AN ABSTRACT OF THE THESIS OF

Weicheng Zhou for the degree of Master of Science in Computer Science presented on April 28, 2004.
Title: Hardwood Lumber Grading Program.

Abstract approved:

Signature redacted for privacy.

Timothy A. Budd

Hardwood lumber is a major forest product, and board grading is an important part of its manufacturing and marketing. Computer grading programs have been used to train graders and to grade lumber for board data banks, but they have not been used to machine-grade boards in an industrial environment because of their slow performance. The Hardwood Lumber Grading Program (HLGP) was developed to quickly estimate a board's grade according to the National Hardwood Lumber Association rules by employing a pair-wise heuristic adopted from a separate board cut-up program. This paper describes how HLGP, using a board's size and defect data, locates the board's non-conflicting, clear-wood regions, uses them to calculate the yield, compares the board's size and defects with the limitations in the grading rules, and then assigns a grade. It also discusses how the original pair-wise algorithm was modified to expand the solution space without introducing too large a time complexity. Using a large board set that had been extensively evaluated both manually and by the most widely recognized computer grading program, HLGP correctly graded 1,289 out of the 1,581 boards. This translates into a misclassification rate of 18 percent, which is too high for industrial applications. However, HLGP averaged only less than a second to grade a board creating ample opportunity to improve the algorithm's accuracy before increased execution time becomes a problem.

©Copyright by Weicheng Zhou

April 28 2004
All Rights Reserved

Hardwood Lumber Grading Program

By
Weicheng Zhou

A THESIS
submitted to
Oregon State University

in partial fulfillment of
the requirements for the
degree of

Master of Science

Presented April 28, 2004
Commencement June 2004

ACKNOWLEDGEMENTS

I would like to express my sincere thanks and gratitude to Dr. Charles Brunner for his support, guidance and enthusiasm during the course of this work. I would also like to thank Dr. Timothy Budd for his time and advice on this thesis. I greatly appreciate Dr. Bella Bose for his helpful guidance during his wonderful course and being on my committee. I would also like to thank Dr. Bruce McCune for his time and being on my committee.

Thanks also go to my friends for their help and encouragement. I'm very grateful to my family for their endless love and wholehearted support.

TABLE OF CONTENTS

Pages

LIST OF FIGURES

LIST OF TABLES

Table Page

LIST OF APPENDIX FIGURES

LIST OF APPENDIX TABLES

Table Page

Hardwood Lumber Grading Program

1 Introduction

Hardwood lumber is the primary material from which many high-demand furnishings are made including finished floors, cabinets, furniture, millwork, and other household products. Hardwood sawmills produce and sell lumber which is segregated into grades based on its appearance and suitability for a given use.

Traditionally the grading process is performed by human graders. It is common for the same lumber to be graded several times in its manufacture. Further grading may occur when there are disputes between the buyer and seller in interpreting the NHLA grading rules [1]. An automated computer grading system becomes attractive when grade determinations are consistent and more accurate than human grading. Figure 1 shows a high level picture of an automated board processing system.

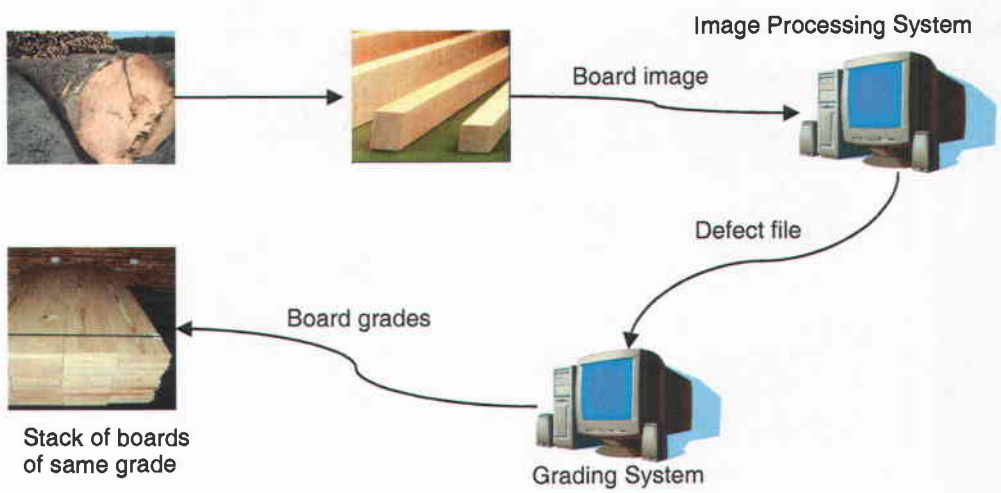

Figure 1 Automated board processing system

In the rest of this paper, we will refer to our program as HLGP. HLGP simulates the grading of hardwood lumber in accordance with the rules of the National Hardwood Lumber Association. The program was developed to provide a board grading function to a larger suite of log and board sawing simulation programs. HLGP was developed with an object-oriented structure using Visual C++ and MFC. The object-oriented structure makes any future modifications easier and the code reusable.

HLGP provides a graphical user interface that makes it unnecessary for the user to know the implementation details to use the program. HLGP displays both faces of a board in a window to show their features. To offer maximum flexibility, HLGP stores system parameters in files where they can be modified without changing the HLGP code. The system parameters include defect descriptions and grade descriptions. HLGP's output includes general information about each board's defect and grading information. It also considers *overlength* which is the fractional length beyond a board's standard length, by shifting the standard length within the board's full length to optimize board grade. In addition, HLGP applies the *first lineal foot rule* which "trims" excessive amounts of wane, stain, splits and other defects from board ends. We will discuss the details of applying this rule in section *3.7*. Refer to appendix A for hardwood grading terms.

HLGP processes the board by finding all conflict-free cuttings that meet minimum size requirements, and then assigns a grade according to NHLA rules. The first step in grading a board is to determine the grading face by grading each face without regard to defect locations on the other side. The side with the highest grade is designated the better face. While, the other side is the poorer face and, unless otherwise specified, is the grading face. The data used in

developing this program was taken from actual boards whose size and defect dimensions were recorded as coordinates in a rectangular system. Processing begins by reading the board and defect data from an input file. Some special defects, such as wane and checks, are recorded as a series of defects to minimize the area included in the defect's bounding box.

Next, all the clear regions of the board that meet the minimum dimensions must be found. A list of these areas is passed to the subroutine that solves overlapping conflicts. After any conflicts are resolved, the program compares the board's number of cuttings and basic yield with the requirements for the highest grade. The program also checks if the board's defects are within allowable limits as stated in the NHLA rules. If the board meets all the rules for this grade, the grade is assigned and processing for the board ends. Otherwise, the program compares the board attributes with the requirements for the next lower grade. Refer to appendix B for the details of hardwood grading rules.

2 Literature Review

Computer-based grading systems have developed over the past thirty years. The first hardwood grading program was developed by Hallock and Galiger in 1971[2]. However, this program lacked code for the grading of specialized species, and the inability to use the program code with other programs to evaluate and control lumber processing. One of the major limitations of Hallock and Galiger's program is its inability to consider more than one board face in the grading process. These deficiencies have been addressed by Klinkachorn et al [3-5] who developed a more sophisticated program in 1988 [4] which considers both faces. Klinkachorn's code was later incorporated into a program for training graders called HaLT [5]. An updated version based on HaLT is called HaLT2 [6]. Another program related to HaLT2 is ReGS (Realistic Grading System) [7], which can grade boards with taper and/or ¼ inch side bend or crook. One disadvantage of these programs is the rectangular modeling of the defects. A polygonal computer lumber grading program [8] models the defects on the boards as polygons. It makes the grade determination process more accurate.

The program developed in this research uses both board faces and has grade classifications for FAS, F1F, Selects, No. 1 Common, No. 2A Common, No. 3A Common and Below Grade. The program considers ¼ inch board taper and crook.

3 Program Description

3.1 User Interface

HLGP provides a graphical display with windows for displaying each board face. The user begins by opening a board data file by clicking on the *Open Board File* button. The board top face and the bottom face are then displayed. The grading rules for hardwood lumber are based on standard lengths and any fractional length (or *overlength*) may be disregarded or used as necessary. HLGP assigns *overlength* to either end of the board, or may divide the *overlength* between the ends. Red lines show the placement of *overlength* on both faces of the board. The user can left click anywhere on the board to retrieve board information, which includes board identity, length, width, and the cursor's location on the board in board coordinates. The user can left click on any defect to obtain defect information which includes defect location and type. Figure 2 shows the user interface.

