

A COMPUTER VISION SYSTEM FOR AUTOMATED GRADING OF ROUGH HARDWOOD LUMBER USING A KNOWLEDGE-BASED APPROACH

Tai-Hoon Cho, Richard W. Conners, and Philip A. Araman

* Department of Electrical Engineering,
Virginia Polytechnic Institute & State University,
Blacksburg, VA 24061

U.S. Forest Service, Brooks Forest Products Center,
Virginia Polytechnic Institute & State University,
Blacksburg, VA 24061

Abstract

A sawmill cuts logs into lumber and sells this lumber to secondary remanufacturers. The price a sawmill can charge for a volume of lumber depends on its grade. For a number of species the price of a given volume of material can double in going from one grade to the next higher grade. Thus, accurately establishing the grade of a volume of hardwood lumber is very important to both seller and purchaser of this material. Currently, the grading of hardwood lumber is done by human inspectors. Unfortunately, they often lack consistency in their grading of lumber. Hence, there is a strong motivation for wanting to develop an automatic lumber grading system. The research reported in this paper is aimed at developing a computer vision system that will be used to automatically grade rough hardwood lumber. The purpose of the computer vision system is to locate and identify grading defects in a species-independent manner. The current system can detect four of the most common types of grading defects: knots, holes, wane, and splits/checks. The system has been designed using a knowledge-based approach employing a Blackboard framework. The system has been tested on a number of boards from the most used hardwood species. The results indicate that the development of a fully automatic grading system is possible.

1. Introduction

Sawmills saw logs into boards. They sell these boards to secondary remanufacturers who use this raw material to manufacture goods such as furniture, cabinets, etc. The price a sawmill can charge for a volume of lumber depends on its grade, e.g., Firsts and Seconds, No. 1 common, No.2 common, etc. For a number of species the price of a given volume of material can double in going from one grade to the next higher grade. Thus, accurate and consistent grading of hardwood lumber is very important to both seller and purchaser of lumber.

Currently, the grading of rough hardwood lumber is done by inspectors according to standardized grading rules developed by the National Hardwood Lumber Association (NHLA). Skilled inspectors can accurately grade lumber. However, it takes considerable time to develop truly skilled graders and time translates into money. Further, while few studies have been conducted to determine factors that can affect grading accuracy, it can be assumed that these factors correspond to those of other human endeavors. Therefore, it can be assumed that boredom, state of mind, fatigue, etc. can all affect the accuracy of the grading process.

An automatic grading system suffers none of those human failings. It never gets tired or bored. It attacks each new board with equal enthusiasm. An automatic system would be an expert at the very first minute it is turned on.

While the grade of a board does to a large extent depend on the distribution of defects on the board's surface, appropriate edging and trimming can increase the grade of a board. Hence optimal edging and trimming provide a mechanism for a sawmill to increase his profits. One preliminary study indicates that optimal edging and trimming could increase profits by at least 20% [1]. Optimal edging and trimming would be possible if an automatic grading system existed. Thus, there are a number of reasons to believe that developing an automatic grading system could have a very positive impact on the sawmilling industry.

Any automatic grading system must have two components. The

first of these is a computer vision system for locating and identifying grading defects on rough hardwood lumber. A grading defect is a surface defect that can affect a board's grade under NHLA rules. The surface defects that occur most frequently include knots, wane, holes, decay, stain (mineral and blue), checks, and splits. The second component is a program that can establish the grade of a board based on the output of the vision system. It is this program that incorporates the NHLA grading rules and grades a board based on the distribution of defects present on the board's surface. Significant progress has already been made on creating the second of these components [2]. This paper is aimed at developing the first component, the vision system for locating and identifying grading defects.

While some investigators [3-7] have reported progress on developing computer vision systems for locating and identifying defects on surfaced lumber, little if any work [8] has been done on locating and identifying defects on rough lumber.

These systems suffer serious problems inherent in statistical pattern classification [9]. The most serious one is that they do not utilize the available knowledge about the various properties of defects such as shape, size, location, syntactic and semantic relationships with other defects, etc. As a consequence, classification is image dependent and unstable. This leads to a conclusion that statistical classification methods cannot be used for inspecting rough lumber. Rather, a knowledge-based approach is appropriate to incorporate all kinds of available knowledge about various properties of the grading defects.

2. overview of The Vision System

The vision system consists of two modules: a low-level module and a high-level module. The block diagram of the system is shown in Figure 1. An overview for each module is given below.

2.1 The Low-level Module : Segmentation and Feature Extraction

The low-level module consists of two parts: segmentation of an input board image and extraction of region properties. The purpose of the segmentation is to reduce the volume of the image data to be processed. Specifically, it tries to extract potential defects from the image. Since 80% or more of the total area of a board is clearwood, it is critical in saving processing time to detect potential defects as fast as possible and at an early processing stage. Once this has been done, higher level processing that is typically more sophisticated and time-consuming can be applied only to those regions believed to potentially contain a defect. Such regions usually occupy a very small percent of the whole image. Segmenting an input board image is performed using a histogram based thresholding method.

