above CT scanner is dated compared to its counterparts used for medical imaging, it was deemed adequate for the design and development of a proof-of-concept prototype system for lumber production planning and optimization.

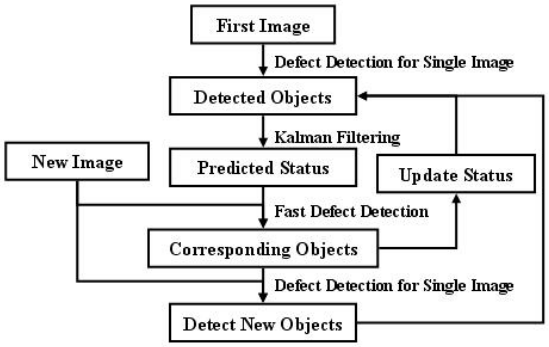


Fig. 2. The flowchart of the defect detection system

The overall flowchart for the proposed defect detection subsystem is given in Figure 2. In the proposed subsystem, both structural (shape) and spectral (grayscale) features are incorporated in the detection and localization of internal log defects in a single CT image slice. This addresses the limitations of conventional pixel-level thresholding or binarization methods which are limited in their classification accuracy, especially when confronted with overlapping pixel grayscale values from different defect classes, in spite of fairly sophisticated analysis of the grayscale histogram [8]. Furthermore, in the proposed scheme, the processes of detection, localization and 3-D reconstruction of internal defects are integrated across multiple CT image slices within a single Kalman filter-based feature tracking framework. This is in contrast to most existing techniques where the detection, identification and localization of internal log defects are performed independently in each CT or MR image slice and the 3-D reconstruction of the defect is achieved via simple correspondence analysis across multiple image slices using the defect shape, size and location information [1], [2]. As a result, existing techniques are computationally inefficient and potentially unsuitable for real-time applications since the spatial coherence of the defects along the axial direction is not exploited. They are also error-prone, especially when dealing with defects with complex 3-D shapes, since only two successive CT image slices are typically used to establish the correspondence.

In the proposed scheme, a Kalman filter [18] is used to track the defect parameters continuously from one CT image slice to the next by predicting the locations of the defects in successive slices. The algorithms for internal defect detection and localization within a single CT image slice are used to initialize the Kalman filter-based tracking algorithm. After a defect is detected and localized within a CT image slice, the tracking algorithm is used to detect and localize the defect in successive CT image slices and also reconstruct its geometry in 3-D space. The net result is much faster detection, identification, localization and reconstruction of the internal log defects since only a *local* search within a fairly small neighborhood of the predicted defect locations is entailed.

Given the 3-D reconstruction of the log and its internal defects, the final subsystem determines the optimal lumber production strategy; one that maximizes the yield and grade of the resulting lumber product. Mathematical models for commonly performed sawing operations such as live sawing, cant sawing, grade sawing and secondary sawing are proposed. The optimal sawing strategy is determined by searching the parameter space of the sawing models using a dynamic programming algorithm. The dynamic programming algorithm determines the optimal spacings and orientations of the sawing planes in order to maximize the value and yield of the resulting lumber products. A heuristic search algorithm is designed and shown to speed up the optimization process while providing a solution that is acceptably close to the optimal result. An automated lumber grading program that is compliant with the National Hardwood Lumber Association (NHLA) grading rules is designed to determine the grades of the (virtual) lumber products resulting from the simulation of the various aforementioned sawing operations on the virtual 3-D log reconstruction.

IV. Detection of Defects in a Single CT Image

Three major categories of defects are detected within a single CT image slice, i.e., knots, holes and cracks, since they are known to greatly impact the grade and value of the resulting lumber.

A. Detection of Knots and the Outer Log Boundary

Knots are detected via analysis of the pixel grayscale intensity in a local window surrounding the pixel as follows:

$$K(i,j) = \begin{cases} 1 & \text{if } \frac{1}{M^2} \sum_{(x,y)\in W(i,j)} F(x,y) > T_k \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

where $K(i,j)$ is the resulting binary image delineating the knots, $W(i,j)$ is a window of predetermined size $M \times M$ centered at pixel (i,j), $F(i,j)$ is a CT image slice of the log and T_k is a predetermined threshold. A CT image slice containing a knot is shown in Figure 3(a) whereas the binary image resulting from the application of equation (1) is shown in Figure 3(b). A comparison of Figure 3(a) and Figure 3(b) shows that the application of equation (1) results in the erosion of the knot boundaries. Consequently, a morphological dilation operation [30] is used to recover the knot boundaries as shown in Figure 3(c). The outer boundary of the log cross-section in a single CT image slice is also detected using the binarization technique given in equation (1); however with a smaller window size (3×3 is typically adequate) and a smaller threshold value (which is chosen to be slightly larger than the grayscale value of a void area in the CT image).

B. Detection of Holes

Holes are actual void areas and appear as dark regions in the CT images with graylevels similar to those of the background. A simple thresholding scheme outlined in equation (2) is used to classify the CT image pixels as holes.

$$K(i,j) = \begin{cases} 1 & \text{if } F(i,j) < T_h \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

where T_h is an empirically determined threshold value. Since holes are usually small in size and approximately round

(a) (b) (c)

Fig. 3. Result of knot detection: (a) CT image slice containing a knot, (b) Result of analysis of local graylevel pixel density, and (c) Extracted knots after dilation

(a) (b) (c) (d)

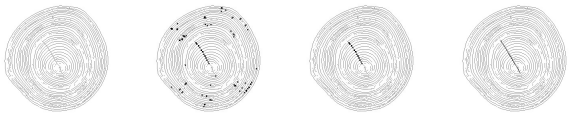
Fig. 4. Result of hole detection: (a) Input CT image, (b) Thresholded image, (c) Removal of cracks and valleys using erosion, and (d) Restoration of holes using dilation

in shape, false holes, typically caused by small cracks or grayscale valleys between successive rings, are removed by using a combination of morphological erosion and dilation operations on the thresholded result [30]. Figure 4 depicts the results of the various stages of hole detection.

C. Detection of Cracks

A crack, in a CT image slice, is usually long and thin. A straightforward grayscale-based binarization of the CT image results in either fragmentation of the detected cracks or too many misclassifications of the regions denoting the grayscale valleys between the annular rings as cracks. Since both, the grayscale valleys and cracks are narrow and long, the grayscale density-based binarization technique (equation (1)) is not able to separate them. However, cracks are typically perpendicular to the grayscale valleys and the local direction of a grayscale valley can be estimated by approximating the grayscale valleys by concentric circles centered at the centroid of the log cross-section. This property is exploited in the crack detection scheme.

The crack detection scheme is summarized as follows. The local linear structures resulting from cracks and valleys are first detected using Sobel-like edge operators and then skeletonized using an edge thinning algorithm [10] as depicted in Figure 5(a). The points of intersection between the lines defining the cracks and the lines defining the local structure of the valleys are represented as *fork* points within a local window (Figure 5(b)). Fork points that are distributed along the same crack-like feature are grouped using a greedy clustering algorithm that exploits spatial connectivity and proximity (Figure 5(c)). A RANSAC-based line fitting algorithm [11] is used to determine the line segment characterizing a group of fork points (Figure 5(d)). Crack-like features which are observed to be parallel (within a certain angular threshold) to the local structure of the valleys are deemed to be spurious and discarded. Given the parameters of the fitted line segment, the actual crack pixels are determined using an iterative depth-first search procedure detailed in [3].



(a) (b) (c) (d)

Fig. 5. Result of crack detection: (a) Binary image resulting from the Sobel-like edge operators and edge thinning, (b) Result of fork detection, (c) Result of fork grouping, and (d) Result of the RANSAC-based line fitting procedure

V. 3-D Defect Reconstruction Using the Kalman Filter

Since internal log defects are observed to exhibit spatial coherence across several successive CT image slices, it is possible to design computationally efficient algorithms for the detection of knots, holes and cracks that take advantage of the defect attribute values predicted by the Kalman filter [18]. Since knots, holes and cracks exhibit very different geometrical attributes in a 2-D image slice, different Kalman filtering models are proposed for each defect class. In addition, the outer boundary of the log cross-section is also detected and tracked across successive CT image slices in order to virtually reconstruct the entire log in 3-D space. The result is an integrated and computationally efficient defect extraction and 3-D defect reconstruction procedure that precludes the need to perform correspondence analysis independently for each pair of successive image slices.

A. 3-D Reconstruction of Knots and the Exterior Log Surface

The knots and the outer boundary of the log cross-section in a single CT image slice are simply encoded by using the positions of their centroids and enumerating the pixels on their respective bounding contours. However, the raw contour pixels are not used directly as the tracked/predicted variables in the Kalman filter model since the number of contour pixels varies significantly from one CT image slice to the next and it is computationally inefficient to use too many tracked/predicted variables in the Kalman filter. Thus, for the purpose of tracking, a knot defect or the exterior log boundary in a single CT slice is encoded by its convex hull. The convex hull is defined by a small number of points and computed using Graham's algorithm [14]. A B-spline contour approximation algorithm [12] is then used to determine the control points of the convex hull. The control points are used as the tracking parameters in the Kalman filtering model. The Kalman filtering model for the reconstruction of knots is depicted in Figure 6 and is described as follows.

1) Initially, a knot defect area is detected in a single CT image as described in Section IV-A.
2) The contour of the knot defect is extracted and its convex hull computed using Graham's algorithm [14]. M B-spline control points are then used to approximate the convex hull.

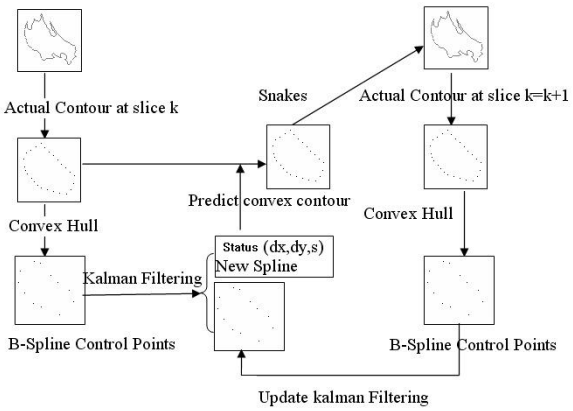


Fig. 6. An outline of the Kalman–Snakes based tracking method

3) The Kalman filter is applied to predict the *velocity* of the knot defect, where the term *velocity* denotes the rate at which the shape of the knot changes across successive CT image slices. Let d_x and d_y denote the velocity components along the x axis and y axis respectively. Let s denote the scale parameter in the time interval $[t, t+1]$ such that $s = 0$ denotes the fact that the object size is unchanged in the time interval $[t, t+1]$, $s > 0$ denotes that the object size has increased and $s < 0$ denotes that the object size has shrunk in the same time interval. If at time t, the centroid of the convex hull is given by (c_x, c_y), and the velocity by (d_x, d_y, s), then for a point (x, y) on the convex hull at time t, its position (x', y') at time $t+1$ can be computed using equation (3).

$$\begin{aligned} x' &= x + d_x + s(x - c_x) \\ y' &= y + d_y + s(y - c_y) \end{aligned} \tag{3}$$

4) The predicted velocity (d_x, d_y, s) is used to estimate the new position of the predicted convex contour using equation (3). The updated convex contour is obtained by using the predicted convex contour to initialize the *Snakes* contour fitting algorithm [19]. The Snakes contour fitting algorithm is used to search for the actual boundary of the knot in the new CT image slice.
5) Steps 2, 3 and 4 are repeated until all the CT image slices are processed or the defect is too small in size to be classified as a knot.

Since the above algorithm uses a combination of Kalman filter-based prediction and Snakes contour fitting, it is termed as the *Kalman–Snakes* algorithm [35]. The Kalman–Snakes algorithm for reconstruction of the exterior log surface is similar, except for some differences in how the convex hull points are generated. The detailed algorithms for convex hull generation, B-spline surface approximation and Kalman filter-based prediction are given in [4].

