Interactive Machine Learning
for Postprocessing CT Images of Hardwood Logs

Erol Sarigul, A. Lynn Abbott
Bradley Department of Electrical and Computer Engineering
340 Whittemore Hall, Virginia Tech, Blacksburg, VA 24061 USA
e-mail: {esarigul, abbott}@vt.edu

Daniel L. Schmoldt
USDA Cooperative State Research, Education, and Extension Service
800 9th Street SW, Waterfront Centre, Room 3422, Washington DC 20024 USA
e-mail: dschmoldt@reeusda.gov

Abstract

This paper concerns the nondestructive evaluation of hardwood logs through the analysis of computed tomography (CT) images. Several studies have shown that the commercial value of resulting boards can be increased substantially if log sawing strategies are chosen using prior knowledge of internal log defects. Although CT imaging offers a potential means of obtaining this knowledge, the automated analysis of the resulting images is difficult, particularly for hardwood species, because of the natural texture/density variations within the wood material. In spite of the difficulties, a few researchers have developed image-analysis systems that demonstrate good statistical accuracy in locating and identifying defects to create "classified" images. Even with good quantitative results, however, classified images can often be improved qualitatively through postprocessing steps that refine the shapes of the detected image regions. To be most effective, postprocessing operations should utilize domain knowledge that is specific to the type and position of different defects. This paper describes an interactive approach for acquiring this domain knowledge. A system has been developed that generates postprocessing rules by observing the operations that are performed as a human user interactively edits a classified CT image. Based on these observations, the system infers rules that can be used subsequently for automatic postprocessing of CT images. The system is incremental, in that the system is capable of updating its rules at a later time. Laboratory tests have shown good improvements to classified images from two red oak logs.

1. Introduction

Decreasing wood resources and increasing demand for lumber have driven the hardwood industry to find improved methods for converting logs to lumber. Conventional practices for log processing, particularly during sawing operations, result in a considerable amount of waste. Several studies have shown that the commercial value of resulting boards can be increased substantially if sawing strategies are chosen using knowledge of internal defects ([4],[6],[17],[23]).

One method of obtaining information concerning the internal structure of logs is through computed tomography (CT) imaging. This nondestructive technique provides image "slices" that represent physical, cross-sectional density distributions of a scanned object. Because of the large amount of data that is typically collected during CT scanning, it is desirable to have automated techniques that can quickly analyze the images to locate and identify defects, and to propose breakdown strategies.

Researchers at Virginia Tech and at the Southern Research Station of the USDA Forest Service have developed an approach that uses artificial neural networks (ANNs) to classify CT pixels individually, using small neighborhoods of CT density values as input feature vectors ([12],[18],[19]). The ANN assigns a label ("knot," "decay," "split", "bark," or "clear wood") to each foreground pixel in the image. In several studies, the results have been quite good statistically, with classification accuracies often exceeding 92% ([12],[18]).

In spite of the quantitative success of the approach, the resulting image classification from the ANN can often be refined through postprocessing. In some cases, for example, the ANN produces small spurious regions that do not seriously affect the statistical accuracy, but are qualitatively undesirable and may adversely affect subsequent analysis of the log. As another example, the ANN occasionally assigns incorrect labels to the outside border of the wood, because these portions of the log appear as low-density points due to spatial quantization effects. A postprocessing step can make use of global, domain-specific information to refine regions, thereby offsetting the ANN's heavy reliance on local pixel neighborhoods.

There are several difficulties in designing a postprocessing module, however. Although many postprocessing steps can be easily implemented, different situations (possibly depending on the species of wood, on particular defect types, on the intended use of a log, and on personal preferences of the user) may require different types and degrees of region refinement. For these reasons, we are developing a system that can *learn* postprocessing rules automatically. The system observes the steps taken as a human user interactively edits a processed image, and then infers rules from those actions.

During the system's "learn mode," the user views labeled images and makes refinements through the use of a keyboard and mouse. As the user manipulates the images, the system stores information related to those manual operations, and develops internal rules that can be used later for automatic postprocessing of other images. After one or more training sessions, the user places the system into its "run mode." The system then accepts new images, and uses its rule set to apply postprocessing operations automatically in a manner that is modeled after those learned from the human user. At any time, the user can return to learn mode to introduce new training information, and this will be used by the system to updates its internal rule set.

The system does not simply memorize a particular sequence of postprocessing steps during a training session, but instead generalizes from the image data and from the actions of the human user so that new CT images can be refined appropriately. Because it learns from a human "teacher," this approach represents a form of supervised machine learning. However, the level of supervision is relatively mild by traditional machine-learning standards, because the teacher does not need to be knowledgeable concerning internal feature spaces or representations for rule selection. Because of its ability to accept new training inputs over time, the system is said to perform "incremental" (or "dynamic" or "on-line") learning. This contrasts with many machine-learning systems, which require all training data to be made available at the beginning. Such systems perform "batch" (or "static") learning.