After segmentation, the low-level module identifies all connected regions, eliminates small noise regions, merges adjacent regions that have similar average gray levels, and computes region properties of merged regions. The result of this processing is a more accurate and concise description of each of the regions passed on for higher level processing.

2.2 The High-Level Module : Recognition

The function of the high-level module is to perform the scene analysis task and to identify the defect present in each of the regions passed to it. In performing this function the high-level

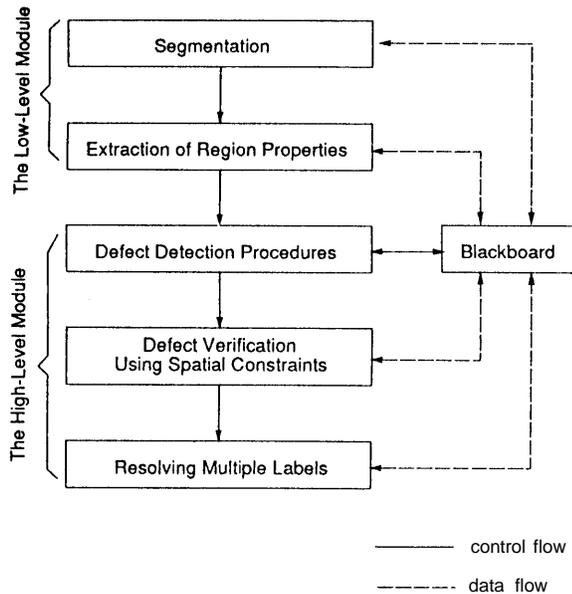


Fig. 1 Block diagram of the vision system

module uses both bottom-up processing on the regions that have been found as well as top-down processing using original image data when a hypothesis needs further substantiation. A Blackboard framework [10] is used as the software architecture for the high-level vision module because it allows the use of diverse, independent, potentially inaccurate sources of knowledge in a cooperative manner. By using focus of attention mechanism, the best data and the most promising methods can be exploited. A fuzzy confidence measure is used to represent uncertainty.

2.2.1 A Blackboard Framework for the Vision System

A typical blackboard consists of three basic components : knowledge sources (KSs), blackboard data structure, and control modules.

Knowledge Sources

The domain knowledge needed to solve a problem is partitioned into KSs, which are kept separate and independent. The condition component dictates when the KS can be put to use and the action component specifies the contribution that the KS can make. A KS can communicate with other KSs through only a global data base, a blackboard.

There are a number of knowledge sources for the high-level vision module. A KS, INIT, does some initialization and computes confidence vectors. Each component of this vector for a region is the confidence that the region is of a particular defect type. LBLCON is a KS for labeling unlabeled regions using spatial contextual dependency. VRFDFC is a KS for verifying labels assigned to regions by defect detection KSs. This verification involves using spatial constraints between different defects. RMLBL is a KS that resolves labeling ambiguity if multiple defect labels are assigned to a region. Other KSs are defect detection procedures, each of which is designed to detect one kind of defect in the surface of rough lumber. It is natural to partition knowledge about defects, according to defect types. Its major advantage is easiness in adding a detection procedure for a new type of defect.

Blackboard Data Structure

The purpose of the blackboard is to hold computational and solution-state data needed by and produced by the KSs. The KSs use the blackboard data to interact with each other indirectly. A blackboard consists of objects that can be input data, partial solutions, alternatives, and final solutions (and possibly, control data). The objects in the blackboard are hierarchically organized into levels of analysis.

The blackboard structure for the high-level module is organized

into the following entities:

- Original image
- Symbolic segmented image
- Region property table
- D_OBJECT property table
- Region adjacency matrix
- D_OBJECT adjacency matrix

The symbolic segmented image is a symbolic image where each pixel in a region is labeled with the same unique region number. The symbolic image is generated by segmentation step in the low-level module. The region property table is a collection of property lists for all regions. A property list is associated with each region. A property list consists of a region attribute vector, a confidence vector, a pointer vector, and an interpretation label. The region attribute vector consists of various region properties whose values are assigned by the low-level module. The interpretation label is the label (clearwood or one of the defect types) assigned to the region. Confidence vectors, pointer vectors, and interpretation labels are assigned values by the high-level module.

There are two basic units in the blackboard: regions and D_OBJECTs which are hierarchically linked by a part-whole relationship. A D_OBJECT consists of a set of connected regions that have the same defect label. Each component of a region's pointer vector points to a D_OBJECT associated with the region. Each D_OBJECT generated stores its properties on the D_OBJECT property table. The D_OBJECT property table has the same structure as the region property table except the pointer vector field. Each component of the pointer vector associated with a D_OBJECT is assigned a specific defect label if the D_OBJECT is verified by the corresponding defect detection procedure. The region adjacency matrix is defined as a matrix whose (i,j)-th element is the common perimeter between region i and region j divided by the perimeter of region i. The D_OBJECT adjacency matrix is defined similarly. These adjacency matrices provide contextual information among regions or D_OBJECTs that is vitally important in interpreting images.