B. 3-D Construction of Holes and Cracks

The Kalman filter-based model for 3-D reconstruction of cracks and holes is given by the following equations:

$$\hat{\xi}_{k+1}^{-} = \hat{\xi}_{k}^{+} + q_k \tag{4}$$

$$Z_k = \hat{\xi}_{k}^{-} + v_k \tag{5}$$

where $\hat{\xi}$ is the tracking variable, Z_k the measured value of the tracking variable, q_k the random additive noise in the velocity transition function which is modeled as zero-mean stationary Gaussian white noise with distribution $\mathcal{N}(0, Q_0)$ and v_k the random additive noise in the velocity measurement function which is also modeled as zero-mean stationary Gaussian white noise with distribution $\mathcal{N}(0, R_0)$.

The contour of a hole is encoded by its bounding rectangle. The velocity of the center point of the rectangle is used as the tracking variable ($\xi = (\xi_x, \xi_y)^T$) for holes. Given a prediction $\hat{\xi}_{k+1}^{-}$, the new center point of the hole is computed. A rectangular image region centered on the predicted center point is then subject to binarization using equation (2) and the new location of the hole is obtained exactly using the technique for hole detection in a single CT image slice. Thus, the 3-D reconstruction of a hole is achieved by tracking the hole region across successive CT image slices.

For continuous crack detection, the CT images are first binarized using the scheme described in [3]. A crack defect is represented by the parameters $(\rho, \theta, x_0, y_0, l)$ where (x_0, y_0) is the crack center and l the length of the line segment describing the crack. The equation of the line describing the crack is given by $x\cos\theta + y\sin\theta = \rho$ where θ is the orientation angle of the line and ρ the perpendicular distance of the line from the origin. The orientation angle θ is used as the tracking variable ξ in the Kalman filter-based tracking model for a crack. After a crack is detected and extracted in the previous CT image slices, its orientation angle in succeeding CT image slices is predicted using equation (4). A fast crack localization scheme detailed in [4] is used to extract the crack in succeeding CT image slices using the predicted orientation.

Note that in the above Kalman filter-based tracking approach, it is not necessary to perform all the steps involved in defect detection and localization in subsequent CT image slices once they have been performed for the first CT image slice in the image sequence. The defects that are detected and localized are removed from further consideration. The procedures for defect detection and localization in a single CT image slice (i.e., without defect tracking) are applied to the remainder of the image to search for new defects.

C. Removal of False Defects and Insertion of Missing Defects

Spatial coherence is exploited to remove false defects and account for missing defects in the CT image slices. If a defect is detected in only one CT image slice, and no corresponding defect is detected in k previous or k succeeding CT image slices, where k is a predetermined threshold (in our case $k = 2$), then the defect is deemed to have been caused by random noise and is removed from further consideration. Likewise, when the size of a defect is small, it is possible that it is not detected in a single CT image slice. If a defect is detected in CT image slices $i-1$ and $i+1$ but not in CT image slice i, then it is necessary to verify whether or not the defect

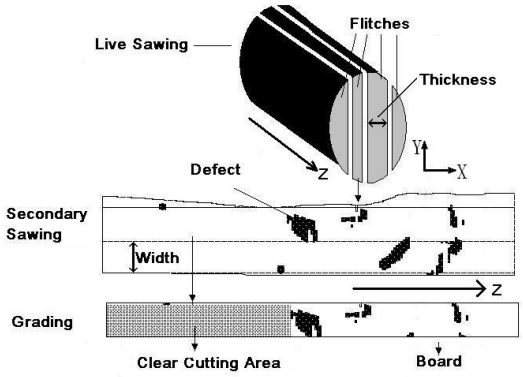


Fig. 7. A typical lumber production system

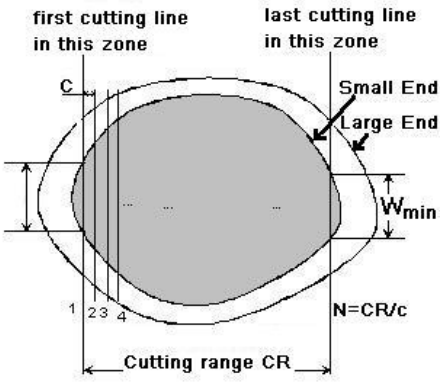


Fig. 8. Cutting range and sawing planes for live sawing

exists in CT image slice i. The approximate position of the defect in CT image slice i is computed via linear interpolation between its positions in CT image slices $i-1$ and $i+1$. The existence of the defect at the interpolated position in the CT image slice is verified by using the defect detection and localization procedure for a single CT image slice (i.e., without tracking) but with relaxed threshold values.

VI. Mathematical Models For Lumber Production Optimization

Lumber production is essentially a decomposition process where the log is broken down to yield the desired lumber products. Given a 3-D model of the log, different sawing schemes can be used to decompose the log. Live sawing, cant sawing and grade sawing are primary sawing schemes [16] used to cut a log into *flitches* along sawing planes parallel to the axis of the 3-D cylindrical abstraction (i.e., the Z axis in our case) of the log (Figure 7). Secondary sawing [16] is used to further refine the flitches into higher quality *boards* (Figure 7). Given a board or a flitch, an automated grading system is used to determine its value based on the grading rules stipulated by the National Hardwood Lumber Association (NHLA) [22]. The grading system determines the value of a flitch or board by classifying it into different grades based on its quality which is determined by the type and size of defects that appear on the flitch or board surface. The size of the clear cutting (i.e., defect-free) areas and the size of the surface defects are two important factors that determine the grade of a flitch or board. The three dimensions of a board or flitch are termed as its length, width and thickness as illustrated in Figure 7. The goal of a computer-aided lumber production planning system is to determine the optimal sawing pattern in an appropriately chosen parameter space such that the value of the lumber products obtained from the log is maximized [31], [32].

A. Live Sawing

Consider a 3-D coordinate system associated with the 3-D cylindrical abstraction of a log such that the axis of the cylinder is parallel to the Z axis and the log cross-sections lie in the XY plane. Live sawing is a straightforward sawing method where, for a given initial orientation of the saw in the XY plane, the log is cut into flitches using sawing planes that parallel to each other and parallel to the Z axis. Different initial orientations of the saw in the XY plane result in different values for the resulting lumber products. The goal in formulating an optimal live sawing strategy is to determine the best initial saw orientation in the XY plane and the optimal spacings between the mutually parallel sawing planes so as to maximize the value of the resulting lumber products. In the context of this paper, the term *lumber* or *lumber product* denotes resulting flitches or boards depending on whether the sawing technique under discussion is a primary sawing technique or a secondary sawing technique respectively.

1) Formalization of Live Sawing: The variables used in the formalization of live sawing are defined below and illustrated in Figure 8. Figure 8 shows the projection of a cylindrical log on the XY plane. The outer (almost) elliptical boundary denotes the projection of the larger cross-sectional end of the log on the XY plane whereas the inner (almost) elliptical boundary denotes the projection of smaller cross-sectional end of the log on the XY plane. Thus the log is modeled as a cylinder with a monotonically tapering cross-section from one end to the other. Note that this subsumes the special case of a cylinder with uniform cross-section.

- $T = \{T_1, T_2, ..., T_t\}$: a finite set of allowable lumber thickness values in mm.
- t_{min}: the minimum of the elements in T.
- $W = \{W_1, W_2, ..., W_w\}$: a finite set of lumber width values in mm.
- w_{min}: the minimum of the elements in W.
- c: cutting plane resolution (smallest separation between two successive sawing planes) in mm.
- K: the kerf of the sawing planes in mm. The kerf denotes the finite thickness of the saw.
- CR: the cutting range in mm.
- $N = \lfloor CR/c \rfloor$ the number of possible sawing planes that can be accommodated within the sawing range.
- θ: The initial orientation of the sawing plane as measured in the XY plane (Figure 10). The value of $\theta \in [0^o, 180^o)$.

It is assumed that the allowable lumber thickness values T_i and the sawing plane resolution c are integers. The sawing range and other important parameters of the sawing surface in the context of live sawing are computed as follows:

Determination of the Cutting Range: Given the initial orientation θ of the sawing plane, the cutting range for a log is first determined. Note that a valid piece of lumber must have a minimum width of w_{min}. Thus the cutting range is determined by searching for the start and end locations of the sawing plane that results in a piece of lumber with valid width. Since the outer boundary of the smaller cross-sectional end of the log is known, the cutting range for a given initial orientation θ of the sawing plane is determined by scanning from the cross-sectional boundary towards the cross-sectional centroid from two opposite ends as illustrated in Figure 8. A pair of sawing planes within this range produces a piece of lumber with a valid width value and prespecified thickness value. To simplify the analysis, it is assumed that the possible positions of the sawing plane are discrete in steps of c mm (Figure 8). The value of c is assumed to be an integer and a reasonable value for c is 1 mm. $N = \lfloor CR/c \rfloor$ represents the number of possible sawing planes that can be accommodated within the cutting range. The possible cutting planes are enumerated as 1, 2, ..., N. It can be shown that the first sawing plane must be one of sawing planes between 1 and $\lceil t_{min}/c \rceil$, whereas the last sawing plane must be one of sawing planes between $N - \lceil t_{min}/c \rceil$ and N.

Cutting surface: For a given sawing plane, the appearance of the resulting lumber surface (determined by the sizes and distribution of the surface defects) determines the quality and price of the resulting lumber product. The lumber surface data needed to evaluate the appearance of the lumber surface include the types, boundaries and sizes of the surface defects and the outer log boundary.

Mathematical model for optimal live sawing: Inputs to the optimal live sawing determination algorithm include the sawing orientation θ, T, W, various lumber price factors, and the log data which includes the types, sizes, positions and orientations of the internal defects. The mathematical model for optimal live sawing can be simply represented by the objective function:

$$F_{live}(log) = (\Theta^*(log), S^*(\Theta^*), V(S^*)) \tag{6}$$

where $\Theta^*(log)$ is the optimal sawing orientation for a given log, $S^*(\Theta^*)$ is the optimal sawing pattern (determined by the locations of the sawing planes) corresponding to the optimal sawing orientation $\Theta^*(log)$ and $V(S^*)$ is the maximum value of the lumber boards resulting from the optimal sawing pattern $S^*(\Theta^*)$ associated with the optimal sawing orientation $\Theta^*(log)$.

For a given value of θ, the cutting range $CR(\theta)$ can be determined as discussed earlier. Let $C(\theta) = \{1, 2, ..., N\}$ be a finite set of all possible positions of the sawing planes. $S^*(\theta)$ is the optimal sawing pattern for the given sawing orientation θ. $V(S(\theta))$ is the value of the resulting lumber (obtained using an automated lumber grading algorithm) for a given the sawing pattern $S(\theta)$. A sawing pattern $S(\theta) = \{s_0, s_1, ..., s_n\}$ is a subset of $C(\theta)$ that satisfies the following constraints:

$$(s_i - s_{i-1} - \lceil K/c \rceil) \cdot c \in T, for\ 1 \le i \le n \tag{7}$$

$$1 \le s_0 \le \lceil t_{min}/c \rceil \tag{8}$$

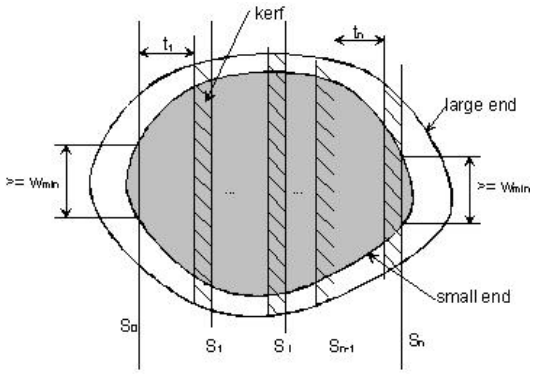


Fig. 9. A feasible solution for live sawing

$$N - \lfloor t_{min}/c \rfloor \le s_n \le N \tag{9}$$

A sawing pattern $S(\theta)$ that satisfies constraints (7)–(9) is a deemed to be a feasible sawing pattern for the log (Figure 9). Suppose the lumber generated by the sawing planes s_{i-1} and s_i is of value v_i. Let $\mathbf{v} = (v_1, v_2, ..., v_n)$ be an n-vector describing the values of the lumber pieces generated by the sequence of sawing planes $S(\theta) = \{s_0, s_1, \ldots, s_n\}$. For $S(\theta) \subset C(\theta)$, define $V(S(\theta)) = \sum_{j \in S(\theta)} v_j$ to be the total lumber value associated with the sawing pattern $S(\theta)$. In practical situations, constraint (7) is relaxed since forcing a lumber surface containing lots of defects into the final lumber product may result in lowering its overall value. Thus constraint (7) is relaxed to the following pair of constraints:

$$(s_i - s_{i-1} - \lceil K/c \rceil) \cdot c \in T, \text{for } 1 \le i \le n \tag{10}$$

or

$$(s_i - s_{i-1} - \lceil K/c \rceil) \cdot c < t_{min} \tag{11}$$

Thus, when constraint (11) is satisfied, a portion of the lumber between sawing surfaces s_{i-1} and s_i is omitted from the final lumber product without lowering its value. Suppose $\zeta(\theta)$ is a collection of subsets of $C(\theta)$, wherein each subset satisfies the sawing constraints (8) and (9) and, either constraint (10) or constraint (11). Then the problem of determining the optimal sawing pattern is one of combinatorial optimization given by:

$$S^*(\theta) = \arg(\max\{v(S(\theta)) : S(\theta) \in \zeta(\theta)\}) \tag{12}$$

This combinatorial optimization problem can be easily solved using a dynamic programming algorithm [9].