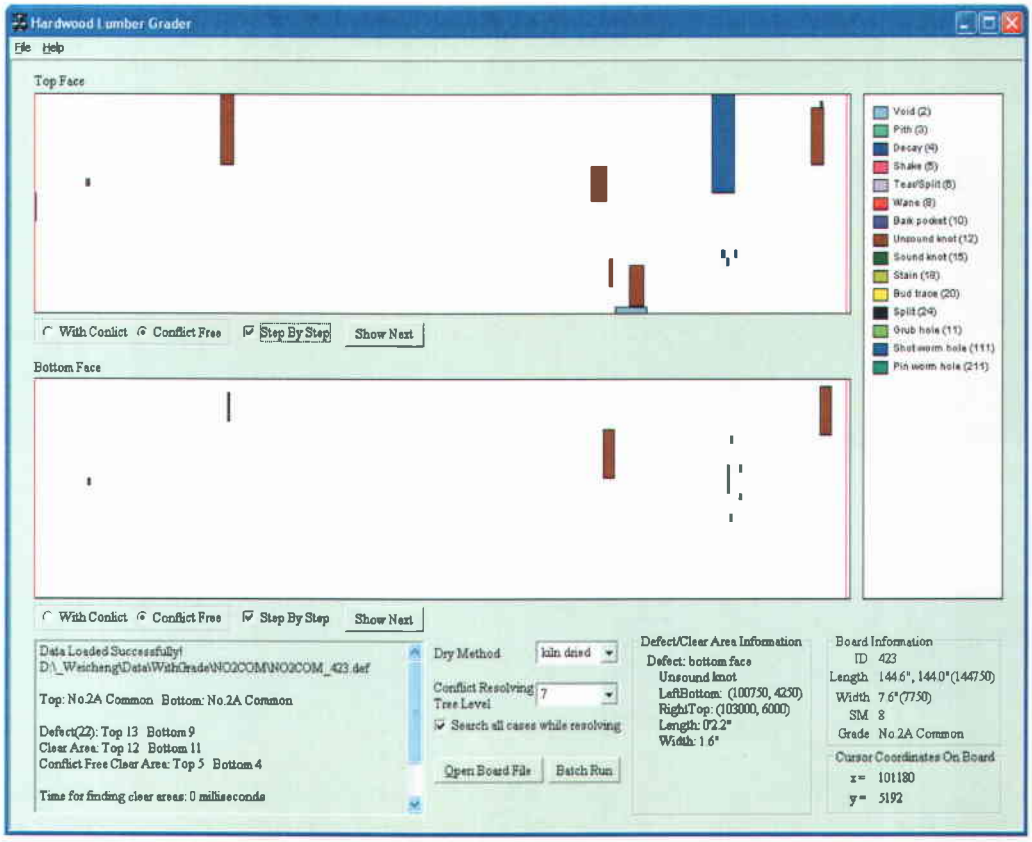

Figure 2 Display of board top and bottom faces

When the user selects the *With Conflict* radio button, the board is displayed with all the clear areas that are of minimum size. A clear area's information is presented in the *Defect/Clear Area Information* edit box whenever the cursor is within the clear area. Clear area information includes its coordinates and board face. Figure 3 shows a board with all its clear areas on each face.

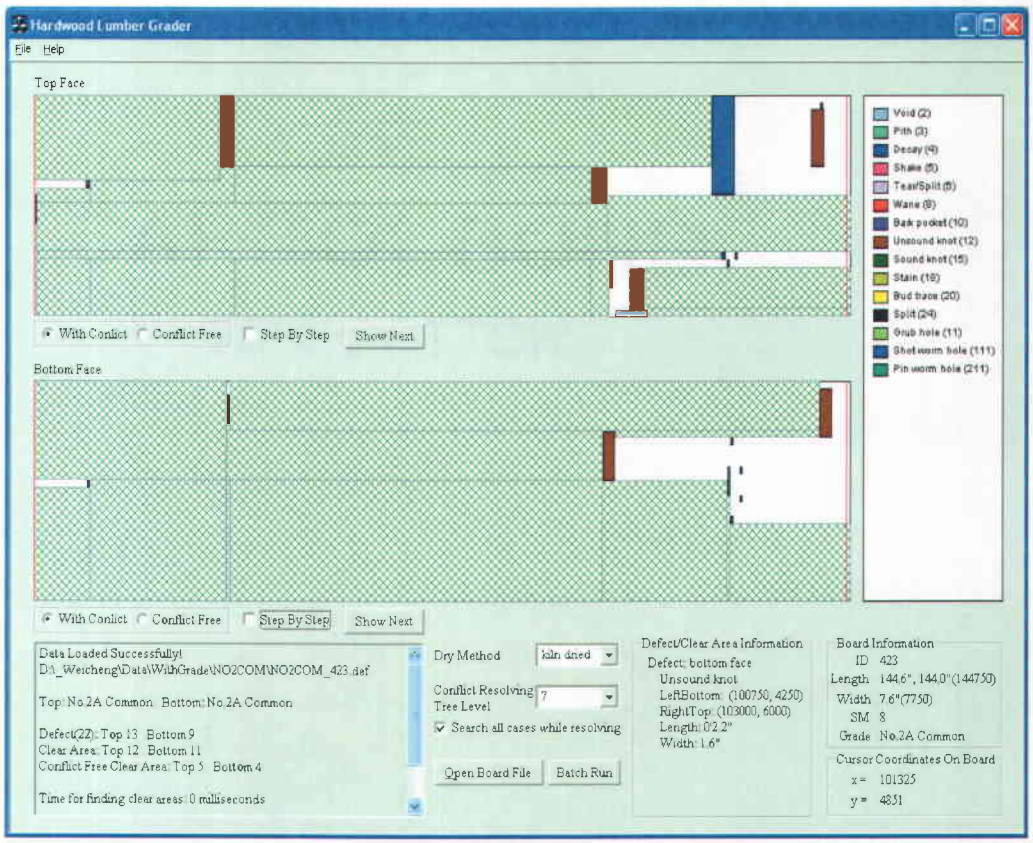

Figure 3 Display of all of a board's clear areas on each face regardless of conflicts

The user may also display each clear area in turn by selecting the *Step By Step* check box and then clicking on the *Show Next* button, which causes the clear areas to be shown one by one in the same order that they are found. Figure 4 shows this feature

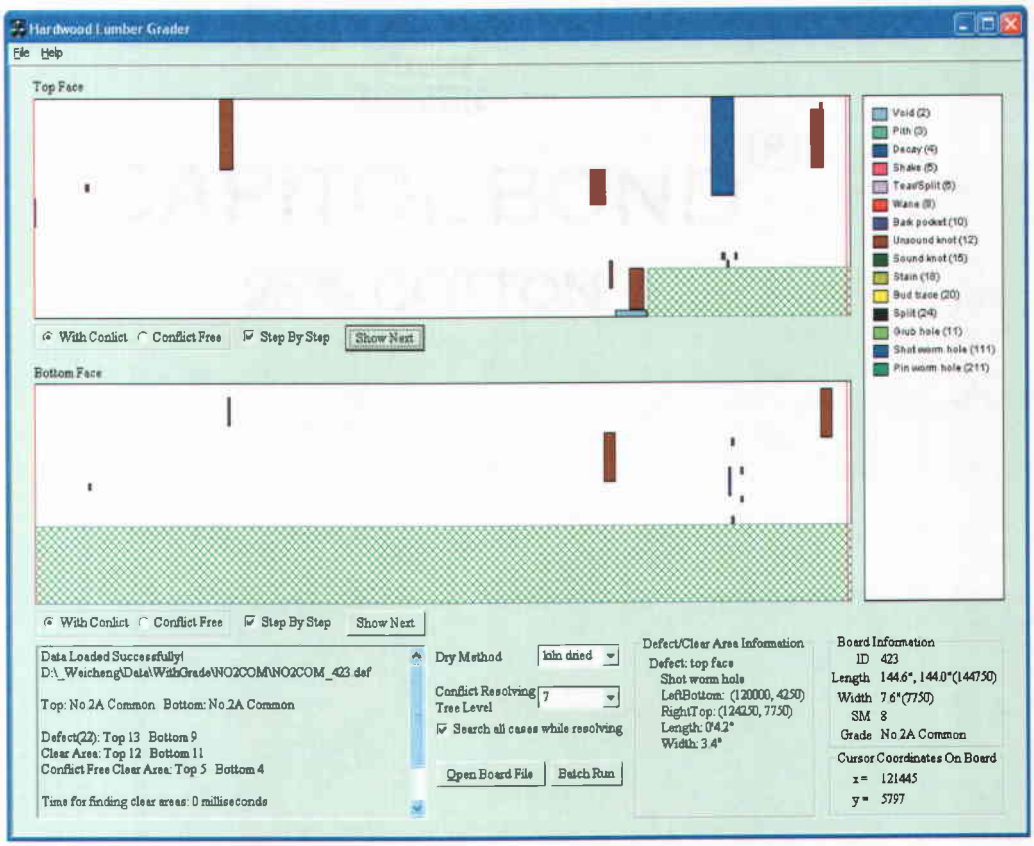

Figure 4 Display each clear area in turn

If the *Conflict Free* radio button is chosen, then the clear areas without overlap are shown (Figure 5).