Control

There is a set of control modules that monitor the changes on the blackboard and decide what actions to take next. Various kinds of information are made globally available to the control modules to determine the focus of attention. The focus of attention indicates the next thing to be processed. The focus of attention can be either the KSs or blackboard objects, or a combination of both.

In the vision system, the control of processing flow is done using a combination of both sequential control and opportunistic control. This combination is more efficient than opportunistic control of a typical blackboard system. At an initial stage of high-level processing, KS INIT is applied for initialization that includes calculation of confidence vectors. Then, KSs for defect detection are run independently and opportunistically. Confidence vectors are used for the focus of attention in selecting which KS (for defect detection) to be activated and next regions to process. Once KSs for defect detection have finished their operations, KS VRFDFC (verification of defects using spatial constraints) and KS RMLBL (resolving labeling ambiguity) are performed sequentially. Before and after applying VRFDFC, a determination is made as to whether there are any regions that are still unlabeled. If there are unlabeled regions, then KS LBLCON is invoked to label those regions using spatial contextual dependency.

3. The Low-Level Module

3.1 Segmentation

This step performs two functions. First it separates pixels of background from pixels of board. Then it separates pixels of clear wood from pixels that might be from area containing a grading defect. The module uses as input a 8-bit 480x512 full color image of a board.

Separating pixels of background from pixels of board is facilitated by the fact that one can control the color of the background. Details of this step are given in [4].

Differentiating pixels of clear wood from pixels that might be of a defect is a more difficult problem. A histogram based thresholding is used. The thresholds are adaptively and automatically selected based on valley points and inflexion points of the gray level histogram of the input image. The motivation for the histogram based thresholding comes from the fact that it is very insensitive to surface roughness, and from the requirement that the vision system must be able to process high resolution images as fast as possible. The detailed steps used to find the thresholds are given in [8].

Once the thresholds have been found, regions formed by each pair of thresholds are marked. All pixels having gray level value between two consecutive thresholds are marked the same. Connected component labeling [11] is performed. All connected regions are found and each such region is given a unique label to differentiate it from the rest of the connected regions.

3.2 Extraction of Region Properties

The second step of the low-level module is to extract a variety of properties of segmented regions that are used for labeling regions as clearwood or a particular type of defect in the high-level, recognition module. Unfortunately, segmented output typically contains many small meaningless regions due to both rough surface structure of a board and digitization noise. Also, some regions are overfragmented. Therefore, to provide more reliable region descriptions to the high-level module, it is necessary to eliminate small noisy regions and to merge overfragmented regions. Reducing the number of regions also decreases computation time needed to extract properties for all regions. The small region elimination operation is performed first because it is computationally fast and it usually significantly reduces the number of regions produced by the original segmentation. A region merging operation is then performed. This region merging operation is computationally more complex than the small region elimination operation because the former is based on a statistical test. It is applied to only those regions that remain after the small region elimination operation.

The small region elimination is done similarly as that discussed in [12]. After all the small regions have been eliminated, further region merging is performed. This merging operation is based on a statistical T-test for equality of average gray levels between adjacent regions R_1 and R_2 . The underlying assumptions associated with the use of the T-test is that the gray levels of the pixels in both R_1 and R_2 are independent, that they have identically distributed normal distributions, and that the variances of the distributions are unknown but equal. The steps associated with using the T-test are given below. Similar statistical tests have been used by other investigators, for example, [13].

Step 1. Select R_1 , the region with the smallest area. Let n_1 be the number of pixels in R_1 .

Step 2. Find R_2 . Of all the regions adjacent to R_1 , R_2 is the region that is the most similar in average gray level. Let n_2 be the number of pixels in R_2 .

Step 3. Compute the sample mean, X_1 , and the sample variance, S_1^2 , of the gray levels of R_1 . Also compute the sample mean, X_2 , and the sample variance, S_2^2 , of the gray levels of R_2 .

Step 4. Using

$$S_p^2 = \frac{(n_1 - 1)S_1^2 + (n_2 - 1)S_2^2}{n_1 + n_2 - 2}$$

compute the statistic

$$t_{obs} = \frac{\bar{X}_1 - \bar{X}_2}{S_p \sqrt{\frac{1}{n_1} + \frac{1}{n_2}}}$$

Step 5. If the absolute value of t_{obs} is less than a threshold, then assign R_1 the region label of R_2 . (The threshold used should be a very conservative one so as to prevent over-merging.)