The Dynamic Programming Algorithm: Function $s^*(i)$ is defined as the optimal sawing pattern for the portion of the log between sawing planes 1 and i and $v^*(i)$ is the corresponding optimal value of the lumber produced. Let $g(i, j)$ be the value of the lumber by the sawing planes i through j. It is obvious that $v^*(i) \ge v^*(i-1)$ and $v^*(k) = 0$, $s^*(k) = \Phi$ (where Φ is the empty set) for all $k < t_{min}/c$. When the values of $v^*(k)$ and $s^*(k)$ for all $k \le i$ are known, then

$$\begin{aligned} v^*(i+1) = \max_{j \in [0,t]} (\quad & (v^*(i+1-T_j/c - \lceil K/c \rceil) \\ + \quad & g(i+1-T_j/c, i+1))) \end{aligned} \tag{13}$$

where $\lceil K/c \rceil$, the kerf size, is the minimum gap between two sawing surfaces. Note that $T_0 = 0$ when $j = 0$. This means that when $j = 0$, then the additional piece of lumber defined by the sawing surfaces $(i, i+1)$ does not increase the maximum value of the overall generated lumber at point $i+1$. Thus this piece of lumber is simply ignored, but may be reconsidered at a later point in the optimization process. The relaxation of constraint (7) to constraint (10) or constraint (11) makes it possible to discard some portion of the lumber that contains too many defects. Equation (13) in conjunction with constraints (8), (9) and either constraint (10) or constraint (11) results in the standard dynamic programming algorithm. Suppose j^* results in the optimal value of $v^*(i+1)$, then

$$s^*(i+1) = s^*(i+1-T_{j^*}/c - K/c^*) \cup \{i+1-T_{j^*}/c^*\} \quad (14)$$

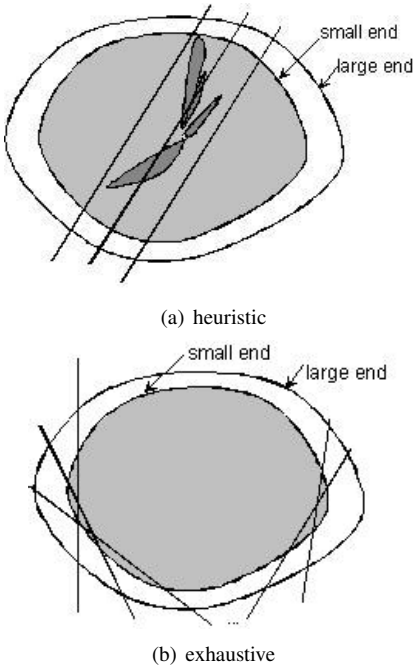
In summary, the dynamic programming algorithm first generates the initial values $v^*(k)$ for $k \leq T_{min}/c$ where $v^*(k) = 0$, $s^*(k) = \Phi$ for $k < T_{min}/c$ and $v^*(k) = g(1,k)$,$s^*(k) = \{k\}$ for $k = T_{min}/c$. Thereafter for each $i > T_{min}/c$, equations (13) and (14) are used iteratively to update $v^*(i)$ and $s^*(i)$ for all values of i. Finally, $s^*(N)$ and $v^*(N)$ correspond to $S^*(\theta)$ and $V(S^*(\theta))$ in the objective function F_{live} for a given value of the sawing orientation θ.

2) Live Sawing Algorithm: In the previous section, a dynamic programming algorithm was designed to determine $S^*(\theta)$ and $V^*(\theta)$ for a given value of the sawing orientation θ. However, to optimize the objective function $F_{live}(log)$, the optimal sawing orientation Θ^* also needs to be determined. A reasonable heuristic would be to examine the major axes of all the various internal log defects and if the internal defects are observed to share a common major axis (within a prespecified error threshold) then the sawing orientation is chosen to be along the common major axis (Figure 10(a)). This heuristic tends to cluster the surface defects on a few lumber surfaces. Alternatively, one can perform an exhaustive search for Θ^* in $[0, 180)$ in discrete steps where the step size is determined by the angular resolution of the saw (typically 2° or 4°) as shown in Figure 10(b).

The exhaustive search method to optimize the objective function, $F_{live}(log) = (\Theta^*(log), S^*(\Theta^*), V(S^*))$ can be summarized as:

1) For each sawing orientation $\theta \in [0, 180)$
   a) Determine the cutting range CR. Let $N = \lfloor CR/c \rfloor$.
   b) Run the dynamic programming algorithm to determine and output the optimal sawing pattern $S^*(\theta)$ and the corresponding lumber value $V(S^*)$
2) Determine the optimal sawing orientation Θ^* such that $V(S^*(\Theta^*))$ is maximized. Output Θ^*, $S^*(\Theta^*)$ and $V(S^*)$.

In order to determine the computational complexity of the exhaustive search algorithm, we define t_{cr} to be the time taken to compute the cutting range CR, t_{cs} to be the time taken to construct a cutting surface, t_g to be the time taken to determine $g(i,j)$ for each pair (i,j). At each step i, $g(i,j)$ needs to be computed $|T|$ times where $|T|$ is the cardinality of set T and there are N steps in all. Therefore, for each value of the sawing orientation θ, the execution time is given by



(a) heuristic

(b) exhaustive

Fig. 10. Sawing orientation selection

$$Time(S^*(\theta)) = \overline{t_{cr}} + \overline{N} \times \overline{t_{cs}} + \overline{N} \times |T| \times \overline{t_g} \quad (15)$$

in which, $\overline{(.)}$ denotes the average, because in practice N varies with the angular orientation θ, and t_g also varies for different (i,j) values in $g(i,j)$. We define the parameter $t_{sg} = \overline{t_{cs}} + |T| \times \overline{t_g}$. Thus the expression for $Time(S^*(\theta))$ can be simplified as $Time(S^*(\theta)) = \overline{t_{cr}} + \overline{N} \times t_{sg}$. We can see that for a given sawing orientation, the execution time is roughly proportional to the cutting range. This confirms our intuition that the larger the log diameter, the longer it takes to determine the resulting lumber value.

In the case of exhaustive search, the total number of angular orientations explored is $n = R/\delta_\theta$, where δ_θ is the search resolution for the sawing orientation. R is the search range for the sawing orientation and is 180°. The execution time for the exhaustive search is given by:

$$Time(F_{live}(log)) = R/\delta_\theta \cdot (\overline{t_{cr}} + \overline{N} \times t_{sg}) \quad (16)$$

Instead of exhaustively searching the entire range of sawing orientations with a fixed angular resolution, one can perform a coarse-to-fine search by initially searching the entire range of sawing orientations at coarse angular resolution and subsequently narrowing down the search range while simultaneously increasing the angular resolution (decreasing Δ).

Consider a class of objective functions given by:

$$F^*(\zeta) = \max_{\zeta \in R}(F(\zeta)) \quad (17)$$

where the goal is to determine the maximum value of $F(\zeta)$ and the corresponding value of ζ. Function $F(\zeta)$ is known and R is a range of ζ and is denoted by $[A, B]$. The minimum search resolution δ for the parameter ζ is also known. The exhaustive

search algorithm, explores all possible values of ζ at a given resolution and guarantees that the optimal value of $F^*(\zeta)$ can be determined at that resolution. However, the exhaustive search algorithm is computationally inefficient. The proposed two-step coarse-to-fine algorithm is termed as **Algorithm 1** and is summarized below:

Algorithm 1

1) Start with an initial angular resolution Δ (where $\Delta > \delta$), initial range $R = [A, B]$ (where $A = 0^\circ$ and $B = 180^\circ$) and starting centroid $c = (A + B)/2$.
2) Search for the optimal value $F^*(\zeta)$ and corresponding ζ^* for the given values of centroid c, angular resolution Δ and range R.
3) Set $c = \zeta^*$, $R = [\zeta^* - \Delta/2, \zeta^* + \Delta/2]$ and $\Delta = \delta$. Perform step 2) again to determine the optimal value $F^*(\zeta)$ and the corresponding argument ζ^* at resolution δ.

The above two-step coarse-to-fine algorithm runs for $R/\Delta + \Delta/\delta$ iterations whereas the exhaustive search algorithm runs for R/δ iterations in order to determine the optimal value $F^*(\zeta)$. When $\Delta = \sqrt{R\delta}$, the two-step coarse-to-fine algorithm runs for $2\sqrt{R/\delta}$ iterations. For example, for $\delta = 2$ and $\Delta = 16$ the two-step coarse-to-fine live sawing algorithm takes 19 iterations whereas the exhaustive search algorithm takes 90 iterations. The two-step coarse-to-fine algorithm could result in a suboptimal result since not all possible values of the objective function F are examined. However, our experimental results (Section VII) show that the value of F obtained using the two-step coarse-to-fine algorithm is nevertheless close to the optimal value.

The computation time for optimal live sawing using **Algorithm 1** for a given log is:

$$Time(F_{live}(log)) = 2\sqrt{R/\delta_\theta} \cdot (\overline{t_{cr}} + \overline{N} \times t_{sg}) \qquad (18)$$

whereas the computation time for optimal live sawing using exhaustive search is:

$$Time(F_{live}(log)) = R/\delta_\theta \cdot (\overline{t_{cr}} + \overline{N} \times t_{sg}) \qquad (19)$$

B. Cant Sawing

The cant sawing algorithm is based on live sawing. Cant sawing breaks a log into three portions along the initial sawing orientation (Figure 11). Portions 1 and 3, at the extremities of the log cross-section, are subject to live sawing along the initial sawing orientation whereas portion 2 in the center of the log cross-section is subject to live sawing with the sawing planes orthogonal to the initial sawing orientation. Thus, in addition to determining the best initial sawing orientation, the optimal positions of the two breakdown planes, denoted by l_1 and l_2 in Figure 11 also need to be determined in order to arrive at the optimal cant sawing strategy.

The objective function for cant sawing can be expressed as:

$$\begin{aligned} F_{cant}(log) \quad = \quad & (\Theta^*(log), L_1^*(\Theta^*), L_2^*(L_1^*), V(L_1^*, L_2^*), \\ & S_1^*(L_1^*), S_2^*(L_1^*, L_2^*), S_3^*(L_2^*)) \end{aligned} \qquad (20)$$

That is, in the case of cant sawing, the objective is to (a) determine the optimal sawing orientation Θ^*, (b) determine the locations of the sawing planes L_1^* and L_2^* for this orientation, and (c) determine the corresponding optimal lumber value $V(L_1^*, L_2^*)$ and the corresponding sawing patterns $S_1^*(L_1^*)$, $S_2^*(L_1^*, L_2^*)$, and $S_3^*(L_2^*)$ for portions 1, 2 and 3 of the log respectively (Figure 11). Note that for a given value of L_1, an optimal value for L_2 can be determined; thus L_2 is directly dependent on L_1.