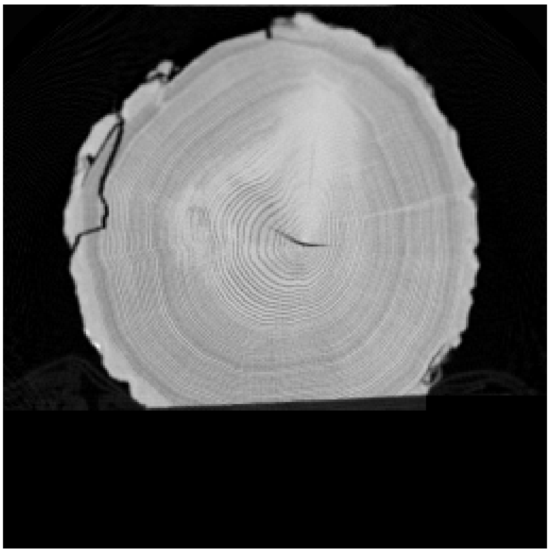
The next section of this paper provides an overview of the approach. Section 3 provides more details concerning the inferencing procedure that we have chosen for the prototype system. Section 4 presents some results that have been obtained using the system, and Section 5 contains concluding remarks.

2. System Overview

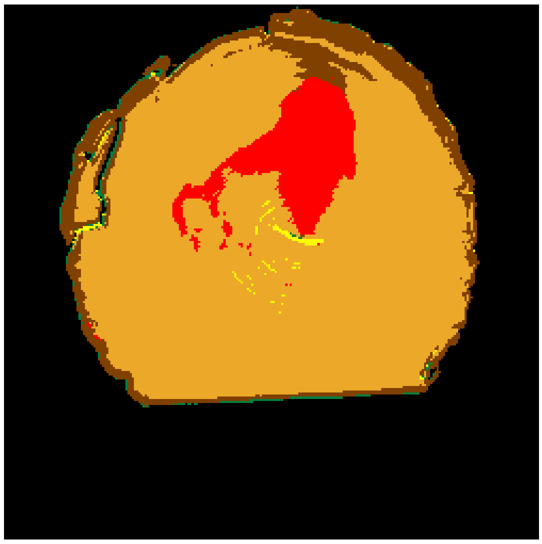
The overall classification system consists of three modules: (1) a preprocessing module, (2) an artificial neural network (ANN) module that performs tentative image segmentation/classification, and (3) a postprocessing module, which is the primary topic of this paper. The preprocessing module distinguishes wood from background (air) and internal voids, and normalizes CT density values. The ANN module labels each non-background pixel of a CT slice using histogram-normalized values from small windows of size 3×3×3 or 5×5, centered on each pixel location to be classified. In the postprocessing module, morphological operations are applied to remove spurious regions and to refine region shapes.

Figure 1 illustrates the nature of the problem with an example image slice of a red oak (*Quercus rubra*, L.) log. In the figure, output from an ANN appears on the right without postprocessing. (This particular ANN was purposely over-trained, to provide a somewhat exaggerated need for postprocessing.) It can be seen, for example, that the ANN has correctly located a split near the center of the image, but has also incorrectly applied the split label to some of the annual rings. The large knot region above the split, and to the left, has an irregular shape that is difficult even for human observers to delineate precisely. For the knot, the ANN has produced several small regions that should be merged or removed for aesthetic reasons. Another problem is the relatively large bark region that appears incorrectly at the top of the knot region.

Many of these incorrect labels have a negligible effect on statistical classification accuracy, which depends on pixel counts alone. Qualitatively, however, the removal of several small regions and the smoothing of region contours can be desirable. Most of the needed changes can be accomplished with relatively simple postprocessing steps. The difficulty lies in the development of rules that determine when to apply these simple steps. For example, the true split region near the center of Figure 1(b) is relatively small in size. A filter that indiscriminately removes all regions smaller than some threshold may also remove valid defect region such as this. Because it is difficult to manually specify an exhaustive set of rules that will work well for all possible situations, the emphasis of this research has been to allow the machine to develop its own rules, based on observations of a human user.



(a)



(b)

Figure 1. Output for a red oak log, without postprocessing. (a) CT slice to be analyzed. (b) Output of ANN classifier. The tan color represents the clearwood region. Red represents knot locations, and we can see several small regions that should be removed. Yellow represents splits, and this label has been assigned incorrectly to several annual rings near the log's center. Brown represents bark, and green represents decay. (For this example, we purposely used an over-trained neural network, which tends to generate output that is relatively low in accuracy.)