After the region merging operation is complete, a vector of features is extracted from each of the resulting regions. These features are as follows:

- 1) area, the number of pixels in a region;
- 2) average gray level, average of all the gray levels of the pixels in a region;

- 3) center of mass,

$$(\bar{r}, \bar{c}) = \left(\frac{1}{N} \sum_{(r,c) \in R} r, \frac{1}{N} \sum_{(r,c) \in R} c \right)$$

where $(r,c) \in R$ represents the coordinates of a pixel in region R , and N is the area of region R ;

- 4) minimum bounding rectangle (MBR), whose upper left vertex point, (r_1, c_1) and lower right vertex point, (r_2, c_2) are defined by

$$(r_1, c_1) = \left(\min_{(r,c) \in R} r, \min_{(r,c) \in R} c \right)$$

and

$$(r_2, c_2) = \left(\max_{(r,c) \in R} r, \max_{(r,c) \in R} c \right)$$

- 5) elongatedness,

$$ELONG = e_1/e_2$$

where e_1 and e_2 are eigenvalues of the covariance matrix of the distribution of $(r - \bar{r}, c - \bar{c})$ and $e_1 > e_2$;

- 6) perimeter, the length of the boundary of a region;

- 7) compactness,

$$C = \frac{(perimeter)^2}{4\pi(area)}$$

- 8) flag_board_boundary, a flag indicating whether a region is touching board boundary or not.

The center of mass of a region gives a rough estimate for the location of the region. There are four kinds of basic shape features: MBR, Elongatedness, Perimeter, and Compactness. The MBR of a region is very useful to efficiently locate the region in the original image or the segmented symbolic image. It also provides a rough estimate for elongatedness of a region. Elongatedness of a region is necessary to differentiate splits/checks from other defects. Compactness of a region is useful for finding compact defects, e.g., holes. Perimeter of a region is used to calculate compactness of the region. Flag_board_boundary of a region is used to determine whether the region is touching board edge or not. This is a helpful feature in finding wane since wane almost always appears on the edge of a board.

4. The High-Level Module

The goal of the high-level module is to identify the type of defect present in each of regions passed from the low-level module and to extract the appropriate characteristics associated with each defect. The basic strategy used is to first detect each defect independently of the others, and then resolve ambiguity for regions [that are not labeled uniquely]. A Blackboard framework is adopted as the software architecture in this module.

The first KS to be used is INIT (initialize), a KS for initialization. All pointer vectors and interpretation labels are initialized as null, that is, all regions are unlabeled. The largest region is labeled as clearwood since 80 percent or more of the total area of a board is always clearwood. Next a region adjacency matrix (RAM) is constructed and is stored in the blackboard for later use. Finally, for each region a confidence vector is determined based on this region's properties, i.e., the properties that were computed by the low-level module. Each component of this vector is the confidence that the region is of a particular defect type. The evaluation of these confidences is based on fuzzy set theory [14]. For details of this step, see [8].

Once all the regions have had their initial confidence vectors computed, defect detection procedures are applied. These defect detection procedures are very specialized KSs. Each defect detection procedure is designed to detect a particular defect using knowledge about spectral and shape characteristics of that defect. It should be noted that each procedure runs independently of the others; the order of applying the procedures is unimportant. A region's confidence vector determines which defect detection procedures will be applied to that region.

Basically, each defect detection procedure consists of two steps. The first step is to detect regions that seem to be the defect of interest, and give them this defect label. As the labeling is being performed links between adjacent regions with the same label are being formed. Such linked regions form the next level in the descriptive hierarchy, D_OBJECT. The second step in each defect detection procedure is labeling verification. The

verification step involves generating a property list for each D_OBJECT. This new property list is used together with defect specific knowledge to verify the labeling. Note that after the application of the defect detection procedures it is possible that a region can be in multiple D_OBJECTS.

Even after all the defect detection procedures have been applied, there may still exist unlabeled regions. Such regions are labeled using spatial contextual dependency by the KS LBLCON (label by context). First, every unlabeled region is labeled as clearwood. This step is motivated by the fact that the defect detection procedures should be able to identify any region that truly is a defect. Based on this assumption any unlabeled region should be clearwood. However, in reality some unlabeled regions are really a part of a defect. To allow for this possibility after LBLCON has finished step 1, it checks to see if each "small" clear wood region is almost enclosed by any D_OBJECT. If this is the case, then this region is given the label of the D_OBJECT with which it shares the longest common boundary. The word "small" is used above to indicate that the largest region is not subjected to this analysis. This prevents a defect around the boundary of the board from causing the interior clearwood area to be labeled as a defect. As clearwood regions are assigned new defect labels by LBLCON, applicable D_OBJECTS are updated to reflect the new labeling. Any updated D_OBJECT has a flag set to indicate it has been updated. At this time no updated property list is generated for any of the D_OBJECTS.

The KS VRFDFC (verify defects) attempts to verify the labeling of each D_OBJECT by using spatial constraints among adjacent D_OBJECTS. For example, if a D_OBJECT that is not labeled as wane is surrounded by a D_OBJECT that is labeled as wane, then the surrounded D_OBJECT is changed to wane. This follows the fact that there are no other defects in an area of wane. If the validation procedure fails to verify the label assigned to a D_OBJECT, the D_OBJECT is removed from further considerations. The pointer vector of each region associated with this D_OBJECT has its component corresponding to the D_OBJECT's label re-initialized as null. If after KS VRFDFC has been applied there are any unlabeled regions, KS LBLCON is used to label these regions using spatial contextual dependency.