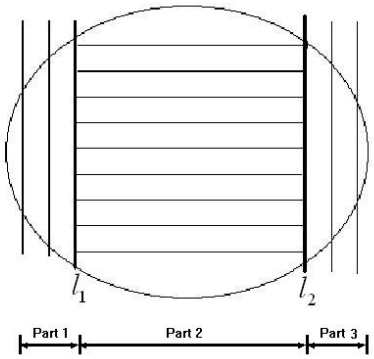


Fig. 11. Cant Sawing

As in the case of live sawing, given a sawing orientation θ, the cutting range CR and the maximum number of sawing planes $N = \lfloor CR/c \rfloor$ can be determined. The variables l_1 and l_2 are used to define the locations of the sawing planes L_1 and L_2 respectively where $0 \leq l_1 < l_2 < N$ (Figure 11). When values of l_1 and l_2 are determined, the optimal processing of portions 1, 2 and 3 can be determined using the optimal live sawing algorithm described earlier (Figure 11). The optimal live sawing algorithm is applied to portions 1, 2 and 3 of the log with sawing orientations θ, $90 - \theta$ and θ respectively. Note that for portions 1 and 3 of the log, it is not necessary to compute the cutting range CR, since $N_1 = l_1$ and $N_3 = N - l_2$ respectively. However, for portion 2, each parameter needs to be recomputed.

For a given log, an exhaustive search algorithm can be used to determine the optimal values for θ, l_1 and l_2 as follows:

1) For each cutting orientation $\theta \in [0, 180)$
   a) Determine the cutting range CR and $N = \lfloor CR/c \rfloor$
   b) For each $l_1 \in [0, N)$
      i) Run the dynamic programming algorithm on portion 1 and determine $V_1^*(l_1)$, $S_1^*(l_1)$.
      ii) For each value of $l_2 \geq (l_1 + 2K/c + W_{min}/c)$.
         A) Run the dynamic programming algorithm on portion 3 and determine $V_3^*(l_2)$, $S_3^*(l_2)$.
         B) Determine the cutting range CR_2 for portion 2 and the corresponding value of N_2.
         C) Run the dynamic programming algorithm on portion 2 and determine $V_2^*(l_1, l_2)$, $S_2^*(l_1, l_2)$.
      iii) Determine the optimal value l_2^*, such that $V_2^*(l_1, l_2^*) + V_3^*(l_2^*)$ is maximized. Output $V_2^*(l_1, l_2^*)$, $S_2^*(l_1, l_2^*)$ and $V_3 * (l_2^*)$, $S_3^*(l_2^*)$.
   c) Determine the optimal value l_1^* such that $V_1^*(l_1^*) + V_2^*(l_1, l_2) + V_3^*(l_2)$ is maximized. Output $V_1^*(l_1^*)$,

$S_1^*(l_1^*)$, $V_2^*(l_1^*, l_2^*)$, $S_2^*(l_1^*, l_2^*)$ and $V_3^*(l_2^*)$, $S_3^*(l_2^*)$.

2) Determine the optimal cutting orientation θ^*, such that, $L_1^* = L_1^*(\theta^*)$ and $V_1^*(l_1^*) + V_2^*(l_1, l_2) + V_3^*(l_2)$ is maximized. Output θ^*, $L_1^*(\theta^*)$, $L_2^*(L_1^*)$, $V_1^*(l_1^*)$, $S_1^*(l_1^*)$, $V_2^*(l_1^*, l_2^*)$, $S_2^*(l_1^*, l_2^*)$ and $V_3^*(l_2^*)$, $S_3^*(l_2^*)$.

Note that $l_2 = l_1 + 2K/c + W_{min}/c$, because the width of portion 2 of the log must be greater than W_{min}. However we allow a special instance of cant sawing where $l_1 = l_2$ (i.e., portion 2 is nonexistent); in which case cant sawing is equivalent to live sawing.

When θ is known, $s^*(l_1)$, $v^*(l_1)$ and $s^*(l_2)$, $v^*(l_2)$ can be precomputed using the dynamic programming algorithm and stored for all possible values of l_1 and l_2. In this case, step i) and step A) in the above algorithm need not be performed at each iteration. However in the case of portion 2, for each pair (l_1, l_2), $S_2^*(l_1, l_2)$ and $V_2^*(l_1, l_2)$ needs to be recomputed. Given a pair (l_1, l_2), the time to compute the optimal sawing pattern and resulting lumber value for portion 2 is given by: $\overline{T(S^*(l_1, l_2))} = t_{cr} + \overline{N_2} \cdot t_{sg}$. The total computation time needed to determine $s_1^*(l_1)$ and $s_3^*(l_2)$ for all l_1 and l_2 is simply given by $Time(S_1^*(N)) + Time(S_3^*(N)) = t_{cr} + N \cdot t_{sg}$. The total number of pairs (l_1, l_2) is given by $(N/\delta_l)^2/2$. In addition, if we let $\overline{N_2} = \overline{N}$, then the total computation time for determination of optimal cant sawing for a specified orientation is given by

$$Time(L^*(\theta)) = ((\overline{N}/\delta_l)/2 + 1)(t_{cr} + \overline{N} \cdot t_{sg}) \tag{21}$$

The total time taken to compute the optimal cant sawing pattern using exhaustive search for determination of the optimal sawing orientation is given by

$$Time(F_{cant}(log)) = ((\overline{N}/\delta_l)^2/2 + 1) \cdot R/\delta_\theta \cdot (t_{cr} + \overline{N} \cdot t_{sg}) \tag{22}$$

If the two-step coarse-to-fine strategy described in **Algorithm 1** is used to compute the optimal positions of sawing planes L_1 and L_2 and the optimal sawing orientation, then the total time taken to compute the optimal cant sawing pattern is given by:

$$Time(F_{cant}(log)) = (2(\overline{N}/\delta_l) + 1) \cdot 2\sqrt{R/\delta_\theta} \cdot (t_{cr} + \overline{N} \cdot t_{sg}) \tag{23}$$

C. Grade Sawing

In the case of grade sawing, for a given sawing orientation θ, a sawing plane (denoted by l_1 in Figure 12) along the same orientation is chosen to divide the log into two portions. Live sawing is performed along sawing orientation θ on portion 1 of the log. For portion 2, a new sawing plane (denoted by l_2 in Figure 12) with orientation orthogonal to θ is chosen to further divide the log into two portions (portions 21 and 22) as illustrated in Figure 12. Live sawing with orientations θ and $90-\theta$ is then performed on the portions 21 and 22 respectively. Therefore, given the sawing orientation, the objective function is given by $F_{grade}(log) = (\Theta^*(log), L_1^*(\Theta^*), L_2^*(L_1^*), V(L_1^*, L_2^*), S_1^*(L_1^*), S_{21}^*(L_1^*, L_2^*), S_{22}^*(L_1^*, L_2^*))$. Thus, in the case of grade sawing, the objective is to (a) determine the optimal sawing orientation Θ^*, (b) determine the locations of the sawing planes L_1^* and L_2^* for this orientation, and (c) determine the corresponding optimal lumber value $V(L_1^*, L_2^*)$ and the corresponding sawing patterns $S_1^*(L_1^*)$, $S_{21}^*(L_1^*, L_2^*)$, and $S_{22}^*(L_1^*, L_2^*)$ for portions 1, 21 and 22 of the log respectively (Figure 12). As in the case of cant sawing, for a given value of L_1, an optimal value for L_2 can be determined; thus L_2 is directly dependent on L_1.

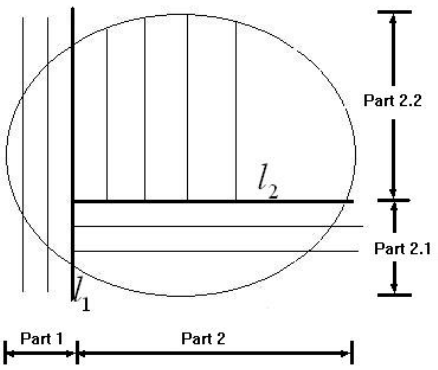


Fig. 12. Grade Sawing

The following algorithm is used to optimize the objective function $F_{grade}(log)$:

1) For each $\theta \in [0, 180)$ and for a given resolution δ_θ.
   a) Determine the cutting range CR_1 and set $N_1 = \lfloor CR_1/c \rfloor$.
   b) For each $l_1 \in [0, N_1)$
      i) Run the dynamic programming algorithm on portion 1 of the log and output $V_1^*(l_1)$, $S_1^*(l_1)$.
      ii) Determine the cutting range CR_{21} for portion 21 of the log, and set $N_{21} = \lfloor CR_{21}/c \rfloor$
      iii) For each $l_2 < N_{21}$
         A) Run the dynamic programming algorithm on portion 21 of the log and output $V_{21}^*(l_1, l_2)$, $S_{21}^*(l_1, l_2)$.
         B) Decide the cutting range CR_{22} for portion 22 of the log and set $N_{22} = \lfloor CR_{22}/c \rfloor$.
         C) Run the dynamic programming algorithm for portion 22 of the log and output $V_{22}^*(l_1, l_2)$, $S_{22}^*(l_1, l_2)$.
      iv) Decide the optimal l_2^*, such that $V_{21}^*(l_1, l_2^*) + V_{22}^*(l_1, l_2^*)$ is maximized. Output $V_{21}^*(l_1, l_2^*)$, $S_{21}^*(l_1, l_2^*)$ and $V_{22} * (l_1, l_2^*)$, $S_{22}^*(l_1, l_2^*)$.
   c) Decide the optimal l_1^* such that $V_1^*(l_1^*) + V_{21}^*(l_1^*, l_2^*) + V_{22}^*(l_1^*, l_2^*)$ is maximized. Output $V_1^*(l_1^*)$, $S_1^*(l_1^*)$, $V_{21}^*(l_1^*, l_2^*)$, $S_{21}^*(l_1^*, l_2^*)$, $V_{22}^*(l_1^*, l_2^*)$, $S_{22}^*(l_1^*, l_2^*)$.
2) Determine the optimal sawing orientation θ^* such that $V_1^*(l_1^*) + V_{21}^*(l_1^*, l_2^*) + V_{22}^*(l_1^*, l_2^*)$ is maximized. Output θ^*, $l_1^*(\theta^*)$, $l_2^*(l_1^*)$, $V_1^*(l_1^*)$, $S_1^*(l_1^*)$, $V_{21}^*(l_1^*, l_2^*)$, $S_{21}^*(l_1^*, l_2^*)$, $V_{22}^*(l_1^*, l_2^*)$, $S_{22}^*(l_1^*, l_2^*)$.