3. Interactive Learning Postprocessing System

3.1 System Architecture

Our system, like many supervised machine-learning systems, operates in two different modes: a *learn mode* and a *run mode*. As depicted in Figure 2, during the learn mode our system provides a graphical user interface that allows a human user to edit classified images. The user selects image operators from a menu, designates portions of the image to be processed, and observes the results. This interaction can continue until the user is satisfied with the resulting labeling for any number of training images.

Most image-editing operators provided by the system are based on mathematical morphology ([9], [19]), which is well suited for modifying region shapes. The menu selections include such region operators as remove, fill, split, and merge. A user selects those operations interactively to refine a given classified image. The system observes those actions, and it retains information concerning the regions that were modified. The collected information is stored in a domain knowledge database.

When a user activates the run mode, our system automatically generates a set of rules based on its stored knowledge database. This is illustrated in Figure 3. A user, possibly a different person, can load a new image, and the system will automatically apply its rules to update the image. Based on the geometric properties of those regions, the system selects operations and applies them.

3.2 Decision Tree Construction

A serious problem faced by designers of knowledge-based ("expert") systems is the knowledge acquisition bottleneck. Traditionally, a knowledge engineer works closely with a domain expert to develop rules and guidelines for a particular application. The rules are encoded, and an inference engine (such as MYCIN-style reasoning) is used to analyze new inputs to the system. Because this process is inherently difficult and time-consuming, active research efforts are in progress within the artificial intelligence community to automate the knowledge acquisition process (e.g., [2],[11],[14]).

For our postprocessing application, we have adopted a variation of supervised learning strategies in which the system can learn through observation of a domain expert. Many different approaches to inductive inference could have been adopted in this study, including support vector machines, artificial neural networks, or explanation-based learning. (For example, see ([1],[3],[7],[8],[9],[11],[21].) In our current prototype, we have implemented an approach based on the induction of *decision trees*.

A decision tree is a graph-theoretic tree in which each interior node represents a decision point, conceptually incorporating an IF-THEN-ELSE statement, and each leaf node represents a final class label that should be assigned. A decision tree can be used to encode knowledge for classifying objects, which are often presented in the form of vectors from a feature space.

We have developed a strategy in which computed region properties, along with other image-related properties, comprise the feature space for decision tree induction. As a simple example, consider *region size* and *radial distance* from the log center as two features that can be computed for a given region in an image. It is possible to map any particular region onto a point in this feature space, and to assign a label indicating a desired action, such as "remove this region." This is illustrated in Figure 4, which depicts a hypothetical set of training vectors in a 2-dimensional feature space. Information such as this can be obtained in a relatively straightforward manner in our system.

With such a training set, it is possible to use information-theoretic methods to construct a decision tree that can select an action for all points in the feature space. Among the most common systems for decision-tree induction are ID3, C4.5, CART, and MSM-T ([1], [24]). All of these induction methods create a classification tree by repeatedly subdividing the feature space, using linear univariate thresholds. The result is a set of separating hyperplanes that are parallel to the feature-space axes, with resulting subsets forming a partition of the feature space. Entropy measures are often used to select which feature variable to be considered at each node of the decision tree.

We have implemented a newer decision-tree inference approach, known as OC-SEP, which is based on the work of Street ([5],[22]). This approach offers the advantage of "oblique" linear boundaries, which means that they are not constrained to be parallel to the feature axes (as illustrated in Figure 4b). Instead of measures based on entropy, the method uses orthogonality-based separation criteria for training vectors having different labels. This tends to reduce the sizes of the resulting decision trees, at the expense of some additional computation at each node.