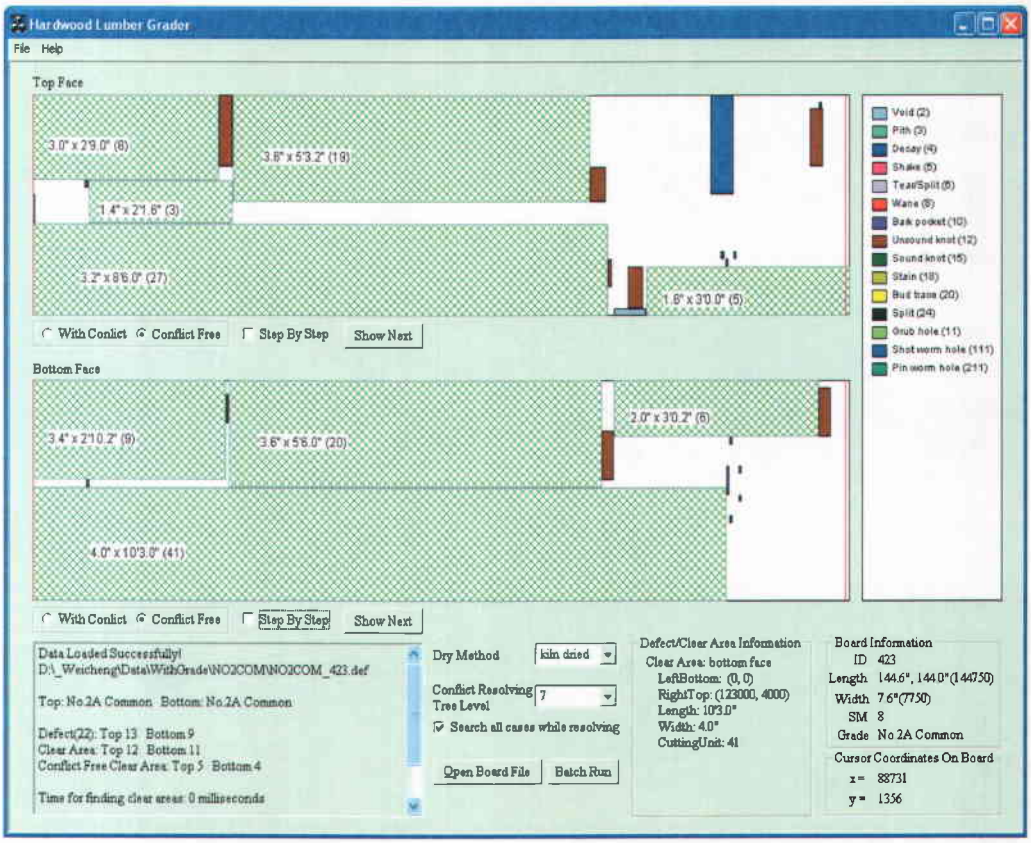

Figure 5 Conflict-free clear areas shown on each face

If the user selects the *Step By Step* check box, when the *Show Next* button is clicked, the clear areas without overlap are displayed one at a time from the largest to the smallest area.

A checklist for selecting the conflict resolution search level is also available. We will discuss these "levels" in section 3.6 *resolving clear area conflicts* subroutine. The higher "level" indicates a larger search space for more accurate grading will be used even though the program's running time will be longer.

The board grade is assigned after the user opens a defect file, or the user can click on the *Batch Run* button, and select in a directory of data files to be graded. A text area presents the board's top and bottom face grades. Also it shows the execution times for finding clear areas, resolving conflict clear areas and grading the board.

3.2 HLGP Classes

The key to being most productive in object-oriented design is to make each object responsible for carrying out a set of related tasks. If an object relies on a task that isn't its responsibility, it needs to manipulate an object of another class whose responsibilities include that task. HLGP uses 9 classes and the relationships between these 9 classes are shown below.

- *CBoard:* An object of the *CBoard* class has two faces which have clear areas, defects, conflict-free clear areas, etc. An object of the *CBoard* class is used to determine the board's grade.
- *CSection:* An object of the *CSection* class represents a board face. It contains its own bounding box, a list of defects, a list of clear areas and one or more lists of conflict-free clear areas. An object of the *CSection* class is also used to find its clear areas and resolve clear area conflicts.
- *CDefectList:* An object of the *CDefectList* class maintains a list of defects. It provides the functionalities of adding a defect, removing a defect, testing the existence of a defect, etc.
- *CDefect:* An object of the *CDefect* represents a defect. It has its bounding box and functionalities such as comparing with another defect for equality, modifying its coordinates, etc.

- *CClearArea:* An object of the *CClearArea* class represents a clear area. It has a bounding box in the board face's coordinate system. It contains methods to compute its area, yield, etc.
- *CConflictResolver:* An object of the *CConflictResolver* class is used to resolve clear area conflicts among a list of clear areas and return the resolved clear areas. It is used by the *CSection* class.
- *CGradeHunter:* An object of the *CGradeHunter* class maintains a pointer to a board object. When the board issues the grade command, the *CGradeHunter* object reads the clear area and defect information from the board, applies the grading rules and outputs the board grade.
- *CSystemParameterServer:* An object of this class is created when the program starts. The object opens the SystemParameter.ini file and reads in the parameters such as defect descriptions and grade rules. It provides the functions to access these values.
- *CGraderView:* An object of this class is created by MFC. It is responsible for the user interface and all the drawings. It maintains a pointer to the *CBoard* object and a pointer to the *CSystemParameter* object. When the *CGradeView* object is being initialized, the *CSystemParameter* object is created. The *CGradeView* is responsible for all the interaction with the user.

3.3 Program Flow Chart

HLGP first reads the board and defect coordinates from the data file. Using these data, the program finds all the rectangular areas within the board's boundaries that do not contain defects. It then resolves conflicts between clear areas that overlap each other's areas. The program uses the conflict-free clear areas to calculate the board's cutting units. It also compares the board's size and

defects with the requirements for the various grades. Figure 6 shows the program's flow.

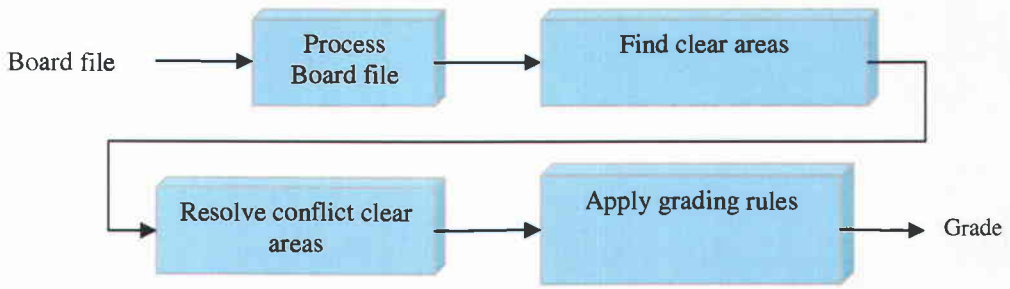

Figure 6 Program flow chart

3.4 Reading Board Data Subroutine

The input to HLGP is a text file containing descriptions of the board and its defects. The description includes the lower left and upper right coordinates of the bounding box for the board, and its defects, the defect's type, and the face on which it appears. The data used are from actual boards. The board's x coordinates represent the length and the y coordinates represent the width. The board's lower left-hand corner is typically at the origin (0,0). Figure 7 shows a board with defects.

Figure 7 Board image with defects

While most are recorded as two pairs of x, y coordinates, some defects may be recorded as a series bounding boxes to reduce the clear area included in a single bounding box. Such defect types commonly treated this way are wane, checks, pitch streak, and splits. HLGP determines the length and width of a stepped defect by treating all touching bounding boxes of the same type as a single defect. Figure 8 and Figure 9 demonstrate how a defect's area, wane in this case, is reduced.

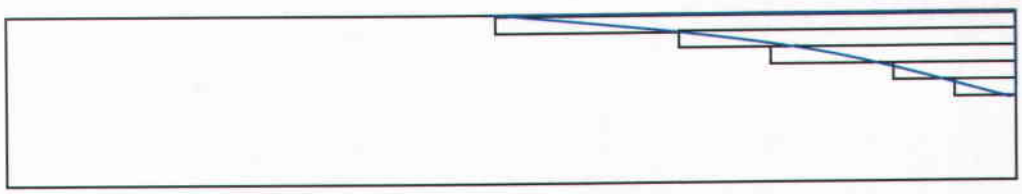

Figure 8 Wane recorded as stepped rectangles

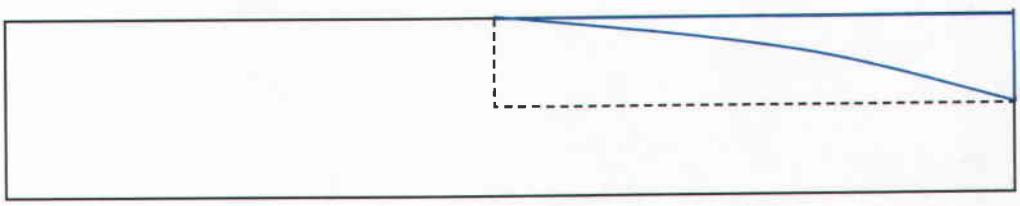

Figure 9 Wane recorded as a single rectangle

3.5 Finding Clear Areas Subroutine

The program uses the board and defect coordinates to find all the clear areas that meet the minimum cutting size. This section describes the program's procedures for completing this task. The terminology used in this section is as follows:

Clear Area - A rectangular portion of a board without defects, and it is bounded on each of its four sides by either a defect or a board edge. The clear areas on each board face are considered separately as the clear areas on the reverse face are ignored.