Note that the dynamic programming algorithm in step i) does not have to be performed for each value of l_1. The $s^*(l_1)$ and $v^*(l_1)$ values can be precomputed for each sawing orientation and for each l_1 value. Similarly, the dynamic programming

algorithm in step C) does not have to be performed for each value of l_2, given a value of l_1. However, for portion 21 of the log, all possible pairs of (l_1, l_2) values have to be considered. For portion 1 of the log, the computation time is $t_{cr} + N_1 \cdot t_{sg}$, For portion 21 of the log, there are N_1/δ_l possible values for l_1, therefore, the computation time is $N_1/\delta_l \cdot (t_{cr} + N_{21} \cdot t_{sg})$. For portion 22 of the log, there are $N_1/\delta_l \times N_2/\delta_1$ possible values for (l_1, l_2) so the computation time is $N_1/\delta_l \times N_2/\delta_1 \cdot (t_{cr} + N_{22} \cdot t_{sg})$. If we let $\overline{N_1} = \overline{N_2} = \overline{N_{21}} = \overline{N_{22}} = \overline{N}$ then the total computation time for determination of optimal grade sawing is

$$Time(F_{grade}(log)) = (1+\overline{N}/\delta_l+(\overline{N}/\delta_l)^2)\cdot R/\delta_\theta\cdot(t_{cr}+\overline{N}\cdot t_{sg}) \quad (24)$$

If the two-step coarse-to-fine strategy outlined in **Algorithm 1** is used to determine the optimal sawing plane positions l_1 and l_2 and the optimal sawing orientation, then the total computation time for determination of optimal grade sawing is given by $Time(F_{grade}(log)) = (4(\overline{N}/\delta_l) + 2\sqrt{\overline{N}/\delta_l} + 1) \cdot 2\sqrt{R/\delta_\theta} \cdot (t_{cr} + \overline{N} \cdot t_{sg})$.

D. Secondary Sawing

In primary sawing (i.e., live sawing, cant sawing, grade sawing) slabs of wood termed as flitches are produced. These flitches are further processed to produce edged (cut lengthwise) and trimmed (cut widthwise) pieces termed as boards (Figure 7). This procedure is called secondary sawing. The purpose of secondary sawing is to produce higher quality board products from flitches.

The parameters used in secondary sawing are the same as those used in primary sawing. The viewing angle for the flitch is perpendicular to the surface of the lumber. The orientation of the sawing plane is fixed and is either parallel or perpendicular to the Z axis as illustrated in Figure 7. In primary sawing, the constraints are primarily imposed on the lumber thickness T. In the case of secondary sawing, the constraints are imposed on the board width W and length L.

There is no concept of cutting range in the case of secondary sawing. A flitch of width W_f, for a given sawing resolution c, is divided into $N = \lfloor W_f/c \rfloor$ sawing planes. The objective function for secondary sawing is $F_{second}(flitch) = (S^*(flitch), V(S^*))$, that is, given a flitch, determine the optimal sawing pattern S^* and corresponding maximum value $V(S^*)$. Note that there is no dependence on the sawing orientation which is fixed (unlike primary sawing). The dynamic programming procedure to optimize $F_{second}(flitch)$ can be stated as follows:

Let $s^*(i)$ to be the optimal sawing pattern for sawing plane positions from 1 to i for the given flitch and $v^*(i)$ to be the corresponding value. If $v^*(k)$ and $s^*(k)$ for all $k \leq i$ are known, then

$$v^*(i+1) = \max_{j\in[0,W]}(\quad v^*(i+1-\lceil W_j/c\rceil - \lceil K/c\rceil) + g(i+1-\lceil W_j/c\rceil, i+1)) \quad (25)$$

where $W = \{W_1, W_2, \ldots, W_n\}$ is the allowed set of width values and $g(i, j)$ is value of the portion of the board between sawing planes i and j. As in the case of live sawing, $\lceil K/c \rceil$ is the kerf size which determines the minimum gap between two boards. Note that the width W_j is used instead of the thickness T_j as in the case of primary sawing. As in the case of live sawing, the constraints on the sawing pattern S are relaxed i.e., j can be 0 in which case $W_0 = 0$. This means that if $j^* = 0$, then the additional portion $(i, i+1)$ would not increase the maximum value of the log at point $i+1$. This portion could be currently ignored, but could also be reconsidered at a later point. The above constraint relaxation makes it possible to discard some portion(s) of the flitch that contain(s) too many defects.

If j^* results in the best value $v^*(i+1)$ then

$$s^*(i+1) = s^*(i+1-W_j^*/c-\lceil K/c\rceil)\cup\{i+1-W_j^*/c\} \quad (26)$$

Equation (26) defines a dynamic programming algorithm. The outputs $s^*(N)$ and $v^*(N)$ of the dynamic programming algorithm are the optimal sawing pattern $S^*(log)$ and the corresponding value $V(S^*)$ respectively.

In secondary sawing, there is no need to generate the lumber surface for a sawing plane as in the case of primary sawing; it is sufficient to only check the appearance of defects on the top and bottom surface of the board. Therefore, the time complexity of secondary sawing for a given flitch is: $Time(F_{second}(flitch)) = N \times w \times t_g$, in which t_g is the time to determine the value of a given board.

A similar dynamic programming approach can be formulated to generate boards of allowable lengths resulting in a two-step dynamic programming algorithm. A flitch of width W_f and length L_f for a given sawing resolution c, is divided into $N_W = \lfloor W_f/c \rfloor$ horizontal sawing planes and $N_L = \lfloor L_f/c \rfloor$ vertical sawing planes. Let $s^*(i, j)$ to be the optimal sawing pattern for the horizontal sawing plane positions from 1 to i and vertical sawing plane positions from 1 to j for the given flitch, and $v^*(i, j)$ to be the corresponding value. If $v^*(k, l)$ and $s^*(k, l)$ for all $k \leq i$ are known, then

$$\begin{aligned} v^*(i+1, j+1) \quad &= \max_{k\in[0,W]}(\max_{l\in[0,L]}(\\ &\quad (v^*(i+1-\lceil W_k/c\rceil - \lceil K/c\rceil, \\ &\quad j+1-\lceil L_l/c\rceil - \lceil K/c\rceil) \\ &\quad +g(i+1-\lceil W_j/c\rceil, i+1, \\ &\quad j+1-\lceil L_l/c\rceil, j+1)))) \end{aligned} \quad (27)$$

where $L = \{L_1, L_2, \ldots, L_n\}$ is the set of allowable lengths and $g(i, j, k, l)$ is the value of the lumber between the horizontal sawing planes (i, j) and vertical sawing planes (k, l).

E. Lumber Grading

Given a flitch or a board, a well-defined and systematic grading system is essential to determine its value. An automated grading subsystem is a critical component of an overall automated lumber production planning system. The grading process is that of classification of a board or flitch into one of various grades based on its size and defect content and computation of its price based on the assigned grade and its surface area. The most commonly used hardwood grading

rules are the ones promulgated by the National Hardwood Lumber Association (NHLA) [22]. In practice, manual lumber grading via application of the various NHLA rules is a complicated process. Consequently, lumber companies usually hire trained professional lumber graders for this important task. In our automated lumber production planning system, an automated lumber grading subsystem is implemented based on compilation of the various NHLA grading rules.

Under the NHLA hardwood grading system, the grade assigned to a hardwood board or flitch is based on the number and sizes of clear/sound face cuttings that can be obtained from a board or flitch. A clear face cutting is a piece that is free of defects on one side except for minor seasoning checks. The standard NHLA grades in the order from highest to lowest are *First and Second (FAS)*, *Selects*, *Number 1 Common*, *Number 2A and 2B Common*, *Number 3A Common* and *Number 3B Common* as shown in Figure 13. For example, *FAS* grade lumber is the best suited for high quality furniture, high quality veneer, interior trim or molding. Some important terminology pertaining to lumber grading using the NHLA rules is given as follows.

1) Lumber Grading Terminology:

Surface measure: Surface measure (SM) is the surface area of a board in square feet and is computed as follows:

$$SM = width(\text{inches}) \times integer\_length(\text{feet}) \div 12 \quad (28)$$

In equation (28) the width of the board is measured in inches (including the fractional portion) whereas the length of the board is measured in integral feet obtained via truncation of the actual length in feet. The result is expressed as an integer obtained via rounding of the product. For example, a flitch of size $6\frac{7}{16}\text{inches} \times (10\text{feet } 9\text{inches})$, has a surface measure of $SM = (6\frac{7}{16} \times 10)/12 = 5.36 \approx 5\text{ft}^2$.

Poor side: One of the two surfaces of a flitch is considered as the poor side. The poor side of the flitch is the one associated with a lower grade. If both sides have the same grade, then the surface with fewer cutting units is deemed the poor side.

Cutting: A rectangular portion of a board or flitch. Different grades have different requirements on the quality, sizes and number of the resulting cuttings.

Clear Cutting: A cutting without any visible surface defects.

Sound cutting: A cutting with only certain allowed types of visible surface defects.

Cutting Unit: Cutting unit (*CU*) is a unit of measure comprising of a 1 inch width and a 1 foot length. The number of cutting units associated with a cutting is computed as:

$$CU = [width\ (\text{inches}) + fraction] \times [length\ (\text{feet}) + fraction] \quad (29)$$

For the sake of clarification consider the example board shown in Figure 14. The *SM* of the board is $10(\text{in}) \times 11(\text{feet})/12 = 9.16 \approx 9\text{ft}^2$. The number of cuttings is two as indicated by the two rectangular boxes A and B in Figure 14. Since both the cuttings are free of defects, they are deemed to be clear cuttings. The total number of cutting units associated with the board is computed as: $CU = (6.5\text{in} \times 7\text{ft}) + (3.5\text{in} \times 9\text{ft}) = 77$

2) Grading Rules: Different lumber grades have different requirements with regard to the quality, sizes and number of the resulting cuttings as shown in the grading table in Figure 13. For example, to qualify as grade *FAS*, a board must have a width greater than 6 inches, length between 8 feet and 16 feet. The *FAS* grade allows a total of at most $SM/4$ cuttings to be used to determine the total number of clear cutting units. A cutting must be larger than $4 \text{ inches} \times 5 \text{ feet}$ or $3 \text{ inches} \times 7 \text{ feet}$. The total number of clear cuttings units must be at least $10/12 \times (SM \times 12)$. For example, in the case of the board shown in Figure 14 the required number of clear *CU*s is $10/12 \times 9 \times 12 = 90$ for the board to be classified as *FAS* grade. If the maximum number of clear *CU*s obtained from the board is 77, as shown earlier, then this board cannot be classified as *FAS* grade. However, it is possible to determine the optimal locations of the cuttings that would enable the board to be classified as *FAS* grade.

Grade	FAS	Select	1 Com	2A&2B COM	3A COM	3B COM
Minimum Width	6 in	4 in	3 in	3 in	3 in	3 in
Length	8-6 ft	6-16 ft	4-16 ft	4-16 ft	4-16 ft	4-16 ft
Minimum Size of Cutting	4 in × 5 ft 3 in × 7 ft	4 in × 5 ft 3 in × 7 ft	4 in × 2 ft 3 in × 3 ft	3 in × 2 ft	3 in × 2 ft	Width > 1.5 in
Overall yield based on clear face cuttings	10/12	10/12	8/12	6/12	4/12	3/12
Formula determining number of cutting	SM÷4	(SM+1)÷4	(SM+1)÷3	SM÷2	Unlimited	Unlimited

Fig. 13. Grading Table

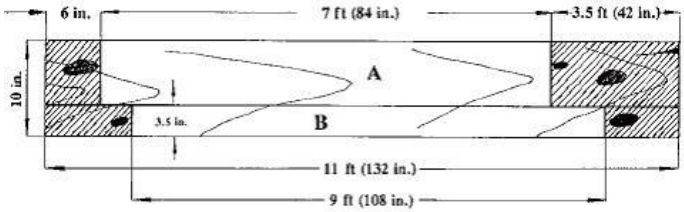


Fig. 14. Clear cutting units and grading of a board

To determine the highest possible grade for a board, we first compute its surface measure (*SM*) and then the number of cutting units (*CU*s). The required number of clear cuttings is determined using the requirements displayed in Figure 13. Next, the maximum number of clear cutting units is determined based on the locations of the defects as described below.

3) A Mathematical Model to Compute Clear Cutting Units: The width, length and the surface measure *SM* of a board or flitch are easily computed. The real difficulty lies in determining the number and locations of the clear/sound cuttings since the stipulated requirements on the number and sizes of the cuttings and the total number of cutting units (*CU*s) varies with the grade. Therefore, to classify a board as belonging to a certain grade, the grading system needs to compute the number and sizes of the clear/sound cuttings and the total number of *CU*s associated with the board and match these with the stipulated requirements for each grade as tabulated in the grading table (Figure 13).