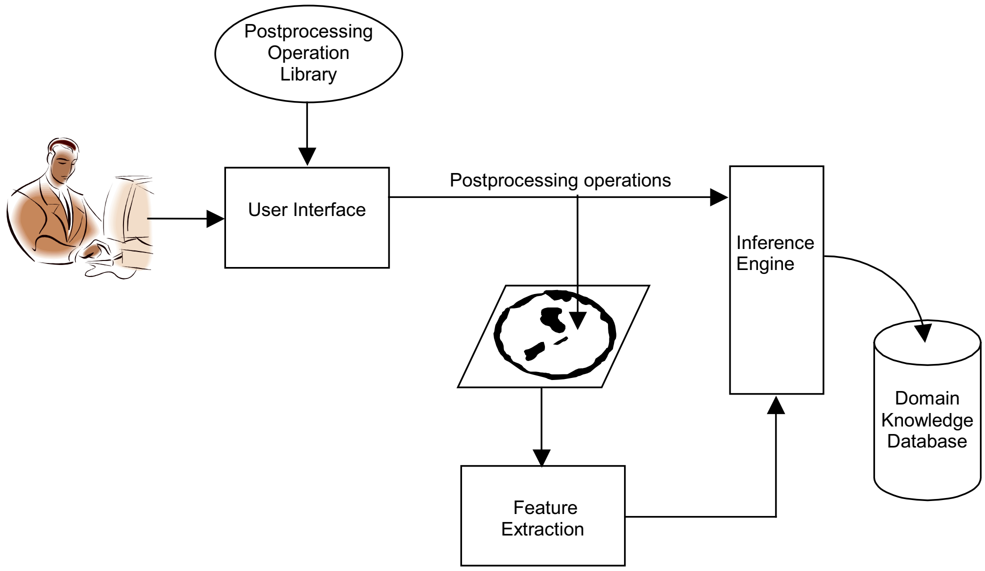


Figure 2: System operation during learn mode.

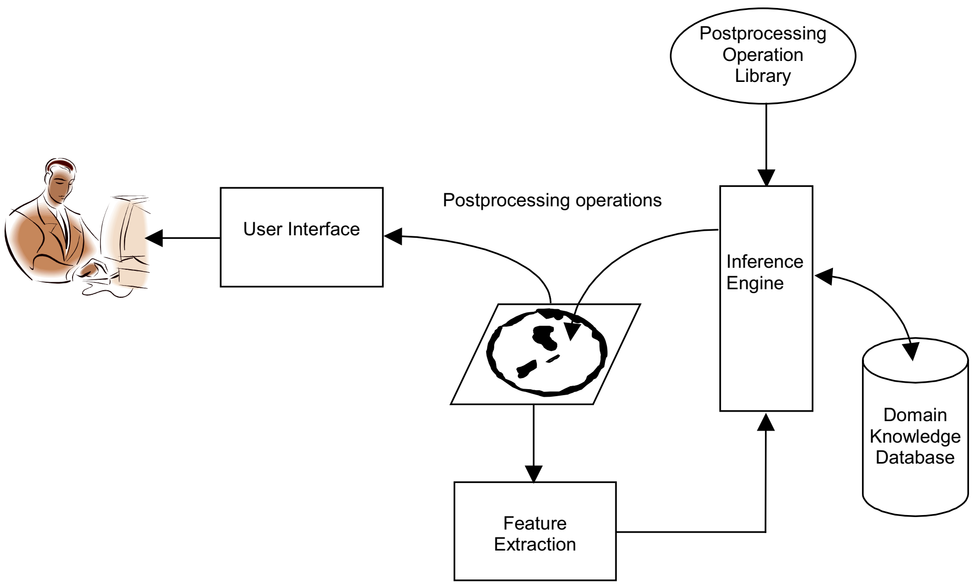


Figure 3: System operation during run mode.

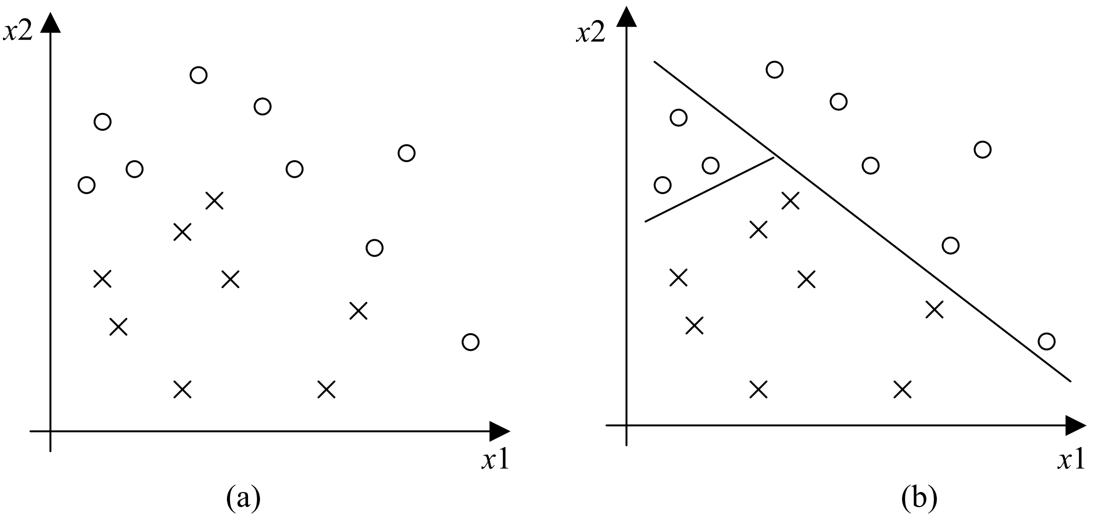


Figure 4. Hypothetical training set for decision-tree induction in a 2-dimensional feature space. (a) The two feature types are represented by x1 *and* x2*, and the two class labels (e.g., "remove region" and "retain region") are indicated by × and o. (b) Linear decision boundaries have been placed in the feature space, separating the two types of training vectors. In a "multicategory" system such as OC-SEP, more than 2 classes are permitted.*