Conflicting Clear Areas – Clear areas with some area in common with another clear area.

Defect bottom group - A group of defects with the same bottom coordinate that, therefore, determines the top of a clear area.

Defect top group - A group of defects with the same top coordinate that determines the bottom of a clear area

Minwidth – The minimum allowable width for a clear area

Minlength – The minimum allowable length for a clear area

3.5.1 Data Structure for Finding the Clear Areas

Clear areas are found by searching the board from lower right to upper left. Four sorted arrays contain the defect top, bottom, left, and right coordinates in ascending order. A defect's right/left edge defines a clear area's left/right edge, and a defect's top/bottom edge defines a clear area's bottom/top edge. The board is treated as a special defect whose left edge may define a clear area's left edge and whose right edge may define a clear area's right edge. Figure 10 illustrates the clear area left and right arrays for a simple sample board. In Figure 10, calft and carht denote the two arrays holding clear area left and right edges, respectively. The edges indicated by black arrows are saved in the left clear area array (calft), and the edges indicated by the blue arrows are saved in the right clear area array (carht).

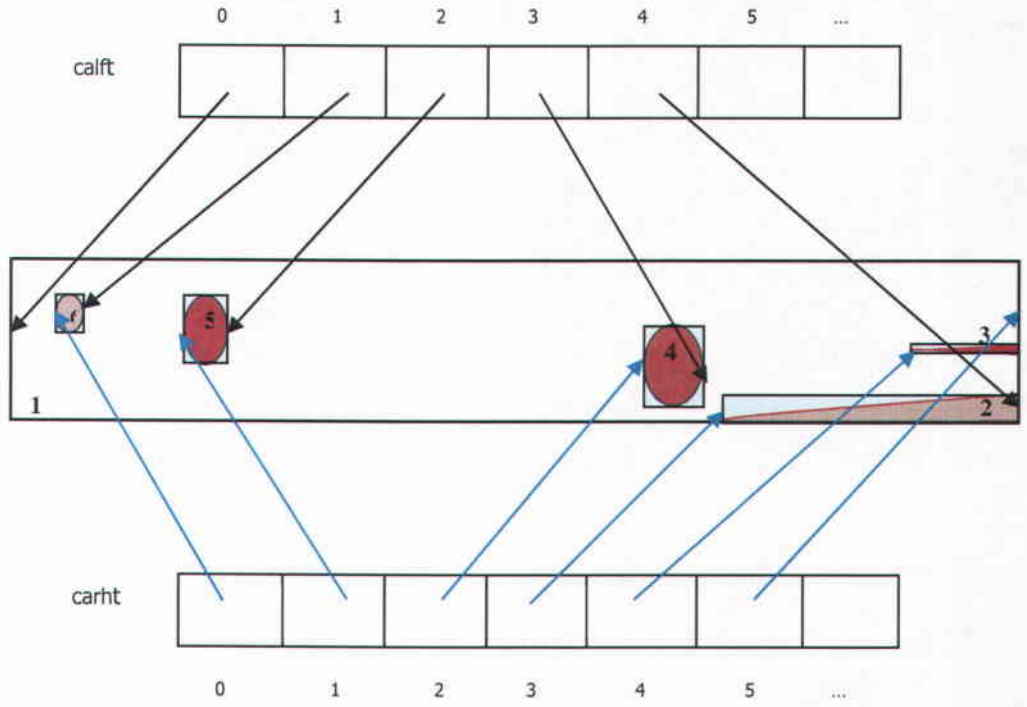

Figure 10 The clear area left and right arrays

The arrays designated as *Defect Top Group*s and *Defect Bottom Group*s contain one or more defects with equal top and bottom coordinates, respectively. Figure 11 shows a sample board with 5 defects. Note that the board is considered defect number 1. In the sample board, *defect top group*s are ordered as defect number 1, 2, 3, 4, 5 and 6. D*efect bottom group*s are ordered as defect number 2, 4, 5, 3, 6, 1.

Figure 11 Board image with defect bounding boxes

3.5.2 Algorithm for Finding Clear Areas

The search for clear areas starts at the board's bottom and proceeds to its top. Figure 12 – 14 shows how this algorithm finds the clear areas on our sample board. Suppose we draw a horizontal line on the coordinate of the minimum width for a clear area (*LineOne* in Figure 12). The defects that partially or completely fall between the board bottom and *LineOne* are called *CurrentDefects*. The *CurrentDefects*' left and right edges determine the right and left edges of the clear areas, respectively. *CurrentDefects* are numbers 2 and 4 in Figure 12 - 14. The left and right coordinates of the *CurrentDefects* are saved in the clear area right and left arrays.

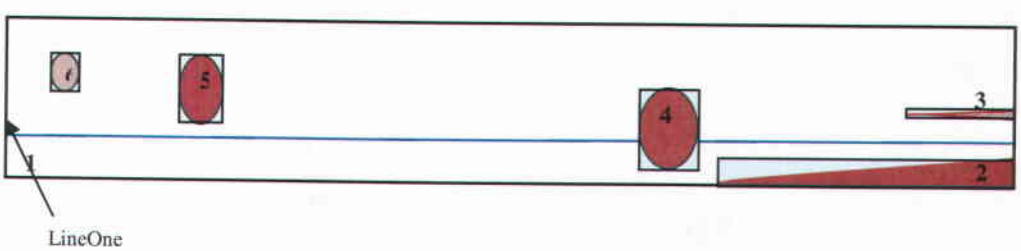

Figure 12 Board image showing CurrentDefect number 2 and 4 between the board bottom and the LineOne

The board's bottom edge forms the first *defect top group* which determines the bottom of the clear area. Now we are trying to find the top of the clear area. The

bottom of defect number 5 is the next coordinate above *LineOne* and is the current *defect bottom group* defining the top of the clear area. Figure 13 shows where the first clear area is located.

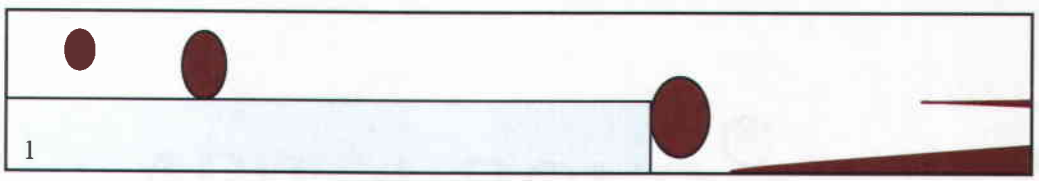

Figure 13 Board image showing clear area 1 of 4 possible clear areas

Once a defect in the *defect bottom group* has been used in the search process, it will be added to *CurrentDefects*, because its left and right coordinates may affect the right and left coordinates of the next clear area in the search. Therefore, defect number 5 is added to *CurrentDefects*, and its left/right coordinates are saved in the clear area right and left arrays, respectively.

As the program continues to search for clear areas, the bottom of the defect 3 is the next coordinate in the array and becomes the second current *defect bottom group*. The program determines that the distance between the right of defect number 4 and the left of defect number 2 does not meet the minimum clear area length. The same circumstance occurs when defect 6 becomes the current *defect bottom group.* The board is the last *defect bottom group*. Figure 14 shows the board after finding the second clear area having its top edge equal to the board's bottom edge.

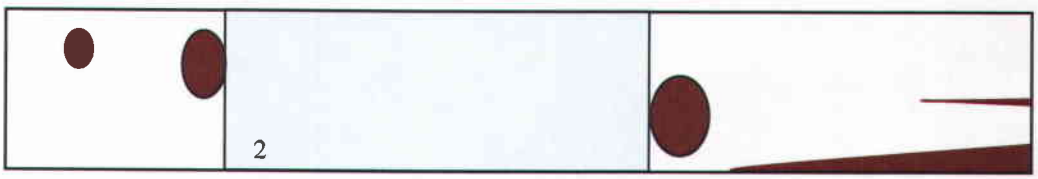

Figure 14 Board image showing clear area 2

This process is repeated until the distance between the top of the board and the coordinate of the *defect top group* is smaller than the minimum clear area width. Figure 15 and Figure 16 show the third and fourth clear areas in the board that meet the minimum size requirements.

Figure 15 Board image showing clear area 3

Figure 16 Board image showing clear area 4

A *ClearArea* object is created once a clear area is found and saved in its *CSection* class. The *CClearArea* objects are stored in the order they are found. A value is assigned to each clear area, which are then sorted into descending order and put into a linked list. Figure 17 shows the board's four possible clear areas

that meet the minimum width and length requirements.