Suppose the length and width of the board are L and W respectively. The image of the board surface is divided into a $L/r_l \times W/r_w$ grid as shown in Figure 15 where r_w and r_l

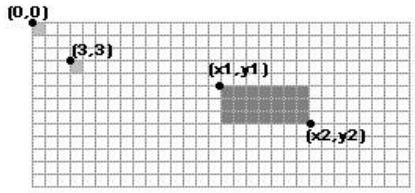


Fig. 15. The grid image used for determining the clear/sound cuttings

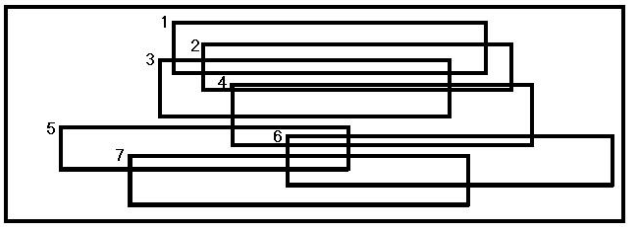


Fig. 16. Sorted legal cuttings

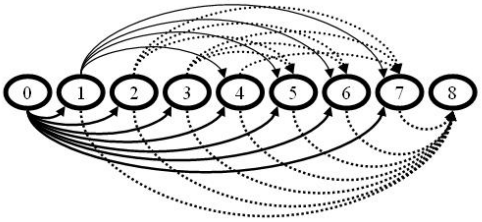


Fig. 17. Cutting graph

denote the dimensions of each grid cell along the width and length respectively. The parameters r_w and r_l thus control the grid granularity or resolution. A finer grid resolution increases the computational complexity whereas a coarser grid resolution decreases the accuracy of the analysis. A grid cell p in the yth row and xth column is represented as $p(x, y)$. Grid cell $p_1(x_1, y_1)$ is said to precede cell $p_2(x_2, y_2)$ if one of the two conditions is satisfied:: (1) $y_2 > y_1$; (2) $y_2 = y_1$ and $x_2 > x_1$. Two cells $p_1(x_1, y_1)$ and $p_2(x_2, y_2)$ where p_1 precedes p_2, define a rectangle with p_1 and p_2 as the end points of its diagonal. A grid cell $p(x, y)$ is deemed to lie within the rectangle when $x_1 \leq x < x_2$ and $y_1 \leq y < y_2$. Given a log and the 3D geometry and locations of its internal defects, the grid cells containing each defect can be determined. To enhance the computational efficiency of the search for clear cuttings, the grid image is encoded using run-length codes along both the x-direction and y-direction in the defect-free regions.

The mathematical model for determining the optimal number of clear cuttings is described below. The variables used in this model are listed as follows.

- N_c: the maximum number of allowable cuttings for a certain grade. This value is determined using the grading table.
- $S = \{(w_i, l_i)\}$, $i = 1, ..., s$: the minimum allowable sizes of the cuttings for a certain grade where the width and length are expressed in grid units. These values are obtained from the grading table. Under the NHLA standard, the value of s is usually small (1 or 2).
- C_u: The minimum number of clear cutting units allowed for a certain grade. This value is also obtained from the grading table.

Thus, given a board/flitch surface, the objective is to obtain the maximum number of clear cutting units under the constraints of N_c and S. Therefore, the target function is represented as $F_{cut} = (C^*, CU(C^*))$, where C^* is the set of optimal cuttings and $CU(C^*)$ is the number of cutting units associated with C^*. The number of cuttings in C^* is no larger than N_c. The size of each cutting in C^* is no less than that of any element in S.

A legal set of cuttings of a board is represented as an array of nonoverlapping rectangles $C = \{c_j(p_1, p_2)\}$, $1 \leq j \leq c \leq N_c$, where c is the total number of cuttings. For convenience, let p_1^j represent the first grid cell and p_2^j the second grid cell that define rectangle $c_j(p_1, p_2)$ (note that (p_1^j, p_2^j) defines the diagonal of the rectangle c_j). A feasible solution C to the cuttings determination problem satisfies the following conditions: (1) Rectangles c_i and c_j are non-overlapping for $1 \leq i, j \leq c$; and (2) Rectangle p_1^i precedes rectangle p_1^j for $1 \leq i < j \leq c$.

Given a board/flitch, all the legal cuttings are scanned and stored in an array $C = \{c_j\}$. The array C is sorted based on the coordinates of the first grid cell of each cutting as shown in Figure 16. A directed graph for array C is then constructed. The nodes of the graph are cuttings labeled based on their order in array C. Each node j in the graph is assigned a value $CU(c_j)$. Two non-overlapping nodes i and j in the graph, where $i < j$, are connected by a directed edge e_{ij} from node i to node j. Two dummy nodes 0 and $c + 1$ are added to the graph. The value associated with nodes 0 and $c + 1$ is 0. For any node j in the graph, where $1 \leq j \leq c$, there is an edge from node 0 to node j from node j to node $c+1$. The resulting graph is shown in Figure 17 in which there are directed edges from node 1 to nodes 4, 5, 6, 7 and 8, because cutting 1 in Figure 16 does not overlap with cuttings 4, 5, 6 and 7.

A path in the above graph is defined as a sequence of nodes $p = 0, i_1, i_2, ..., i_k, c+1$, that satisfies the conditions $0 < i_1 < i_2 < ... < i_k < c+1$ and $k \leq N_c$. The total number of cutting units associated with this path is given by

$$CU(p) = \sum_{j=1}^{k} CU(c_{i_j}) \tag{30}$$

The optimal set of cuttings corresponds to the path p^* which maximizes $CU(p)$. A dynamic programming algorithm outlined below is used to find the optimal path p^*:

(1) Initialization: $C^*(0) = \Phi$ and $CU^*(0) = 0$ for node 0 where Φ is the empty set.
(2) For each node $i \in [1, c + 1]$, compute

$$CU^*(i) = CU(i) + \max_{j \in p(i)} CU(j) \tag{31}$$

where $p(i)$ is the set of all nodes which have edges to node i. $C^*(i) = C(arg\ (\max_{j \in p(i)} CU(j))) \cup \{c_i\}$

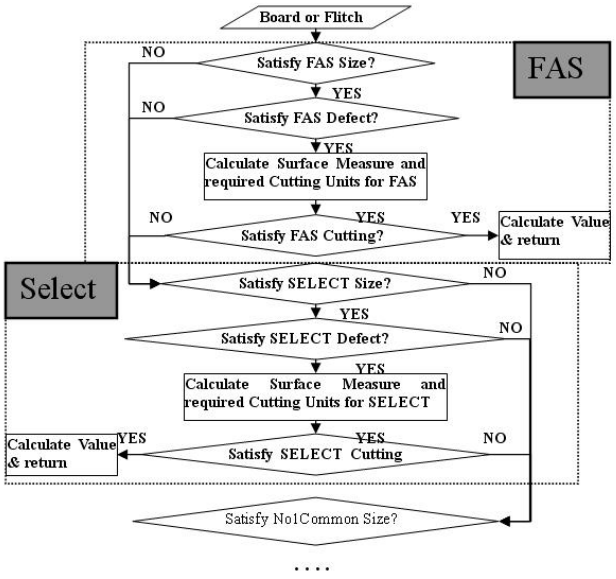


Fig. 18. Flow chart of the grading system

(3) $C^* = C^*(c+1)$ is the optimal set of cuttings and $CU^* = CU^*(c+1)$ is the corresponding optimal number of cutting units.

If the optimal number of cutting units $CU^* \geq C_u$ for a given grade, then the board/flitch surface can be classified as belonging to that grade. The classification of a board/flitch surface proceeds from the highest grade *FAS* to the lowest grade *3B COM*. The values of C^* and CU^* are computed subject to the constraints of each grade category and the board/flitch surface assigned to the highest grade that it qualifies for.

4) Grading system: The flowchart of the overall grading system is summarized in Figure 18. Given the two surfaces of a board or flitch, both surfaces are graded. Once the grades of both surfaces determined, the final grade of the entire board/flitch is determined. The final grade of the entire board/flitch is typically the lower of two surface grades. After the final grade of a board/flitch is determined, the price of the board/flitch is computed based on a pricing table. The pricing table typically contains the cost per clear cutting unit for each grade category.

F. Lumber value determination

In practice, the lumber value is determined by its species group, its grade, its size (width, thickness and length) and market demand. The hardwood lumber grade is based on the appearance of the defects on lumber surfaces and determined according to the NHLA rules [22]. The effect of the market demand factor can be typically represented by other factors related to species, grade, volume and dimensions of the lumber. Thus the value of a piece of lumber v may be obtained using the following formula:

$$v = VP_sP_gP_wP_tP_l \tag{32}$$

where V is the lumber volume, P_s, the base price for a unit volume of a certain lumber species, P_g, the price factor related to the lumber grade, and P_w, P_t and P_l the price factors related to the lumber width, lumber thickness and lumber length, respectively.

TABLE I
THE LUMBER VALUE FACTOR: GRADE AND THICKNESS

Grade	P_g	thickness (mm)	P_t
FAS	1100	(5, 10]	0.75
Select	800	(10, 15]	0.8
No1 Common	500	(15, 25]	1
No2 Common	400	(25, 40]	1.1
No3a Common	350	(40,60]	1.05

TABLE II
THE LUMBER VALUE FACTOR: LENGTH AND WIDTH

length (mm)	P_l	width (mm)	P_w
(50,200]	0.7	(50, 100]	0.8
(200, 400]	0.8	(100,150]	0.95
(400, 800]	0.9	(150,250]	1
(800, 1000]	1	(250,350]	1.1
(1000, 1200]	1.05	(350,450]	1.15
(1200, 1400]	1.1		

The pricing model is typically a complex function of several market conditions. Therefore the pricing model shown above is just an example to illustrate the procedure for determination of the optimal sawing strategy. Derivation of an accurate pricing model is beyond the scope of the paper. Table I and II list the price factors used in our system for the hardwood species White Ash. Similar price factor tables can be obtained for other hardwood species.

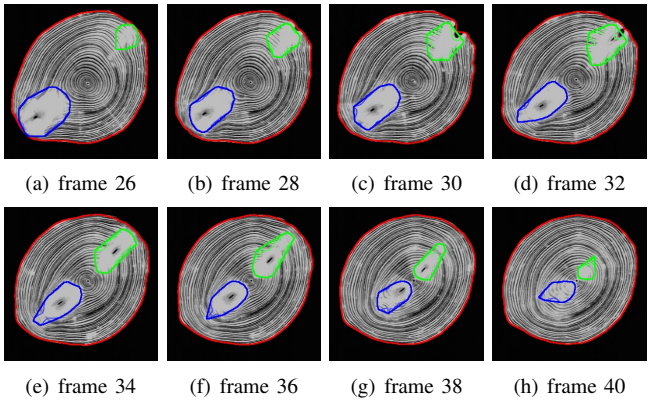
VII. EXPERIMENTAL RESULTS

The proposed Kalman filter-based tracking scheme for detection, localization and 3-D reconstruction of internal defects in hardwood logs from CT image data was subject to experimental verification and validation. Experiments were conducted on four sets of log data, from three popular hardwood species found in the United States, namely, White Ash, Red Oak and Hard Maple, and were labeled as *Ash1*, *Ash2*, *Maple* and *Oak* respectively. The cross-sectional CT images of the hardwood logs were captured using a Toshiba TCT 20AX CT scanner described in Section III. All the programs were run on a 2.0 GHz Pentium 4 Xeon workstation with 1.5 GByte RAM and 1.0 MByte of cache memory.

A. Experimental Results for Defect Detection

Figure 19 shows the results of Kalman filter-based tracking and contour fitting using Snakes for detection and localization of knots over a continuous sequence of CT image slices. Likewise, Figure 20 shows the results for detection and localization of cracks and holes over a sequence of CT images. Holes are represented by their bounding rectangles. Rectangles with the same color in different image slices correspond to the same hole. A crack is modeled as a line, but for the sake of clarity a rectangle is used to mark its locations (Figure 20). A semi-transparent view of the virtually reconstructed log showing its internal defects is depicted in Figure 21.