The final step in the processing is to resolve instances where regions have been assigned multiple labels, i.e., where a region is a part of more than one D_OBJECT. This processing is performed by KS RMLBL (resolve multiple labels). To find regions that are part of more than one D_OBJECT and to identify the D_OBJECTS involved the region pointer vectors are used. Each D_OBJECT corresponds to one possible label for the region. To determine which label the region should have, a new property list is computed for each D_OBJECTS involved if required; it depends on whether a flag has been set by KS LBLCON. Each D_OBJECT's property list is used to calculate a confidence value that the D_OBJECT has its defect label. The region is assigned the label of a D_OBJECT whose confidence value is the highest among the confidence values of the D_OBJECTS associated with the region.

4.1 The Defect Detection Procedures

There is a defect detection procedure for each type of defect. The first step in each defect detection procedure is to label a region as the corresponding defect if the region has a high confidence of being of its defect type. The basic strategy used in this step is to first detect regions that represent "exemplary" examples of the defect the procedure was designed to recognize. Specialized defect detection methods are applied to detect such "exemplary" defects. After all of the exemplary defects have been recognized, then regions representing more "ambiguous" examples of that defect type are considered. Labeling these regions is based on defect detection methods that gauge contextual dependency and similarity of region properties.

As the labeling is being performed links between adjacent regions with the same label are being formed. Such linked regions form the next level in the descriptive hierarchy, D_OBJECT. A D_OBJECT is formed for each group of regions formed.

The next step of a defect detection procedure is a labeling

verification step. It is necessitated by the fact that each region does not usually correspond to a defect; typically a defect is fragmented into several regions. At this stage of the processing a property list is computed for each D_OBJECT. The properties at the level D_OBJECT should be more representative of the actual defect that is present. These new properties, together with other knowledge about the defect, are used to verify the labeling of the D_OBJECT. Some specialized image processing techniques might be used to extract further evidence for the defect which is to be recognized.

4.1.1 Split/Check Detection Procedure

This K is activated only if there is at least one region in the image whose confidence vector indicates it might be a split/check. Usually, splits/checks appear in an image as a linear structure that is elongated, thin, and dark. To detect regions representing "exemplary" splits/checks, a "crack" detector is applied to each unlabeled region, R, whose confidence vector indicates it might be a split/check. If the crack detector has successfully found a crack within R, then R is labeled as a split/check.

The crack detection in R is performed by 1) applying four directional masks shown in Figure 2 to each pixel of R, 2) counting the number of pixels where maximum response of the four masks is greater than a threshold, and 3) determining if $N/Area(R) > T$, where Area(R) represents the area of region R, N is the count obtained by step 2, and T is a threshold. If R satisfies step 3, then R is considered to have a "crack" in it. Otherwise, R is considered to have no crack in it.

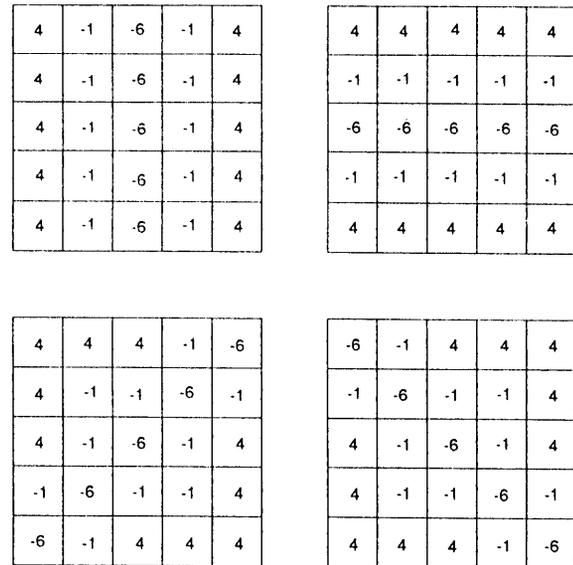


Fig. 2 Four directional masks for crack detection

Next, regions representing more "ambiguous" examples of splits/checks are detected. Let R be an unlabeled region. If split/check component of the R's confidence vector is low and if there is an adjacent region, R', that has been already assigned a split/check label and if the elongatedness of R merged with R' is greater than that of either R or R', then R is labeled as a split/check. This rule detects regions that have low confidence of being a split/check but are aligned with a region already recognized as being a split/check.

After all the initial labeling has been performed, D_OBJECTS labeled as splits/checks are generated and their properties are computed. Two D_OBJECTS are connected into a single D_OBJECT, if they are colinear and have a small gap between them. The reason for this merging is that a single split/check might be broken into several disconnected pieces, due to the imperfect segmentation.