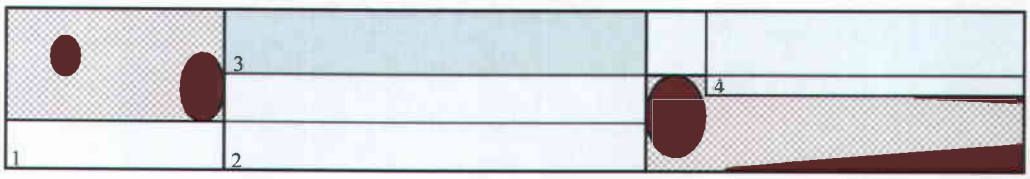

Figure 17 Board image showing all 4 clear areas

3.6 Resolving Clear Area Conflicts Subroutine

An individual clear area is also related geometrically to other board clear areas. The relationships fall into one of three conflict categories, *form1*, *form2* and *form3*. *Form1* is where the clear areas have no overlap in either width or length , meaning the clear areas are independent and have no conflicts. The other two forms involve overlapping areas and will be described in the section *3.6.1*.The program resolves the overlapping conflicts using two possible solutions. The methods *Left Recovery* and *Right Recovery* resolve the overlap for *form2* and methods *Long Salvage* and *Wide Salvage* resolves the overlap for *form3*. After all the conflicts between individual cuttings are resolved, the program selects a solution from the clear area list. The flow chart for resolving overlapping clear areas is shown in Figure 18.

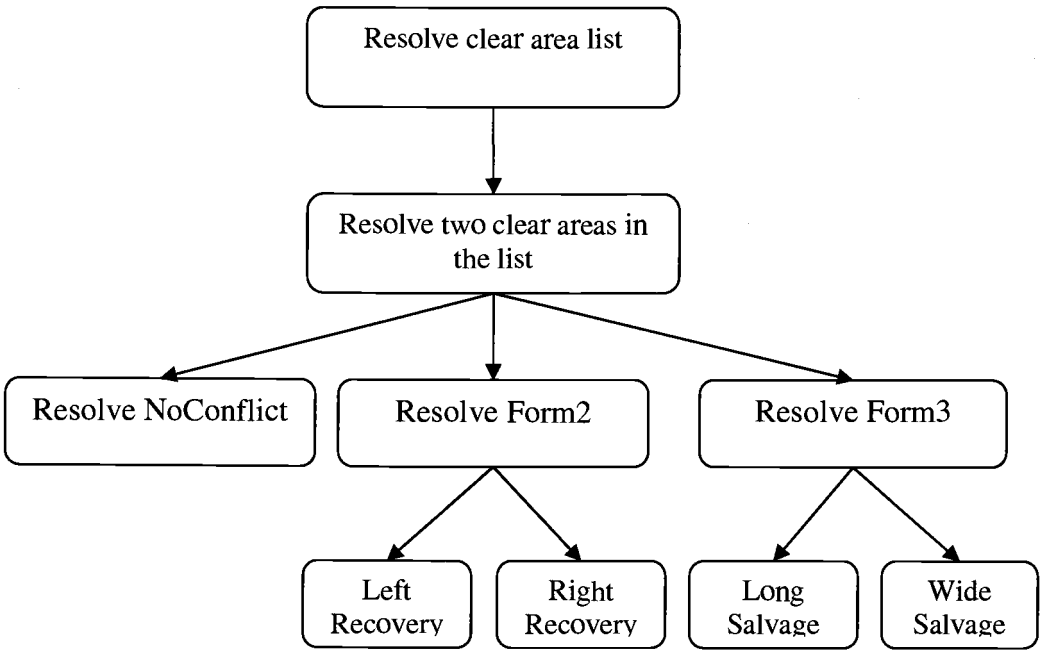

Figure 18 Flow chart for resolving clear area conflicts

3.6.1 Two Conflict Forms

Two general classes of area overlap are a "partial overlap" (*form2*) and a "complete overlap" (*form3*). A "partial overlap involves an overlap in which clear areas do not overlap each other completely in either width or length. Figure 18 shows different forms of a "partial overlap".

Figure 19 Different forms of a "partial overlap"

A "complete overlap" involves the complete overlap of two clear areas width or length. Figure 19 shows different forms of a "complete overlap".

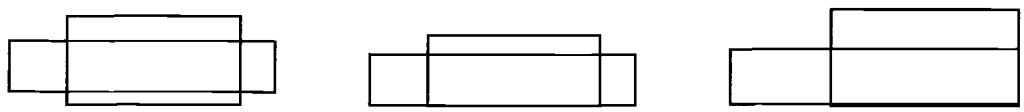

Figure 20 Different forms of a "complete overlap"

3.6.2 Algorithm for Resolving the Conflict between Two Clear Areas

3.6.2.1 Resolving Partial Overlap Using Left Recovery

The conflict-free area available from Left Recovery is calculated by adding any salvage (non-overlapping) areas from the right cutting to the entire area of the left cutting. See the example in Figure 20. The salvage areas are determined by dividing the right cutting with a rip line (y_1) into one area with overlap (①) and the one without overlap (②). The length of area is reduced by its overlapping portion before the dimensions of both salvage areas are checked to insure they meet the minimum cutting size. All viable cuttings are then saved to the appropriate list.

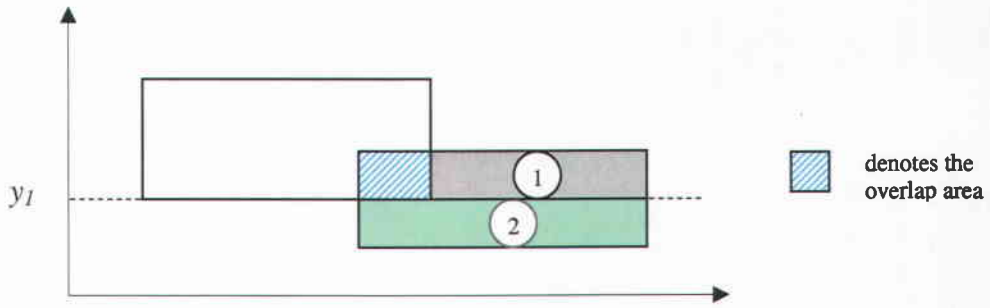

Figure 21 Resolving the conflict while saving the entire left cutting

3.6.2.2 Resolving Partial Overlap Using Right Recovery

The conflict-free area available from Right Recovery is calculated in the same manner as Left Recovery except any salvage areas from the left cutting are added to the entire area of the right cutting. See Figure 21 for an example.

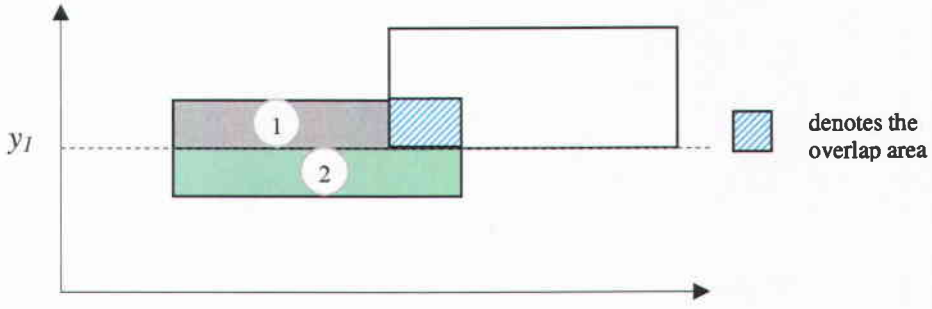

Figure 22 Resolving the conflict while saving the right cutting

3.6.2.3 Resolving Complete Overlap Using Long Salvage

The conflict-free area available from Long Recovery is calculated by adding any salvage areas available from the wide cutting with a pair of rip lines (y_1, y_2) that follow the top and bottom edges of the long cutting. See Figure 22. This produces three areas, one that fully overlaps the long cutting and is not considered, and two potential salvage areas adjacent to the top (①) and bottom (②) edges of the long cutting. If these salvage areas meet the minimum cutting size, they are saved to the appropriate list.

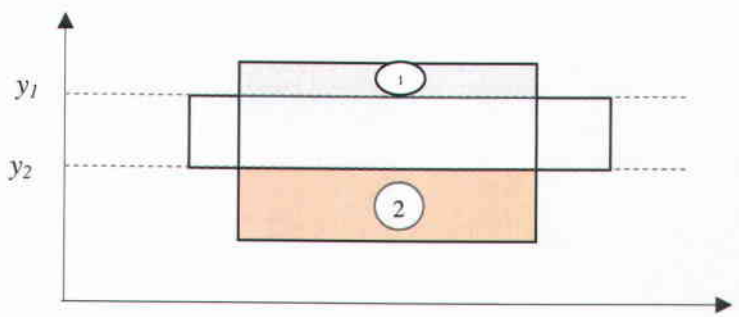

Figure 23 Resolving the conflict while saving the longer cutting

3.6.2.4 Resolving Complete Overlap Using Wide Salvage

The conflict-free area available from Wide Recovery is calculated by adding any salvage areas from the long cutting to the entire area of the wide cutting. The salvage areas are determined by dividing the long cutting with a pair of crosscut lines(x_1, x_2) that follow the left and right sides of the wide cutting. See the example in Figure 23. This produces three areas, the one that fully overlaps the wide cutting is ignored, and two potential salvage areas adjacent to the left (①)

and right (②) sides of the long cutting. If these salvage areas meet the minimum cutting size, they are saved to the appropriate list.