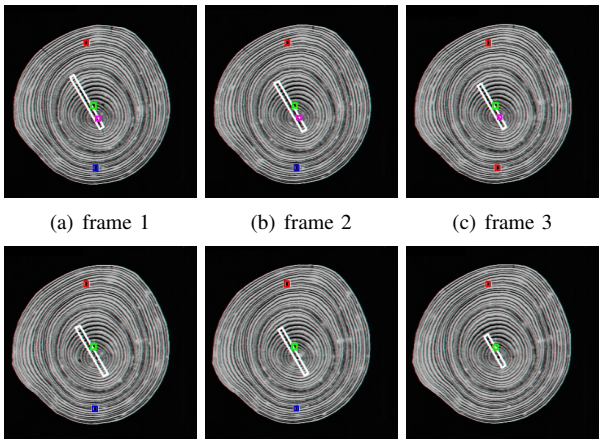
Table III summarizes the defect detection performance of the proposed scheme for over 224 cross-sectional CT image slices of hardwood log data *Ash1* for each of the three major



(a) frame 26 (b) frame 28 (c) frame 30 (d) frame 32

(e) frame 34 (f) frame 36 (g) frame 38 (h) frame 40

Fig. 19. Results of Kalman filter-based tracking and Snakes contour fitting for detection, localization and 3-D reconstruction of knots



(a) frame 1 (b) frame 2 (c) frame 3

(d) frame 4 (e) frame 5 (f) frame 6

Fig. 20. Results of Kalman filter-based tracking and Snakes contour fitting for detection, localization and 3-D reconstruction of cracks and holes

internal defect types: knots, holes and cracks. The detection rate, false positive rate and false negative rate of the proposed scheme are computed by comparing the results of the proposed scheme with those obtained from a human expert grader examining the physically sawn lumber. Also, the performance of the previous scheme [1], [2] that detected and localized defects in each CT image slice independently (i.e., slice by slice without tracking the defects across multiple CT image slices) was compared with the performance of the proposed scheme (Table III). Although the Kalman filter-based tracking scheme did not result in any improvement in the detection rate or the false positive rate for knots, it did improve the detection rate for cracks from 94% to 98% and the false positive rate from 12% to 2% when compared to the scheme that processed and analyzed each CT image slice independently. Likewise,

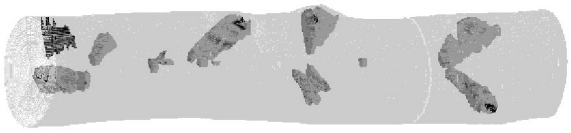


Fig. 21. A semi-transparent view of the log showing its internal defects

TABLE III
THE DETECTION RATE FOR THE THREE MAJOR DEFECT TYPES FOR LOG DATA *Ash1*

	correct	false negative	false positive
	knot	(total 24)	
slice by slice	24(100%)	0(0%)	0(0%)
tracking	24(100%)	0(0%)	0(0%)
	crack	(total 112)	
slice by slice	105(94%)	7(6%)	13(12%)
tracking	110(98%)	4(4%)	2(2%)
	hole	(total 25)	
slice by slice	24(96%)	1(4%)	3(12%)
tracking	24(96%)	1(4%)	1(4%)

TABLE IV
THE DETECTION RATE FOR THE THREE MAJOR DEFECT TYPES FOR LOG DATA *Ash2*

	correct	false negative	false positive
	knot	(total 135)	
slice by slice	135(100%)	0(0%)	0(0%)
tracking	135(100%)	0(0%)	0(0%)
	crack	(total 22)	
slice by slice	21(95%)	1(5%)	3(15%)
tracking	21(95%)	1(5%)	1(5%)
	hole	(total 159)	
slice by slice	159(100%)	0(0%)	0(0%)
tracking	159(100%)	0(0%)	0(0%)

the proposed Kalman filter-based scheme did not improve the detection rate for holes but did improve the false positive rate from 12% to 3%.

Table IV, Table V and Table VI summarize the performance of the proposed Kalman filter-based tracking scheme for hardwood log data *Ash2*, *Maple* and *Oak* respectively. In all cases, it can be seen that the proposed Kalman filter-based tracking approach does improve upon the performance of the scheme that processes and analyzes each CT image slice independently in terms of both, defect detection rate and false positive rate. Overall, it was empirically observed, based on the available CT image data sets in Tables III–VI, that detection of false knots or insertion of missing knots is typically not an issue since knots tend to be fairly large and distinct. Holes, on the other hand, could be missed if they are small in diameter. However, after exploiting spatial coherence it was possible to restore missing hole defects. In the case of crack defects, the Kalman filter-based tracking technique was observed to be robust enough to detect, localize and compute a 3-D reconstruction of a crack, once detected and localized in previous CT image slices. Thus, in the case of false cracks and holes, verifying spatial support from previous and/or succeeding CT image slices was found to be very effective in their removal.

Table VII compares the average processing time per CT image slice of the proposed Kalman filter-based tracking scheme and the scheme that processes and analyzes each CT image slice independently (i.e., slice by slice) on the hardwood log data *Ash1*. In order to ensure a fair comparison, the detection time for the outer log boundary and each of the three major internal defect types i.e., knot, crack and hole are

TABLE V
THE DETECTION RATE FOR THE THREE MAJOR DEFECT TYPES FOR LOG DATA *Maple*

	correct	false negative	false positive
	knot	(total 12)	
slice by slice	12(100%)	0(0%)	0(0%)
tracking	12(100%)	0(0%)	0(0%)
	crack	(total 21)	
slice by slice	19(90%)	2(10%)	1(5%)
tracking	21(100%)	0(0%)	1(5%)
	hole	(total 27)	
slice by slice	27(100%)	0(0%)	0(0%)
tracking	27(100%)	0(0%)	0(0%)

TABLE VI
THE DETECTION RATE FOR THREE MAJOR DEFECT TYPES FOR LOG DATA *Oak*

	correct	false negative	false positive
	knot	(total 15)	
slice by slice	15(100%)	0(0%)	0(0%)
tracking	15(100%)	0(0%)	0(0%)
	crack	(total 0)	
slice by slice	0(0%)	0(0%)	0(0%)
tracking	0(0%)	0(0%)	0(0%)
	hole	(total 15)	
slice by slice	15(100%)	0(0%)	0(0%)
tracking	15(100%)	0(0%)	0(0%)

measured and tabulated independently.

B. Experimental Results for Optimal Sawing Scheme Determination

Tables VIII and IX summarize the performances of the exhaustive search algorithm and the two-step coarse-to-fine search algorithm (**Algorithm 1**) proposed in this paper, respectively, when used to determine the optimal sawing orientation and optimal sawing pattern for live sawing. In this experiment, the angular resolution $\delta_\theta = 2^\circ$ and the angular search range $R = [0, 180)$. A comparison of Tables VIII and IX shows that **Algorithm 1** results in the same optimal solution as the exhaustive search algorithm in the case of the first two log samples. In contrast, **Algorithm 1** produces suboptimal results in the case of the third log sample *Maple* resulting in 96.7% of the optimal value and in the case of the fourth log sample *Oak* resulting in 81.7% of the optimal value. However, the execution time of **Algorithm 1** is, on average, only 17% of the execution time of the exhaustive search algorithm in the context of live sawing.

The three primary sawing methods (live sawing, grade sawing and cant sawing) are compared in Figure 22 on the

TABLE VII
THE AVERAGE PROCESSING TIME PER CT IMAGE SLICE IN MILLISECONDS

	boundary	knots	crack	hole
slice by slice	40	95	101	15
tracking	16	61	81	15

TABLE VIII
DETERMINATION OF OPTIMAL LIVE SAWING USING THE EXHAUSTIVE SEARCH ALGORITHM

log species	orientations	time	value
Ash1	90	447 sec	$13.59
Ash2	90	272 sec	$17.40
Maple	90	210 sec	$28.29
Oak	90	206 sec	$32.53

TABLE IX
DETERMINATION OF OPTIMAL LIVE SAWING USING **Algorithm 1**

log species	orientations	time	value
Ash1	19	73 sec	$13.59
Ash2	18	66 sec	$17.40
Maple	20	35 sec	$27.35
Oak	20	35 sec	$26.57

log sample data *Ash1* using exhaustive search to determine the optimal log orientation. In the absence of **Algorithm 1**, live sawing takes 447 seconds (7 minutes 27 seconds), cant sawing takes 82593 seconds (22 hours 56 minutes 33 seconds) and grade sawing takes 92993 seconds (25 hours 49 minutes 53 seconds) to process the 4 meter log. These run times are currently not suitable for real time application in sawmills. Note that the optimal lumber yield values returned by grade sawing and cant sawing are comparable whereas those returned by live sawing are significantly lower (Figure 22). However, optimal live sawing determination is at least 2 orders of magnitude faster than the determination of optimal grade sawing or optimal cant sawing.

Table X compares the execution time of the optimal sawing determination algorithm for each of the three primary sawing techniques and for each log data set using **Algorithm 1** to determine the optimal log orientation. The use of **Algorithm 1** clearly reduces the execution time by an order of magnitude for each of the three primary sawing techniques. Once again, the optimal lumber yield values returned by cant sawing and grade sawing are comparable whereas those returned by live sawing are significantly lower. However, the algorithm for determination of optimal live sawing is computationally far

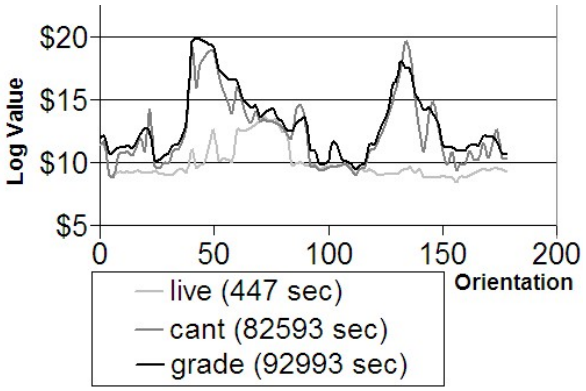


Fig. 22. A comparison between the optimal lumber yield values returned by various primary sawing schemes using exhaustive search to determine optimal log orientation

TABLE X
COMPARISON OF EXECUTION TIME OF THE OPTIMAL SAWING DETERMINATION ALGORITHM FOR LIVE SAWING, GRADE SAWING AND CANT SAWING USING **Algorithm 1** TO DETERMINE THE OPTIMAL SAWING ORIENTATION

Sawing Scheme	Ash1	Ash2	Maple	Oak
live sawing	73 sec	66 sec	35 sec	35 sec
cant sawing	2482 sec	2598 sec	997 sec	959 sec
grade sawing	4032 sec	4099 sec	2332 sec	2600 sec

TABLE XI
THE COMPARISON OF OPTIMAL LUMBER YIELD RECOVERY FOR LIVE SAWING, GRADE SAWING AND CANT SAWING USING **Algorithm 1** TO DETERMINE THE OPTIMAL SAWING ORIENTATION

Sawing	Ash1	Ash2	Maple	Oak
live sawing	$13.59	$17.40	$27.35	$26.57
cant sawing	$19.08	$27.98	$49.53	$54.86
grade sawing	$19.78	$28.12	$50.11	$57.54

more efficient compared to its cant sawing and grade sawing counterparts.

VIII. CONCLUSIONS AND FUTURE DIRECTIONS

This paper described the design and implementation of an automated system for planning and optimization of lumber production using Machine Vision and Computed Tomography (CT). Cross-sectional CT images of hardwood logs were analyzed using Machine Vision algorithms. A novel feature-based tracking approach was proposed and implemented for the detection, localization and 3-D reconstruction of internal defects in hardwood logs from cross-sectional computed tomography (CT) images. In contrast to traditional methods, where the defects are detected and localized independently in individual CT image slices and the 3-D reconstruction of the defects accomplished via correspondence analysis across the various CT image slices, the proposed system integrated defect detection, defect localization and 3-D defect reconstruction within a Kalman filter-based feature tracking scheme. The defects were simultaneously detected, classified, localized and reconstructed in 3-D space, making the proposed scheme computationally much more efficient than existing methods. Robust techniques for defect detection and classification were proposed and implemented. Defect class-specific tracking schemes based on the Kalman filter were designed which used the geometric parameters of the defect contours as the tracking variables. The geometric parameters of the defect contours were computed using a combination of B-spline contour approximation and an improvised Snakes contour fitting procedure termed as Kalman Snakes. Robust techniques for extraction and characterization of the external log surface were also designed.