Finally, a verification method is applied to each of the D_OBJECTS. For a D_OBJECT to be considered a split/check,

it should not be small, it should not touch background or image frame boundary significantly, it should find a “crack” inside it, and it should be long enough.

4.1.2 Hole Detection Procedure

This KS is activated only if there is at least one region in the image whose confidence vector indicates it might be a hole. There are two types of holes : small and large holes. Usually, birds and worms make small holes which appear circular and very dark. Large holes are caused by the dislocation of knots. They appear very dark with shape of either circle or ellipse.

To detect regions representing “exemplary” holes, all regions that have a positive “hole” component in their vector are considered. Let R be such an unlabeled region. If the area of R is small, elongatedness of R low, and R much darker than its adjacent regions, then R is assigned the label hole. (This rule is based on the way small holes typically manifest themselves in images) If the area of R is large and R is “very dark,” then R is also assigned the label hole. (This rule is based on the way large holes typically manifest themselves.)

Next, regions representing more “ambiguous” examples of holes are labeled using the following rule. Let R be an unlabeled region. If an adjacent region R’ has been labeled as a hole and if R and R’ have similar average gray values, then label R as a hole.

Once initial labeling has been performed, D_OBJECTs labeled as holes are generated and their properties are computed. Finally, a verification procedure is applied to each resulting D_OBJECT. For the verification procedure to consider a D_OBJECT to be a hole, it should satisfy all the following conditions:

- 1) It should have a large confidence for holes.
- 2) Its area should not be very small.
- 3) If its area is large, then it should not be highly elongated.
- 4) If its area is small, then it should satisfy all the following:
 - i) It should have low elongatedness
 - ii) It should be compact
 - iii) Both N1 and N2 should be 1 and minimum gray level of the region should be very low, where N1 (N2) is the number of valleys in gray levels of pixels lying on the horizontal (vertical) line passing through the centroid of the region.

4.1.3 Wane Detection Procedure

This KS is activated only if there is at least one region in the image whose confidence vector indicates it might be wane. Wane is defined as bark or lack of wood. It typically consists of two parts: an external part and an internal part. (Some boards have only external wane.) External wane appear very dark and is on the edge of a board while internal wane is highly elongated, looks less dark, and exists between external wane and clearwood.

External wane is first detected and then internal wane is detected, if it exists. To detect external wane, 1) label regions of high confidence as wane, and then 2) label a region as wane if it is surrounded by regions already labeled as wane.

Internal wane is detected using the following steps. 1) Label a region as wane if it is dark, highly elongated, and shares significant portion of common boundary with regions already recognized as wane. 2) Label a region as wane if it is surrounded by regions already labeled as wane.

Once initial labeling has been performed, D_OBJECTs labeled as wane are generated and their properties are computed. Finally, verification is performed for each resulting D_OBJECT. For a D_OBJECT labeled as wane to be verified, 1) it should touch background considerably, 2) it should not be extremely elongated, and 3) it should be much darker than its adjacent regions.

4.1.4 Knot Detection Procedure

This KS is activated only if there is at least one region in the image whose confidence vector indicates it might be a knot. It is very difficult to characterize knots because they appear quite different from one knot to another in gray level, shape, and/or size. We can have both a very small knot and very large knot. Knots can be circular or elliptical. Knots are usually dark, but average gray value difference between knots can be large.

In order to detect regions, representing “exemplary” knots, all regions that have high confidence level are labeled as knots. For any remaining unlabeled region R, apply the following rules. If R has some degree of confidence as a knot and if R shares a significant portion of its region boundary with regions that have already been labeled as knots, then label R as a knot. This rule combines together many regions formed by the segmentation but that go together to form the whole knot structure.

Once initial labeling has been performed, D_OBJECTs labeled as knots are generated and their properties are computed. Finally, verification is performed for each resulting D_OBJECT. For a D_OBJECT labeled as a knot to be verified, 1) if it is small, it should not touch the background, and 2) it should be much darker than its adjacent regions.

4.2 Defect Verification Using Spatial constraints

Each defect detection procedure runs independently of others without consideration of other defects. Thus, it is necessary to remove any incompatibility between adjacent defects using spatial constraints between them. A KS, VRFDFC is activated for this purpose, once all defect detection procedures have been run and finished. For each remaining D_OBJECT, O, to be verified, none of the following conditions should hold:

- 1) LABEL(O)=split/check and ENCLOSED_BY(O,X), where LABEL(X)=wane
- 2) LABEL(O)=split/check and ADJACENT(O,X), where LABEL(X)=hole
- 3) LABEL(O)=knot and ADJACENT(O,X), where LABEL(X)=hole
- 4) LABEL(O)=knot and ADJACENT(O,X), where LABEL(X)=wane

In the above, LABEL(O) denotes defect label of D_OBJECT O, and both ENCLOSED_BY and ADJACENT are relational predicates. ENCLOSED_BY(O,X) is true if O is enclosed by X, and ADJACENT(O,X) is true if O is adjacent to X.