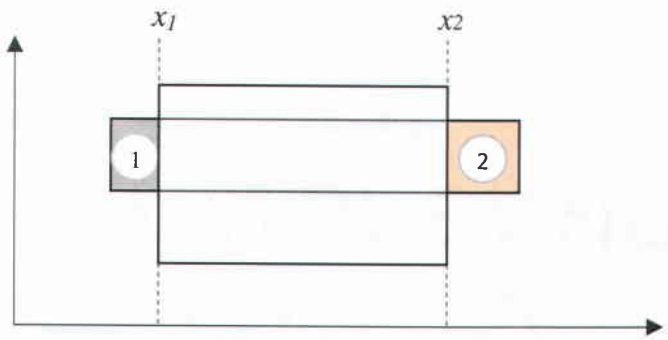

Figure 24 Resolving the conflict while saving the longer cutting

3.6.3 Resolve the Conflict Clear Area List

This section describes how conflicts are resolved among the whole clear area list. The clear areas which meet the minimum cutting size are stored in the list in descending order. Before resolving any conflicts, the conflict type between the first two clear areas must be determined. The program resolves conflicts using the methods described in section *3.6.2*. Once any conflicts are resolved between the first two clear areas, the program will resolve the conflict between the first and the third clear areas, and so on, until there are no conflicts between the first clear area and other elements in the list. The program then processes the second clear area with the other clear areas in the list in a similar manner. Each time one of the two conflicting clear areas is divided into smaller cuttings inserted into the list in descending order. The program continues until all the clear areas have been processed for possible conflicts with each other. This produces a new clear area list in which the clear areas are conflict-free. Refer to appendix C for the pseudo code for resolving clear area conflicts.

3.6.4 Procedures for Selecting Conflict-Free Solutions

The original procedure for selecting conflict-free combinations was based on the assumption that selecting the combination with the largest total conflict-free area would typically result in assigning the highest grade possible. However, after grading 1,581 boards of known grade, HLGP consistently assigned lower than correct grades with a misclassification rate of 43 percent. This high error rate occurred because the original assumption frequently prevented subsequent, potentially better, clear area comparisons from even being considered. For example, a solution that retains the entire clear area *A* along with the salvage from its conflict partner clear area *B* precludes considering any overall solution containing the entire area of *B*.

Therefore, the assumption was discarded and a new procedure written that retains both solutions from a pair-wise comparison. This results in a binary tree structure where each node represents a clear area list (See Figure 24). The tree's root contains the entire clear area list generated by the F*inding Clear Areas* subroutine along with pointers to two children. One child contains the parent's list after reducing one member of the largest clear-area pair to salvage areas, while the second child contains the list after reducing the other member to salvage areas. Salvage areas are inserted so as to maintain the descending order by clear-area size. The procedure continues until there are no more clear areas to process at which point each terminal leaf's list is evaluated to determine the highest possible grade for a board face.

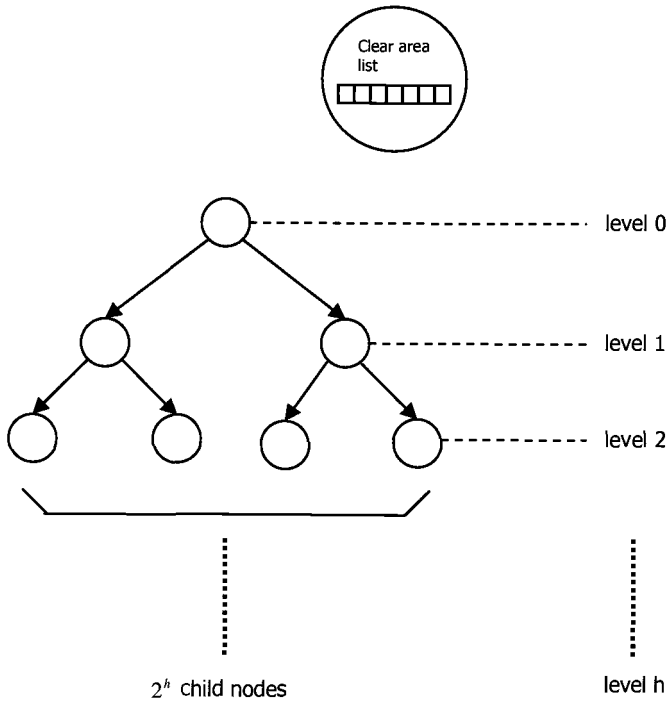

Figure 25 A binary tree is to store 2^h clear area lists

While this approach lowers misclassifications, it also consumes extensive amounts of memory and time because a tree of height h (0 to h node levels) produces 2^h lists to evaluate. The number of lists became quite large with our test boards which caused some of them to use all the available memory and 14 percent of the remaining ones to take over five minutes to be graded. Some type of modification was clearly desirable to alleviate these problems. Fortunately, only a few of the largest clear areas are generally needed for grading, so the height of the tree can be reduced without greatly increasing the error rate. To control tree height, the procedure was modified to include a new variable that specifies the maximum number of allowable node levels. The modified program operates normally until it reaches the specified node level at which point any remaining clear-area conflicts are resolved using the original largest total clear

area solution. Notice that using the 0 level is the same as using the original procedure and using the unlimited level is the same as the unaltered new procedure.

HLGP permits the user to select 0 through 7, or unlimited node levels for the tree's structure. The effect of tree height on error rate and execution time was explored by varying the node level and recording the results.

The grade classification for 198 FAS boards in different "levels" is shown in Table 1. In the analysis, we combined FAS ONE FACE (F1F) grade and FAS into one grade. We will explain the reason in section *4 Performance and Results*.

Table 1 The misclassification rates of 198 FAS boards in different "levels"

Levels	Boards Misclassified	Board Misclassification Rate
Level 0	26	13.31%
Level 1	13	6.56%
Level 2	8	4.04%
Level 3	7	3.53%
Level 4	6	3.03%
Level 5	5	2.52%
Level 6	5	2.52%
Level 7	5	2.52%

3.7 Grading the Board Subroutine

This subroutine uses the clear area list received from *Resolving Clear Area Conflicts* to grade the board. The standard hardwood lumber grades are FAS, FAS 1 Face (abbreviated F1F), Selects, No. 1 Common, No. 2A Common, No. 3A Common, No. 2B Common and No. 3B Common. Each standard grade has the minimum board requirements that include minimum board size, minimum cutting size, basic number of cuttings and basic yield. Additional considerations include supplemental cuttings and yields as well as defect limitations for the grade. See appendix for details.

The procedure for grading lumber is as follows:

1. Check board length and width for minimum board size of the highest grade.
2. Determine the Surface Measure using the standard board length and actual board width.
3. Determine the grade face of the board, usually the poorest face, except when otherwise specified. The poorest side of the board has the smallest clear area, or the lower grade.
4. Determine the number of cuttings and the minimum cutting size permitted for the grade.
5. Determine the clear-face cutting units (SM × the multiplier for the grade).
6. Calculate the total area of clear-face cutting units on the poorest face.
7. If the board does not yield sufficient clear-face cutting units, consider the next lower grade.
8. Check the defect limitations for the grade.
9. Each board grade is assigned.
10. Assign a board grade if the reverse side of the grading cuttings is sound.

The grading rules for hardwood lumber are based on standard whole-foot length, so any additional fractional length (or *overlength*) is disregarded. Standard lengths for hardwood lumber are 4', 5', 6', 7', 8', 9', 10', 11', 12', 13', 14', 15', and 16'. HLGP optimizes for *overlength* by assigning *overlength* to either the right or the left end of the board, or dividing it into any amount between both ends. HLGP repeatedly grades the board to determine optimal *overlength* placement for the board.

According to NHLA grading rules, the first foot of either end of a board will be considered for FAS as follows:

"Within one lineal foot from the ends of the boards of standard lengths there shall not be less than 50% clear-face in not more than two pieces of any shape. And in addition, there shall not be less than 25% of sound wood in the aggregate [1]."

HLGP adds up all non-clear areas and checks that the total does not exceed 50%. Then, HLGP also adds up all non-sound areas and checks that the total does not exceed 25%. Any non-sound area within the board's *overlength* may be disregarded.

4 Performance and Results

HLGP has been tested using the USDA Forest Service databanks [9]. The grades represented in this databank are FAS, Selects, No. 1 Common, and No. 2A Common. The boards are all one inch thick. Among them, 198 boards are graded as FAS, 209 as Selects, 593 as No. 1 Common and 581 boards as No. 2 Common. These boards were manually graded by expert graders and by the computer program ReGS [4]. The grading results for HLGP are shown in Table 2 where the matrix of actual versus predicted classifications by different grades is given.

Table 2 Confusion matrix for grade classification

Actual Classification	Predicted Classification					
	FAS	Selects /F1F	No.1 Com	No.2 Com	No.3 Com	BG
FAS	192	0	6	0	0	0
Selects	18	166	20	5	0	0
No.1 Com	3	38	461	88	33	0
No.2 Com	0	1	20	470	74	16

Table 2 shows that the FAS boards are sometimes misclassified as No.1 Common grades. Selects broads are sometimes misclassified as FAS and No.1 Common. Few of them are misclassified as No.2 Common. No.1 Common grades are sometimes misclassified as FAS, Selects, No.2 Common and No.3 Common. No.2 Common grades are sometimes misclassified as Selects, No.1 Common, No.3 Common and Below Grade.