4. Results and Discussion

We have developed a prototype system that illustrates the fundamental approach for learning postprocessing rules. The user interface, shown in Figure 5, allows the user to select either learn mode or run mode for the session. During learn mode, the user can select regions for modification. For the example results shown here, the only action considered is region removal. The user selects a particular "layer" to modify at any given time, where each layer of the image corresponds to a label assigned by the ANN ("knot," "decay," "split", "bark," or "clear wood"). As the user removes particular regions, the system computes and stores information for each region (region size, radial distance, and layer type) in a feature table.

When the user selects run mode, the system automatically calculates and stores feature vectors for the remaining regions in the image, and notes that these are examples of regions that should be retained. The system then constructs a decision tree based on these training vectors. When the user loads a new image to be postprocessed, the system automatically considers each region in the image, and consults the decision tree to determine which of those regions should be removed and which should be retained.

As an example of system operation, Figure 6 shows a CT slice that was used to train the system. Beginning with the image in part (b) of the figure, the user removed several regions from the different layers to obtain the result shown in (c). Because the image in (b) contains a total of 177 distinct foreground regions, the system automatically collected 177 feature vectors during the training session. The system then automatically constructed a decision tree using this information.

Using this feature table in run mode, the system postprocessed several image slices. (These images were not used in training the system.) Three of these images are shown in Figure 7: one from the same log (top row), and two from a different red oak log. Column (b) contains images that were tentatively classified by an ANN, and column (c) presents images that were refined by our system automatically.

For all three image slices, the removal of small regions has improved the quality of the results considerably. For the example in the top row, several small split regions have been correctly removed from the center portion of the image. Other spurious regions, incorrectly labeled split, decay, and clear wood, have been removed from the bark area. Of particular interest is one narrow region incorrectly labeled decay, near the left edge of the log, that has been removed based on the rules encoded in the decision tree. Although it may be difficult to see in the figure, this region is relatively large in area, and would not be removed by a filter based solely on small region size. Our system has learned rules based on both size and distance from the center of the log slice, and that causes this region to be removed in spite of its size.

The middle and bottom rows of this figure contain slices from a different log, with large decay regions near the center. The ANN has incorrectly labeled several small regions as split and knot, and the postprocessing system has removed them. The postprocessing system has also cleaned up the bark portion of these images considerably.

As a final example, Figure 8 shows a postprocessed version of the image from Figure 1. Most of the postprocessing operations were performed automatically and correctly, but for this case an additional unlearned operation was performed. The large bark region near the top of the image, which borders the knot, is actually an extension of the correctly labeled bark region from the perimeter of the log. Hence, region removal is not appropriate in this case. Instead, the system employed a "hard-coded" region-split operation, and we include this example as an illustration of the potential of the system to perform region modifications that are more sophisticated than simple region removal. Work is currently in progress to automate the learning of this operation, as well as others.

These results were obtained using a PC (Pentium 4, 1.13 GHz, 256 MB), and the system is implemented with Matlab (R13, version 6.5). The postprocessing required approximately 10 seconds per image. As described earlier, the system is capable of incremental learning, which means that the user does not have to have available all training images initially. This is possible because all training vectors are retained from all previous training sessions. When converting to run mode, a full decision tree is computed from the full set of training vectors.