If the validation procedure fails to verify the label assigned to a D_OBJECT, the D_OBJECT is removed from further considerations. The pointer vector of each region associated with this D_OBJECT has its component corresponding to the D_OBJECT’s label re-initialized as null. If after KS VRFDFC has been applied there are any unlabeled regions, KS LBLCON is used to label these regions using spatial contextual dependency.

5. Experimental Results

The species considered were cherry, red oak, yellow poplar, and maple. These species were selected because approximately 50 percent of all household wood furniture is made from these species. Hence having a system that should only grade these species would still be a viable commercial product. Approximately 30 boards were selected for each species from the data base of rough lumber in the Spatial Data Analysis Laboratory. The samples were chosen to reflect the variation in appearance of each of the four defects, knots, holes, wane, and splits/checks

An 8 inch by 8 inch area of each sample was selected to be the area scanned. This area was digitized using a 512x480x8 bits resolution black and white camera. The spatial resolution is approximately 64 points per inch. To obtain the desired color images color filters were used. Tungsten halogen bulbs were used to approximate afternoon sun viewing conditions. To normalize for, the difference in spectral sensitivity of the silicon/tungsten halogen combination neutral density filter were used. The same lighting conditions, camera settings, and viewing angle were employed in creating the color image of each sample. Each color image was shading corrected to remove any non-uniformities in lighting or sensitivity across the camera’s imaging array [15].

Several processing examples are presented to demonstrate capabilities of the current system. First, a detailed illustration of each step is given using a sample image. A black and white image of a rough cherry board is shown in Figure 3(a). Figure 3(b) shows region boundaries right after segmentation (before small region elimination). Figure 3(c) shows region boundaries after small region elimination and region merging. We can notice that a lot of small noisy regions have been removed.

Recognized defects after high-level processing, splits/checks and a knot are shown in Figure 3(d). (Each defect label is given a unique gray level with its name. White area represents clearwood.)

Figure 4 shows the results obtained for a rough red oak sample. Observe that wane and a knot have been correctly recognized. Figure 5 shows the results obtained from processing a rough yellow poplar sample. A large knot has been correctly identified. Also note that this sample contains both heartwood (dark area) and sapwood (light area). Both comprise the clearwood of the board. Neither is a grading defect. Observe that the vision system ignored the difference in appearance.

6. Conclusions

A computer vision system for automated grading of rough lumber has been described. The purpose of the computer vision system is to locate and identify surface defects that affect the grade of boards. The current system can detect four of the most common types of defects : knots, holes, wane, and splits/checks. The system has been designed using a knowledge-based approach employing a Blackboard framework. The system has been tested on a number of boards from four hardwood species. While it has limited recognition capabilities, only being able to identify four defect types, it does suggest that species independent methods can be found for accomplishing the required tasks. However, much further research is required. The future research should be directed toward expanding the list of defects that can be identified by the system as well as improving the accuracy of defect recognition. To do so, more knowledge about the defects and more sophisticated image processing methods will probably have to be incorporated in the system.

REFERENCES

- [1] E. Kline, *Personal Communication*, 1990.
- [2] P. Klinkhachorn, J.P. Franklin, C.W. McMillin, R.W. Conners, and H.A. Huber, "Automated computer grading of hardwood lumber," *Forest Products Journal*, vol.38, no.3, pp.67-69, 1988
- [3] R.W. Conners, C.W. McMillin, K. Lin, and R.E. Vasquez-Espinosa, "Identifying and locating surface defect in wood: part of an automated lumber processing system," *IEEE Trans. Pattern Analy. and Mach. Intelli.*, vol. PAMI-5, no.6, pp.573-583, 1983.
- [4] R.W. Conners, C.T. Ng, T.H. Cho, and C.W. McMillin, "Computer vision system for locating and identifying defects in hardwood lumber," *SPIE Vol. 1095, Applications of Artificial Intelligence VII*, pp.48-63, 1989.
- [5] A.J. Koivo and C.W. Kim, "Parameter estimation of CAR models for classifying wood boards," *Proc. IEEE Int. Conf. Syst. Man Cybern.*, Aug. 1988.
- [6] A.J. Koivo and C.W. Kim, "Robust image modeling for classification of surface defects on wood boards," *IEEE Trans. Systems, Man, and Cybernetics*, vol.19, no.6, pp.1659-1666, 1989.
- [7] D. Paul, W. Hattich, W. Nill, S. Tatari, and G. Winkler, "VISTA : Visual Interpretation System for Technical Application - Architecture and Use," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 10, pp.399-407, 1988.
- [8] T.H. Cho, R.W. Conners, and P.A. Araman, "A computer vision system for analyzing images of rough lumber," *Proc. 10th International Conference on Pattern Recognition*, pp.726-728, Jun. 1990.
- [9] M. Nagao and T. Matsuyama, *A Structural Analysis of Complex Aerial Photographs*, Plenum, 1980,
- [10] R. Engelmores and T. Morgan (Eds.), *Blackboard Systems*, Massachusetts : Addison-Wesley Publishing Company, 1988.
- [11] Y. Shirai, *Three-Dimensional Computer Vision*, Springer-Verlag, 1987.
- [12] T. Pavlidis, *Structural Pattern Recognition*, New York : Springer-Verlag, 1977.
- [13] Y. Yakimovsky, "Boundary and object detection in real world images," *J. Assoc. Comput. Mach.* 23, pp.599-618, 1976.