FAS ONE FACE (F1F) is a new standard NHLA grade. F1F and Selects require the better face to be FAS and the reverse side to be No. 1 Common. For F1F and Selects, the faces are graded independently and there is no requirement that the reverse side of the grade cuttings be sound. HLGP places each board in the highest grade possible. Therefore, the boards placed in Selects before may meet the requirements for F1F. For this reason we combined F1F and Selects into a F1F/Selects category. Table 4 shows the rates of misclassification for each board grade.

Table 3 Misclassification rates for each board grade

Board Grade Type	Overall Misclassification Rate
198 FAS Boards	3.03%
209 Selects Boards	20.57%
593 No.1 Common Boards	22.26%
581 No.2 Common Boards	19.10%
Overall 1,581 boards	18.47%

Possible reasons for the lower than anticipated grades are that some cuttings used by HLGP for grading are not large enough for the grade, or if the cutting size is large enough, the reverse side of the cutting is not sound as required for the FAS, No.1 Common, No.2A Common, and No. 3A Common grades. Or the board drops a grade because of defects.

We have manually checked some of the boards and found the HLGP results to be correct.

The time complexity of *Resolving Conflict Clear Areas* is $O(n^2 \times 2^h)$, where n is the number of clear areas in the list, h is the node level of the tree's structure. The time complexity of *Grading the Board* is $O(2^{2h})$. The average total execution time per board of this program is 972 milliseconds when level 7 is specified (See Table 4). From the results, we found most of the execution time is spent in resolving clear area conflicts and grading the board. The program was executed on a computer with a AMD 2500+ CPU and 1 gigabyte of memory.

Table 4 The average execution time

Boards	Average Time for Finding Clear Areas (milliseconds)	Average Time for Resolving Clear Area Conflicts (milliseconds)	Average Time for Grading (milliseconds)	Total Average Time Spent (milliseconds)
198 FAS	0.23	46	145	192
209 Selects	0.37	82	360	442
593 No. 1 Com	0.39	134	1174	1309
581 No. 3 Com	0.65	193	1753	1946

5 Conclusion

Hardwood lumber is an important forest product and automated lumber grading systems could improve the productivity of lumber producers. HLGP analyzes a board by finding its clear cuttings and assigning a grade according to NHLA rules. The program manages all input and output operations providing any information needed to accomplish these functions. To determine the grade cuttings, all regions of the board that are clear on one side and meet the minimum dimensions are found. These clear areas may overlap, but the program resolves these conflicts. The resolved cuttings are also checked to see if they meet the cutting unit requirements. The program uses an enhanced limited search algorithm in the *Resolving Clear Area Conflicts* subroutine to improve the results and avoid a large time complexity. The user can control the trade off between accuracy and time by selecting a search "level", which is from "level 0" to "level 7", or unlimited, for the tree's structure. The conflict free cuttings are used in the *Grading the Board* subroutine to determine a face's grade by comparing cutting area with grade requirements. To determine the board grade, the program checks if the reverse sides of the grade cuttings are free of unsound defects and on any defect limitations. If the board's grade face meets the grade requirements, the program assigns a grade. If it does not, then the board is compared to the next lower grade requirements.

HLGP uses object-oriented technology that will be incorporated into a larger simulation system. An object-oriented structure makes each object responsible for carrying out a set of related tasks. For example, before a board object determines its grade, it needs to have access to a board face object, which

finds clear areas and resolves clear area conflicts. The board face object then sends a message to a clear area object to calculate all possible board grading parameters, such as cutting units and surface measure. Both design and debugging are simplified when building small objects that perform a few tasks, rather than large objects with internal data that are extremely complex, with hundreds of functions to manipulate the data. The object-oriented design maximizes reusability, reduces data dependency and minimizes debugging time. Most of the classes can be either reused directly, or derived into child classes. The existing code is also easy to modify, maintain, and to use for developing new functionalities.

6 Future Work

The purpose of this research was to develop a program to grade boards using the NHLA grading rules. HLGP with its relative accuracy and speed can be used for educational/training purposes to illustrate most of the grading rules.

HLGP is designed to work only on boards that are rectangles. To avoid the negative effects of crook or side bend on yields, the board data used in this program has no more than ¼ inch of crook or side bend. Many real boards have more than ¼ inch of crook, so this is one limitation of the program. HLGP is also not designed to handle tapered boards, but future modifications will provide this capability. A board editor is also planned for the graphical user interface. It will permit the user to create boards and defects using either the keyboard or the mouse. These features will make the program more flexible and useful for the user.

The initial objective was to develop an automatic computer system for grading hardwood lumber, because these grades are more precise than softwood grades and are, therefore, easier to implement by computer. However, softwood is widely used for interior walls, window, sash, doors, architectural woodwork, cabinets and hundreds of other standard and special applications. The ability to implement the softwood grading rules written by the Western Wood Products Association is planned because it would be very useful. One of the largest markets is for *Shop* grades of lumber used for windows, sashes and doors, so the softwood grading program will initially implement the Shop lumber grading rule and then will be integrated into the HLGP program.

Bibliography

[1] Rules for the measurement and inspection of hardwood and cypress, 1990 & 1994 *National Hardwood Lumber Association, Memphis, TN.*

[2] Hiram Hallock, Lynn Galiger, 1971. **Grading Hardwood Lumber by Computer.** *USDA Forest Service Research Paper FPL 157.*

[3] Powsiri Klinkhachorn, J. P. Franklin, C. W. McMillin, R. W. Conners and H.A. Huber. 1998. **Automated computer grading of hardwood lumber.** *Forest Products Journal, 38(3): 67-69*

[4] Charies Gatchell, Powsiri Klinkhachorn, Ravi Kothari, 1992. **ReGS – a realistic grading system.** *Forest Products Journal, Vol. 42, No. 10.*

[5] Moody J., C. Gatchell, P.Klinkhachorn, and B. Walker, 1998. **An Introduction to UGRS: The Ultimate Grading and Remanufacturing System**, *Forest Products Journal, Vol. 48, No.9, September 1998, pp 45-50.*

[6] C.J.Schwehm, C.W.McMillin and H.A.Huber. 1989. **HaLT: A hardwood lumber training program for graders**. *Forest Prod. J. 39(2):38-40.*

[7] C.W.McMillin, R.Kothari, and D.Yost. 1992. **HALT2-an enhanced lumber grading trainer**. *Forest Prod. J. 39(2):38-40.*

[8] P.Klinkhachorn, R.Kothari, D.Yost, and P.Araman. 1992. **Enhancement of the computer lumber grading program to support polygonal defects**. *Forest Prod. J. 42(10):41-46.*

[9] Charles J. Gatchell, R. Edward Thomas, Elizabeth S. Walker, 1998. **1998 Data Bank for Kiln-Dried Red Oak Lumber**. *United States Department of Agriculture, Northeastern Forest Experiment Station.*

[10] Charles J. Gatchell, Janice K. Wiedenbeck, Elizabeth S. Walker, 1992. **A red oak data bank for computer simulations of secondary processing**. *Forest Products Society. J. 3(6):38-42*

[11]1998, **Understanding Hardwood Lumber Grading.** www.woodweb.com *Independent Sawmill & Woodlot Magazine*

APPENDICES

Appendix A Hardwood Grading Terms

The following are some basic definitions.

- **Log**s are the main stems of trees which are sawn into lumbers (Figure 25).

Figure 26 Log

Figure 27 Dimension Lumber

- **Lumber** consists of piece of wood cut to size for building, furniture, etc (Figure 26).
- **Board** is a single piece of lumber (Figure 27).
- **Wide face**s are the two largest surfaces of a dimension lumber.
- **Top face** and **bottom face** are the two wide faces of a board – if one of them is called top face, the opposite wide face is called bottom face and vice versa (Figure 27).

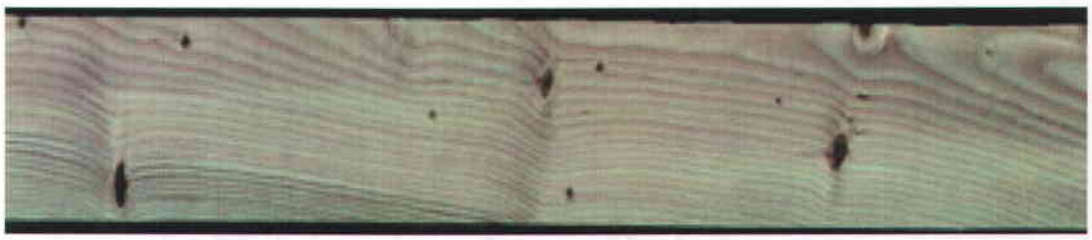

Figure 28 Top face and bottom face

- **Edge** is the intersection of a wide face and a narrow face (Figure 28).