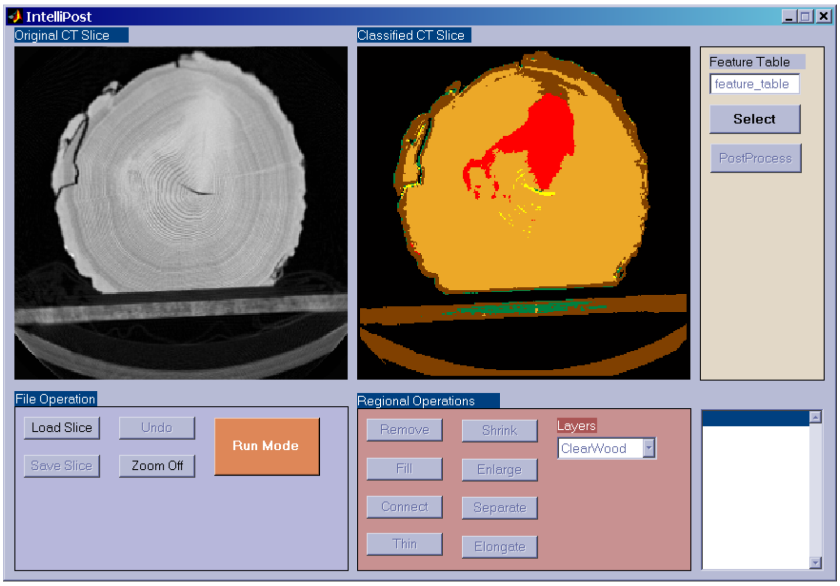
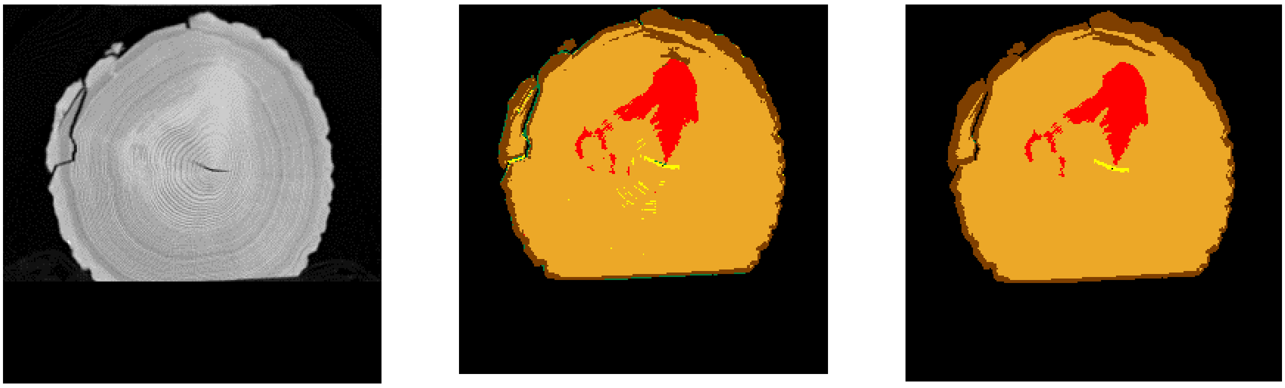
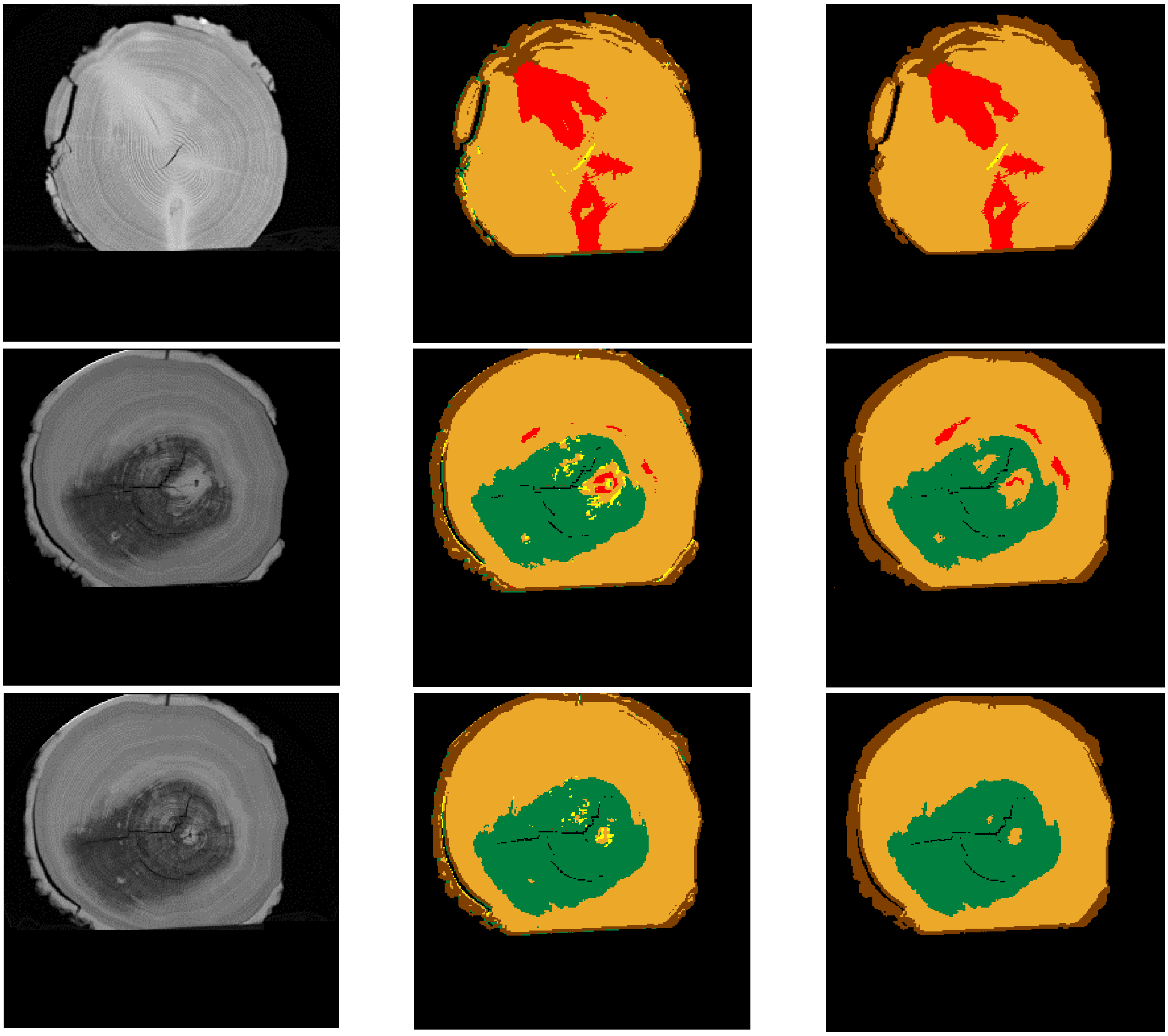