The proposed system was shown to be capable of simulating various sawing operations on the virtual 3-D reconstruction of the log and exploiting the knowledge of the internal defects to formulate sawing strategies that optimize the value yield recovery of the resulting lumber products. Algorithms for the determination of the optimal primary sawing strategy were designed and implemented for live sawing, grade sawing and cant sawing which are commonly used primary sawing strategies in real sawmills. Algorithms for the determination of the optimal secondary sawing strategy (i.e., decomposing a board/flitch into cuttings) were also implemented. The system was designed to be capable of automatic grading of the lumber products to compute their value using the grading rules stipulated by the National Hardwood Lumber Association (NHLA). A prototype implementation of the system showed significant gains in value yield recovery when compared to lumber processing strategies that use only the external log structure information. The system could be used as a decision aid for lumber production planning as well as a training tool to train novice sawyers.

Although the execution times of the algorithms for defect detection, defect tracking, defect reconstruction, optimal primary and secondary sawing strategy determination and automated lumber grading are currently not suitable for real time application in a sawmill, the execution times could be improved significantly by the incorporation of parallel processing. The operations of CT scanning, defect detection, defect tracking, defect reconstruction, optimal primary and secondary sawing strategy determination and automated lumber grading could be pipelined. These operations could also be performed on different sections of a 4 meter log in parallel (i.e., data parallelism). The operations for optimal primary and secondary sawing strategy determination could also be parallelized by analyzing multiple log orientations and multiple sawing planes concurrently (i.e., task parallelism). Making the system capable of real time performance by means of parallel computing is a promising topic for future research.

ACKNOWLEDGEMENT

This work was supported in part by the US Department of Agriculture through an NRICGP grant (Award Number: 2001-35103-10049) to Drs. Bhandarkar, Daniels and Tollner.

REFERENCES

[1] S.M. Bhandarkar, T. Faust and M. Tang, CATALOG: A system for detection and rendering of internal log defects using computer tomography, *Journal of Machine Vision and Applications*, Vol. 11, pp. 171–190, 1999.

[2] S.M. Bhandarkar, T.D. Faust and M. Tang, Design and Prototype Development of a Computer Vision-based Lumber Production Planning System, *Journal of Image and Vision Computing*, Vol. 20, No. 3, pp. 167–189, 2002.

[3] S.M. Bhandarkar, X. Luo, R. Daniels and E.W. Tollner, Detection of Cracks in Computer Tomography Images of Logs, *Pattern Recognition Letters*, Vol. 26, No. 14, pp. 2282–2294, Oct. 2005.

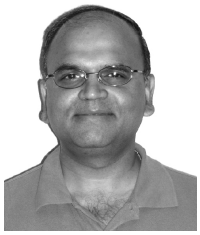
[4] S.M. Bhandarkar, X. Luo, R. Daniels and E.W. Tollner, A Novel Feature-based Tracking Approach to the Detection, Localization and 3-D Reconstruction of Internal Defects in Hardwood Logs Using Computer Tomography, *Pattern Analysis and Applications*, Vol. 9, Nos 2/3, pp. 155–175, Oct. 2006.

[5] D.A. Butler, C.C. Brunner, and J.W. Funck, An adaptive image preprocessing algorithm for defect detection in Douglas-fir veneer, *Forest Products Journal*, Vol. 43, No. 5, pp. 57–60, 1993.

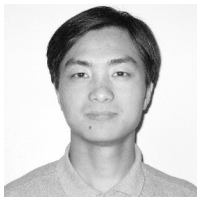
[6] S.J. Chang, J.R. Olson, and P.C. Wang, NMR imaging of internal features in wood, *Forest Products Journal*, Vol. 39, No. 6, pp. 43–49, 1989.

[7] S.J. Chang, S. Guddanti and C.S. Cooper, Measuring the benefits of internal log defect scanning: a mill-based study, *Proc. Intl. Conf. Scanning Technology and Process Optimization for the Wood Products Industry*, Charlotte, NC, pp. 25–28, Nov. 12–14, 1997.

[8] S. Chiorescu and A. Grönlund, Validation of a CT–based simulator against a sawmill yield, *Forest Products Journal*, Vol. 50, No. 6, pp. 69–76, June 2000.
[9] T.H. Cormen, C.E. Leiserson and R.L. Rivest, *Introduction to Algorithms*, MIT Press, Cambridge, MA, 1994.
[10] J.M. Cychosz, Efficient Binary Image Thinning Using Neighborhood Maps, *Graphics Gems IV*, Academic Press, pp. 465–473, 1994.
[11] M. Fischler and R. Bolles, Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography, *Communications ACM*, Vol. 24, No.6, pp.381–395, 1981.
[12] J. Foley, A. van Dam, S. Feiner and J. Hughes, *Computer Graphics – Principles and Practice*, Addison-Wesley Publishing Company, Second Edition, Reading, MA, 1990.
[13] B.V. Funt and E.C. Bryant, Detection of internal log defects by automatic interpretation of computer tomography images, *Forest Products Journal*, Vol. 37, No. 1, pp. 56–62, 1987.
[14] R.L. Graham, An Efficient Algorithm for Determining the Convex Hull of a Finite Point Set, *Info. Proc. Letters*, Vol. 1, No. 4, pp. 132-133, June 1972.
[15] S. Guddanti and S.J. Chang, Replicating sawmill sawing with TOPSAW using CT images for a full-length hardwood log, *Forest Products Journal*, Vol. 48, No. 1, pp. 72–75, January 1998.
[16] J.G. Haygreen and J.L. Bowyer, *Forest Products and Wood Science: An Introduction*, 3rd Edition, Iowa State Univ. Press, Ames, Iowa, 1996.
[17] D.G. Hodges, W.C. Anderson and C.W. McMillin, The economic potential of CT scanners for hardwood sawmills, *Forest Products Journal*, Vol. 40, No. 3, pp. 65–69, 1990.
[18] R.E. Kalman, A New Approach to Linear Filtering and Prediction Problems, *ASME Journal of Basic Engineering*, Series D, pp. 35–45, 1960.
[19] M. Kass, A. Witkin, and D. Terzopoulos, Snakes: Active Contour Models, *Intl. J. Computer Vision*, Vol. 1, No. 4, pp. 321–331, 1988.
[20] P. Klinkhachorn, R. Kothari, R. Annavajjhala and C.W. McMillin, TRSys–Hardwood lumber grading training and remanufacturing system, *Forest Products Journal*, Vol. 44, No. 9, pp. 69–72, 1994.
[21] J. Moody, C.J. Gatchell, E.S. Walker and P. Klinkhachorn, An introduction to UGRS: the ultimate grading and remanufacturing system, *Forest Products Journal*, Vol. 48, No. 9, pp. 45–50, 1998.
[22] National Hardwood Lumber Association, *Rules for the measurement and inspection of hardwood and cypress lumber*, NHLA, Memphis, TN, 1998.
[23] L.G. Occena and D.L. Schmoldt, GRASP: a prototype interactive graphic sawing program, *Forest Products Journal*, Vol. 46, No. 11/12, pp. 40–42, November/December 1996.
[24] D.T. Pham and R.J. Alcock, Automated grading and defect detection: a review, *Forest Products Journal*, Vol. 48, No. 4, pp. 34–42, 1998.
[25] D.B. Richards, Value yield from simulated hardwood log sawing, *Forest Products Journal*, Vol. 27, No. 12, pp. 47–50, 1977.
[26] M. Samson, Modeling of knots in logs, *Wood Science and Technology*, Vol. 27, pp. 429–437, 1993.
[27] M. Samson, Method for assessing the effect of knots in the conversion of logs into structural lumber, *Wood and Fiber Science*, Vol. 25, No. 3, pp. 298–304, 1993.
[28] D.L. Schmoldt, P. Li and P.A. Araman, Interactive simulation of hardwood log veneer slicing using CT images, *Forest Products Journal*, Vol. 46, No. 4, pp. 41–47, 1996.
[29] D.L. Schmoldt, P. Li and A.L. Abbot, Machine vision using artificial neural networks with local 3D neighborhoods, *Computers and Electronics in Agriculture*, Vol. 16, pp. 255–271, 1997.
[30] M. Sonka, V. Hlavac and R. Boyle, *Image Processing, Analysis and Machine Vision*, 2nd Edition, PWS Publishing, Boston, MA, 1998.
[31] C.L. Todoroki and E.M. Rönnqvist, Secondary log breakdown optimization with dynamic programming, *Jour. Operations Research Society*, Vol. 48, pp. 471–478, 1997.
[32] C.L. Todoroki and E.M. Rönnqvist, Combined primary and secondary log breakdown optimization, *Jour. Operations Research Society*, Vol. 50, pp. 219–229, 1999.
[33] USDA Forest Service, *The South's Fourth Forest: Alternatives for the Future*, Forest Service Report No. 24, 1988.
[34] F.G. Wagner, F.W. Taylor, D.S. Ladd, C.W. McMillin and F.L. Roder, Ultrafast CT scanning of an oak log for internal defects, *Forest Products Journal*, Vol. 39, Nos. 11/12, pp. 62–64, 1989.
[35] C. Xu and J.L. Prince, Gradient Vector Flow: A New External Force for Snakes, *Proc. IEEE Conf. Computer Vision Pattern Recognition*, San Juan, Puerto Rico, pp. 66–71, June 1997.
[36] D. Zhu, R.W. Conners, D.L. Schmoldt and P.A. Araman, A prototype vision system for analyzing CT imagery of hardwood logs, *IEEE Trans. Systems, Man and Cybernetics, Part B: Cybernetics*, Vol. 26, No. 4, pp. 522–532, 1996.



Suchendra M. Bhandarkar received a B.Tech. in Electrical Engineering from the Indian Institute of Technology, Bombay, India, and an M.S. and Ph.D in Computer Engineering from Syracuse University, New York. He is currently a Professor in the Department of Computer Science at the University of Georgia where he directs the Visual and Parallel Computing Laboratory (VPCL). His research interests include computer vision, pattern recognition, multimedia systems, artificial intelligence and parallel algorithms and architectures for computer vision and multimedia.



Xingzhi Luo received a B.S. from the Department of Automation and Control, Huazhong University of Science and Technology, Wuhan, China, and an M.S. from the Institute of Automation, Chinese Academy of Sciences, Beijing, China. He received a Ph.D in Computer Science from the University of Georgia. He is currently employed as a Software Engineer at NVIDIA Corporation in Sunnyvale, CA. He is a member of IEEE.

Richard F. Daniels graduated with a B.S. in Forestry from Rutgers University and an M.S. and Ph.D. in Forest Biometrics from Virginia Tech. He is currently a Professor of Quantitative Forest Management at the University of Georgias Warnell School of Forest Resources where he teaches silviculture, and forest mensuration, modeling, and management and directs a research program in forest modeling and wood quality. He also directs the Wood Quality Consortium, a research partnership of 8 major forest industry firms and 6 research institutions with goals to improve predictions of wood quality and management.

E. William Tollner received a B.S. and M.S. in Agricultural Engineering from the University of Kentucky, and a Ph.D. in Agricultural Engineering from Auburn University. He is currently a Professor of Biological and Agricultural Engineering at the University of Georgia. His research interests include watershed assessment and modeling, modeling tools for post harvesting, water supply development and pond water balance modeling, sediment control modeling and techniques and X-ray computed tomography for soil property and food quality determination and agricultural and forestry applications.