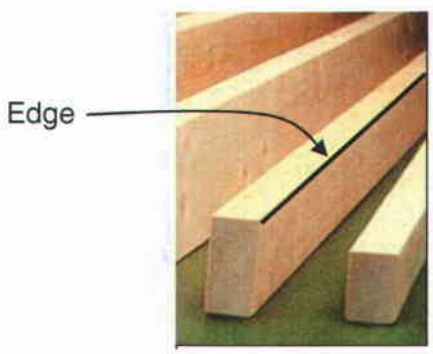

Figure 29 Board edge. The edge in the picture is highlighted using a black line

- **Defects** are the imperfect characteristics of a board that result from branches, decay, physical damage, etc. Typical defects include knots, blue stain, brown stain, splits, wane, bark pockets, checks, etc (Figure 29).
- **Wane** is where the board was cut near the outside of the trunk and an edge contains bark or is not squared (Figure 29).

Knot Wane Bark pocket

Figure 30 Defects

- **Knot**s are a portion of a branch that remains in a board (Figure 29).
- **Stain** is a bluish gray discoloration on the wood surface. This feature is most common in woods like Holly, Pine, and Sycamore. (Figure 30).

Blue Stain Brown Stain

Figure 31 Stain

- **Check**s are breaks in the wood normally occurring across the annual growth rings.
- **Splits** are breaks of the wood through the piece to the opposite surface or to an adjoining surface due to the tearing apart of the wood cells. (Figure 31)

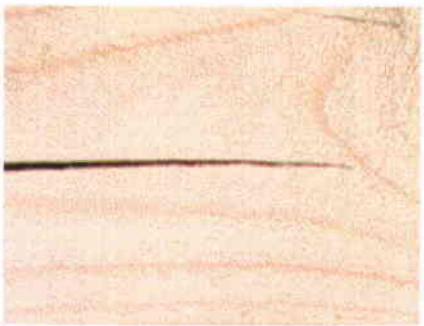

Figure 32 Splits

- **Standard Grades** of hardwood lumber are FAS, FAS 1 Face (F1F), Selects, No. 1 Common, No. 2A Common, No. 3A Common, No. 2B Common, No. 3B Common.

- **Surface Measure** (SM) is the measure of square feet on the face of a board.

  $$SM = \frac{\text{(Full width in inches and fractions of an inch)} \times \text{standard length}}{12}$$

  (Rounded to the nearest whole square foot)

  SM is used for both grading purposes and for recording the amount of wood in each board.

- **Cutting Unit** is a unit of measure that is 1 inch wide and 1 foot long. Cutting units are used to measure the amount of area that is contained in the cuttings of a board.

  (Width in Inches + Fractions) × (Length in Feet + Fractions) = Cutting Units

- **Minimum Size Board** is the minimum length and minimum width requirement of the board. There are different minimum sizes required for different grades. For example, the minimum size board for FAS grade is 6" wide by 8' long.

- **Minimum Size Cuttings** are the minimum length and width requirement for a cutting. There are different minimum size cutting requirements for different
- grades. For example, 4" × 2' and 3" × 3' are the minimum size cutting requirements for No.1 Common grade.
- **Basic Yield** is the Surface Measure multiplied by a specified number for each grade. It is the minimum number of cutting units required for the grade. For example, SM × 10 is the basic yield requirement of FAS grade.
- The **National Hardwood Lumber Association (NHLA)** is a non-profit service corporation which exists for the benefit and protection of buyers, sellers, and consumers of hardwood lumber. A primary objective of the NHLA is the development and maintenance of uniform lumber standards. It writes and promulgates lumber grading rules, and provides grade inspection and supervision services.

Appendix B Hardwood Lumber Grading Rules

Most hardwood lumber produced in this country is graded using the rules developed and maintained by the National Hardwood Lumber Association (NHLA). The NHLA grading rules are based on the amount and size of clear cuttings. The standard grades of hardwood lumber are FAS, FAS One Face (F1F abbreviation), Selects, No. 1 Common, No. 2A Common, No. 2B Common, No. 3A Common, and No. 3B Common. FAS is the highest and No. 3B Common is the lowest. In short, the higher grades of lumber have more large clear area than the lower grades. The basic concept of grading is that the grade of all lumber is determined from the poorest face or side of the lumber, with a few cases considering the better face as well.

The surface measure of a board is used to determine the number of cuttings permitted for a given grade. For example, the FAS grade specifies a minimum size of 4" x 5' or 3" x 7' for cuttings taken from a board that is at least 6" wide and 8' long. The maximum number of cuttings is nominally four to produce a clear-face yield of 83 1/3 percent. If the surface area of the board is greater than 6 square feet, an additional cutting is allowed if the yield can be raised to 91 2/3 percent.

The standard hardwood lumber grades are summarized in Table 5:

Table 5 Hardwood lumber grades standard inspection

Grade	Minimum Board Length	Minimum Board Width	Minimum Cutting Size	Min. Area of Clear Cuttings Required	No. of Cuttings	One Extra Cutting Yield
FAS	8'	6"	4" x 5' 3" x 7'	83-1/3%	SM/4 (4 max)	92-2/3%
F1F	Same as FAS for species being graded. Better Face to Grade FAS, Poor Face to Grade 1Com					
Selects	6'	4"	4" x 5' 3" x 7'	83-1/3%	SM/4 (4 max)	92-2/3%
1Com	4'	3"	4" x 2' 3" x 3'	66-2/3%	(SM+1)/3 (5 max)	75%
2Com	4'	3"	3" x 2'	50%	SM/2 (7 max)	66-2/3%
3ACom	4'	3"	3" x 2'	33-1/3%	Unlimited	
3BCom	4'	3"	1-1/2" x 2'	25%	Unlimited	

FAS

Boards 6" and wider, 8' and longer. Yields 83-1/3 percent of clear face cuttings with minimum sizes of 4" x 5', or 3" x 7'.

F1F

Same as FAS for species being graded. Better face to grade FAS, poor face to grade No. 1 Common.

Selects

Face side is FAS, back side is No. 1 Common. Boards are 4" and wider , 6' and

longer. Yields 83-1/3 percent clear face cuttings with minimum sizes of 4" x 5', or 3" x 7'.

No. 1 Common

Boards are 3" and wider, 4' and longer. Yields 66-2/3 percent clear face cuttings with minimum sizes of 4" x 2', or 3" x 3'.

No. 2A & 2B Common

Boards are 3" and wider, 4' and longer. Yields 50 percent clear face cuttings 3" and wider by 2' and longer.

No. 3A Common

Boards are 3" and wider, 4' and longer. Yields 33-1/3 percent clear face cuttings 3" and wider by 2' and longer.

No. 3B Common

Boards are 3" and wider, 4' and longer. Yields 25 percent clear face cuttings 1-1/2" and wider by 2' and longer.

Table 6 is the summary of the defect limitations for the standard grades.

Table 6 Hardwood lumber grades defect limitations

FAS limits	Pith = SM in inches. Wane = 1/2 Length. Knot = 1/3 SM. Splits: not to exceed 2 × SM or 12" whichever is greater. Splits shall not diverge 1" in 12"
Wane in F1F	FAS limitations apply to Better face. 1Com side: 1/3 W × 1/2 L. Widest Wane added together; Length can be on both edges.
Wane in Selects (Pcs. 6" & Wider)	FAS limitations apply to Better face. 1Com side: 1/3 W × 1/2 L. Widest Wane added together; Length can be on both edges.
Wane in Selects (Pcs. 4" & Wider)	1/3 W × 1/2 L applies to both faces. Add widest wane together. Add total length of wane from both edges

Note these charts summarize the main requirements or the standard grades. For complete information, refer to the NHLA Rule Book.

Source: National Hardwood Lumber Association

Appendix C Pseudo Code

Pseudo code for resolving two conflict clear areas

```
ConflictType Resolve(CClearArea& ca1, CClearArea& ca2,
vector<CClearArea*>& result)
{
   determine the conflict type by the defect coordinate;
   if ( NoConflict )
         {
          ResolveNoConflict(ca1, ca2, result);
          return NoConflict;
   }
   else  if (form2)
   {
          ResolveForm2(ca1, ca2, result);
          return Form2;
   }
   else
   {
          ResolveForm3(ca1, ca2, result);
          return Form3;
   }
}
```

Pseudo code for resolving the conflict clear area list

```
Resolve(vector<CClearArea*>& caList, vector<CClearArea*>& result)
    {
   // Don't resolve if caList is empty or result list is not empty.
   if(calise is empty || result list is not empty)    return;

   int i = 0;
   copy all the clear areas from caList to result list;
   Sort descending by value;

// resolve conflict between each pair of clear areas in the result list until all
pairs are conflict free.
   while( i < result.size() - 1)
```

```
    {
int j = i + 1;//resolve conflicts between ith and each one in the list

            bool noConflict = true;
            while(j < result.size())

            {
                    vector<CClearArea*> v;

                    if(Resolve(*(result[i]), *(result[j]), v) == NoConflict)
                    {
                            delete v;
                            j++;
                    } else {
                            noConflict = false;
                            delete result[i], result[j];
                            add v into result;
                            break; // start from (i+1)th clear area again
                          }
            } // end of inner while loop

            // no conflict between ith and all clear areas behind it
            if(j >= result.size() && noConflict)
                    i++;
    } // end of outer while loop

}
```