Figure 5: Screen capture of the postprocessing system's user interface. The input image appears on the left, and the classified image (being modified) is on the right. In learn mode, the user modifies the segmented image interactively. In run mode, as shown above, the system operates automatically.



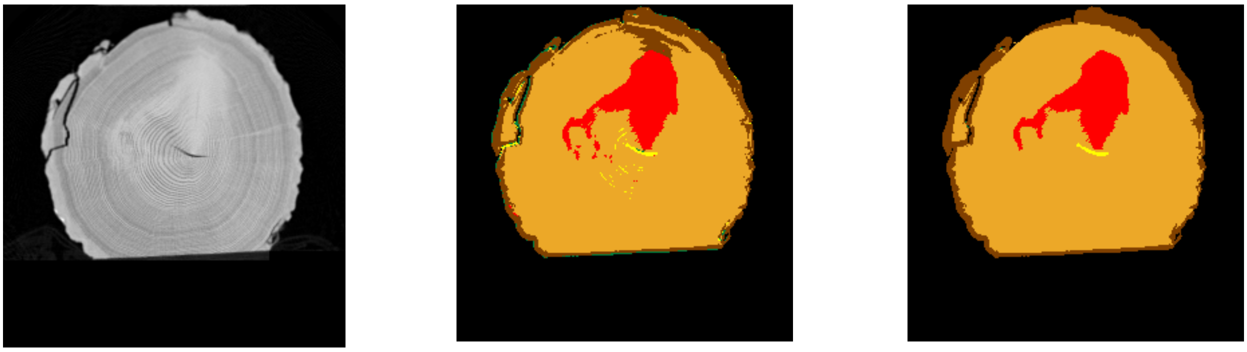
(a) (b) (c)

Figure 6: The CT slice from a red oak log used for training. (a) Original CT image. (b) Result from ANN classification. (c) Result of postprocessing, obtained by manually modifying the image in (b).



(a) (b) (c)

Figure 7: Classification results for three red oak log slices, taken from two different logs. (a) The original CT images. (b) Initial classifications performed by ANN. (c) Automatic postprocessing results, in which many spurious regions have been removed.



(a) (b) (c)

Figure 8: Classification results for the red oak slices from Figure 1. (a) The original CT image. (b) Initial classifications performed by ANN. (c) The result of post-processing. In addition to automatic learning, the system has applied a "hard-coded" operation to remove a portion of the large bark region near the top of the image.

5. Conclusions

This paper has introduced a new approach for refining images in which regions have been detected and labeled. We have developed a prototype system that observes the actions of a human operator who interactively edits a set of test images. The system then applies automated inferencing techniques to develop its own postprocessing rules based on those actions. After this learning process, the system is capable of automatically applying similar refinement steps to other images.

The system does not simply memorize a sequence of operations by the user, as is often used for robotic teach pendants. Instead, the system develops more general rules based on labeled region properties, such as size, elongation, defect type, and position in the image. Although this approach has been developed particularly for use with CT image slices of hardwood logs, it is sufficiently general that it can be used for other applications, such as refining aerial image classification.

Acknowledgments

This work was supported in part by USDA National Research Initiative competitive grant #98-35504-6581.

References

[1] Andreae, P., "Justified generalization: acquiring procedures from examples", Ph.D. thesis, Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, 1985.