- [14] L.A. Zadeh, "Fuzzy sets," *Information Control*, vol.8, pp.338-353, 1965.
- [15] A.A. Sawchuk, "Real-time correction of intensity nonlinearities in imaging system," *IEEE Trans. Comp.*, vol.C-26, No.1, pp.34-39, 1977.

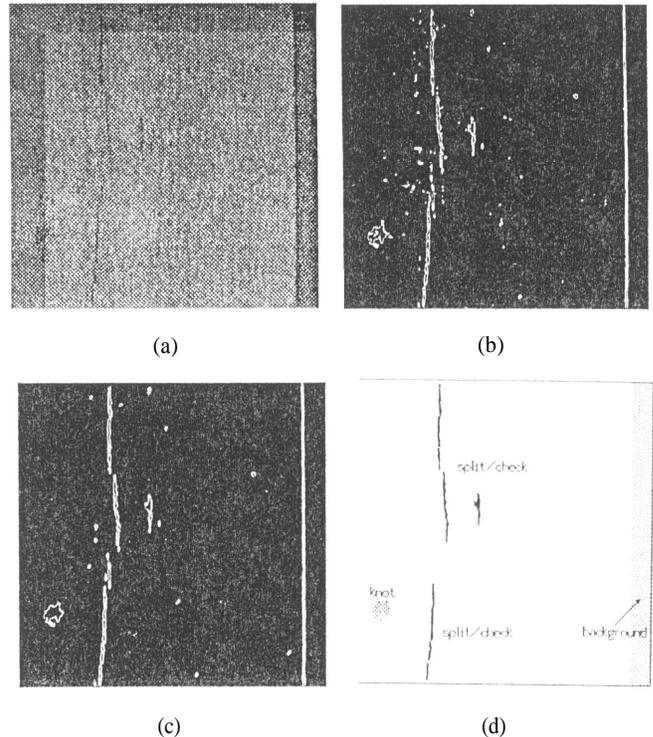


Fig.3. A rough cherry board. (a) Original image. (b) Region boundaries right after segmentation. (c) Region boundaries after region merging. (d) Results of defect recognition.

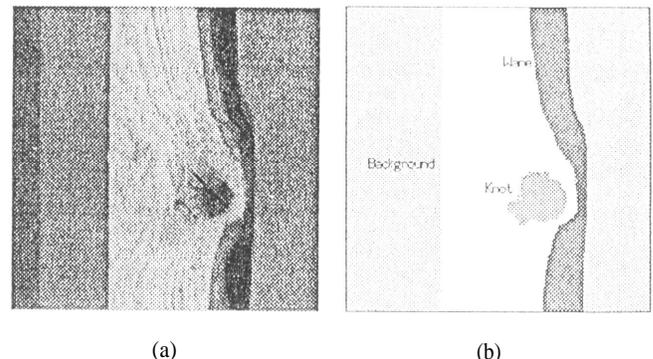


Fig.4. A rough oak board. (a) Original image. (b) Results of defect recognition.

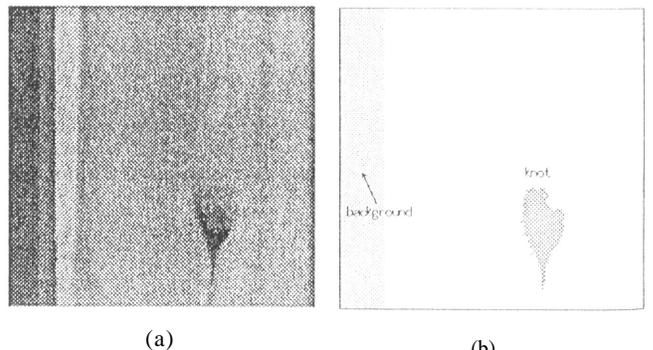
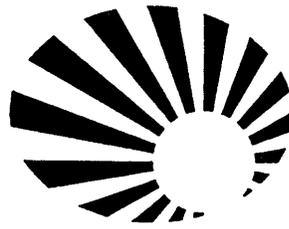


Fig.5. A rough poplar board. (a) Original image. (b) Results of defect recognition.

1990 IEEE International Conference on Systems, Man, and Cybernetics



Conference Proceedings

November 4-7, 1990

Sheraton Universal Hotel
Los Angeles, California

90CH2930-6



IEEE Systems, Man
and Cybernetics Society