[2] Barstow, D., *A Knowledge-Based Program Construction*, North-Holland, New York, 1979.

[3] Bauer, M., "Programming by examples", *Artificial Intelligence*, Vol. 12, No. 1, pp. 1-21, May 1979.

[4] Bhandarkar, S.M., Faust, T., and Tang, M., "A system for detection of internal log defects by computer analysis of axial CT images", *Proceedings of IEEE Workshop on Applications of Computer Vision*, Sarasota, FL, pp. 258-263, Dec. 1996.

[5] Bennet, K.P., "Decision tree construction via linear programming," *Proceedings of the 4th Midwest Artificial Intelligence and Cognitive Science Society Conference*, Utica, IL, pp. 97-101, 1992.

[6] Burgess, A.E. "Potential application of imaging techniques to wood products," *Proceedings of 1st International Conference on Scanning Technology in Sawmilling*, San Francisco, CA, pp. VII 1-13, 1985.

[7] Cypher, A., "Eager: programming repetitive tasks by example", *Proceedings of ACM Conference on Computer Human Interaction,* New Orleans, LA, pp. 33 - 39, May 1991.

[8] Fisher, D., "Knowledge acquisition via conceptual clustering", Technical Report No. 8722, Department of Information and Computer Science, University of California-Irvine, 1987.

[9] Halbert, D., "Programming by example", Ph.D. thesis, Department of Electrical Engineering and Computer Science, University of California-Berkeley, 1984.

[10] Haralick, R.M., Sternberg, S.R., and Zhuang, X., "Image analysis using mathematical morphology," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. PAMI-9, pp. 532-550, 1987.

[11] Kurlander, D., "Graphical editing by example", Ph.D. thesis, Department of Computer Science, Columbia University, 1993.

[12] Li, P., Abbott, A. L., and Schmoldt, D. L., "Automated analysis of CT images for the inspection of hardwood logs", *Proceedings of the IEEE International Conference on Neural Networks*, Washington, DC, pp. 1744-1749, 1996.

[13] Lieberman, H., "Communication of expert knowledge in graphic design", Internal document, Visible Language Workshop, Massachusetts Institute of Technology, 1988.

[14] Lieberman, H., "Capturing design expertise interactively by example", *Proceedings of the East - West International Conference on Human-Computer Interaction*, ICSTI, St. Petersburg, USSR, pp. 360-366, August 1992.

[15] Maulsby, D., "Prototyping an instructible interface: Moctec", *Proceedings of ACM Conference on Computer Human Interaction '92*, Monterey, CA, pp. 153-154, May 1992.

[16] Maulsby, D., Witten, I., Kittlitz, K. and Franceschin, V., "Inferring graphical procedures: the complete metamouse," *Human-Computer Interaction*, Vol. 7. No. 1, pp. 47-89, 1992.

[17] Occeña, L. G., "Computer integrated manufacturing issues related to the hardwood log sawmill", *Journal of Forest Engineering*, Vol. 3, No. 1, pp. 39-45, 1991.

[18] Schmoldt, D.L., Li, P., and Abbott, A. L., "Machine vision using artificial neural networks and 3D pixel neighborhoods," *Computers and Electronics in Agriculture*, Vol. 16, No. 3, pp. 255-271, 1997.

[19] Schmoldt, D.L., He, J., and Abbott, A.L., "Automated labeling of log features in CT imagery of multiple hardwood species", *Wood and Fiber Science*, Vol. 32, No. 3, pp. 287-300, 2000.

[20] Serra, J., *Image Analysis and Mathematical Morphology*, Academic Press, London, 1982.

[21] Smith, D., *Pygmalion: A Computer Program to Model and Stimulate Creative Thought,* Basel, Stuttgart, Birkhauser Verlag, 1977.

[22] Street, W. Nick, "Oblique multicategory decision trees using nonlinear programming", to appear in *INFORMS Journal on Computing*.

[23] Tsolakides, J.A, "A simulation model for log yield study", *Forest Products Journal*, Vol. 19, No. 7, pp. 21-26, 1969.

[24] Quinlan, J.R., "C4.5: Programs for machine learning," Morgan-Kaufman, San Manteo, CA., 1993.

[25] Quinlan, J.R., "Induction of decision trees," *Machine Learning*, Vol. 1, pp. 81-106, 1986.

[26] Zhu, D. P., Conners, R.W, and Schmoldt, D.L., "Nondestructive evaluation of hardwood logs using automated interpretation of CT images," *Proceedings, Review of Progress in Quantitative Nondestructive Evaluation*, edited by Thompson, D.O., and Chimenti D.E., Plenum Press, New York, Vol. 12, pp. 2257-2264, 1993.