



A fast pith detection for computed tomography scanned hardwood logs

Rado Gazo^{a,*}, Juraj Vanek^b, Michel Abdul_Massih^b, Bedrich Benes^b

^a Department of Forestry and Natural Resources, Purdue University, 175 Marsteller Street, West Lafayette, IN 47907-2083, United States

^b Department of Computer Graphics, Purdue University, 401. N. Grant Street, West Lafayette, IN 47907, United States



ARTICLE INFO

Keywords:

Computed tomography
Log scanning
Hardwood
Pith
Automatic detection
CUDA
GPU

ABSTRACT

Tree and log evaluation prior to processing is traditionally conducted by visual inspection of the outside surface. This approach has obvious limitations, because many internal features cannot be directly observed. Just recently, computed tomography-based (CT) approaches allowed visualization and analysis of the internal wood structure. The data gathering reveals the internal structure in a form of successive 2D discrete images in limited resolutions in both spatial and value domains. However, the size of scan data can be very large (gigabytes) and their processing using traditional approaches can be time-intensive. There is a need for classification and quantification of internal log defects in real time to keep up with processing speed at modern mills. The aim of this study was to develop a real-time pith detection from CT-scanned log data. The speed necessary for real-time processing is achieved in two ways: first, by adaptive method that uses precise detection only when necessary; and second, by parallel processing power of graphic processors (GPU) that are more suitable for parallel data processing of large datasets than classical central processing units (CPU). The input of our system is a set of 2D images that were collected during the CT scanning and the output is a set of locations within the slice that have been identified as pith. Results of our algorithm tested on data from North American species of Black Cherry, Black Walnut, Hard Maple, Red Oak, White Oak, and Yellow Poplar show that on average, the algorithm found pith with precision of 4.2 mm as compared to manual pith detection. The GPU acceleration by using CUDA enables processing speed of about 0.003 s per image with high precision. This makes the developed algorithm suitable for an industrial application in hardwood sawmills and veneer slicing operations.

1. Introduction

First step in manufacturing of many wood products is harvesting trees. Once felled, trees are cut into shorter sections, logs, to accommodate transport to sawmills and veneer mills for further processing. For centuries, the logs are cut into lumber or veneer using only the external log characteristics to deduce the interior quality of wood. Chang and Gazo (2009) and Gazo and Chang (2012) showed that processing optimization based on knowing what is “inside the log” can increase the value of lumber produced by an average of 46%. The poorer the quality of the log, the higher the potential improvement. This improvement is possible because the log can be oriented prior to processing in such a way, as to place most internal defects in fewer boards of poor quality, leaving the majority of the boards clear of defects, and thus of higher value. Internal defects that typically reduce the value of a board, if present, are knots, splits and cracks, pith, decay, insect damage, mineral streaks, bird pecks, etc. This study was the basis for development introduction of CT.Log™ in 2014, the world’s first industrial grade CT scanner for logs by Microtec, GmbH. At the same time,

Gazo and Benes (2013), Gazo et al. (2013, 2014) developed a visualization and optimization software LogView™, which works with the CT.Log™ scanner to optimize the lumber and veneer production. To date, there are eight such systems installed globally. The LogView™ software first identifies and classifies internal characteristics inside the log and then uses this information to simulate the cutting operation. The virtual boards are then graded for quality and assigned monetary value. Processing parameters are changed and process is repeated several hundred thousands of times. At the end, the optimization algorithm selects the most feasible simulation solution. In order for this to work, the LogView™ has to be able to detect internal features of logs that are considered defect by the lumber industry governing body (NHHLA, 2019) as described above. This paper focuses on description of pith detection as used in LogView™ software. The cycle time for this process in an industrial setting is between 20 and 40 s, thus the need for fast detection algorithm.

Pith is important in hardwood lumber processing because it is considered a grading defect according to the National Hardwood Lumber Association grading rules (2015). Both size and location of pith

* Corresponding author.

E-mail address: gazo@Purdue.Edu (R. Gazo).

<https://doi.org/10.1016/j.compag.2019.105107>

Received 26 April 2019; Received in revised form 10 November 2019; Accepted 12 November 2019

Available online 01 January 2020

0168-1699/ © 2019 Elsevier B.V. All rights reserved.

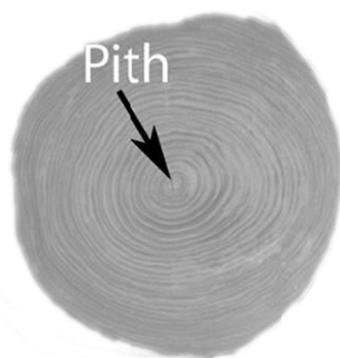


Fig. 1. Pith position on a cross-section image from CT-scanned log.

are factors in determining of board grade and thus its value. In decorative hardwood veneer slicing, knowing the exact location of pith is important for determining how a high-value log is going to be “flitched” into two halves (optimally right through the pith) prior to slicing. The lumber quality is determined by the amount and shape of the clear areas (areas free of defects) on a board’s surface and this can only be estimated before the log is processed. Zhu et al. (1996) reports that an improvement of approximately 10–21% in lumber value can result from a well-chosen breakdown strategy. Our own study showed an even higher potential improvement.

This project is concerned with determining the *pith* position from cross-section images of CT scanned logs. An illustration of the pith position in a cross-section image of a log can be seen in Fig. 1. Knowing the pith position inside logs is useful for sawyers and scientists to better understand their inner structure. If a cross-section of the log is taken, the pith will rarely be at its geometric center. The location of the pith is useful to detect knots and cracks because the longitudinal axes of knots and cracks normally pass through it, and for knotty wood, the pith position represents the origin of branches. Because annual rings form around the pith, knowing its position is important in stem analysis that seeks to understand the tree growth response to environmental factors and reconstruct its entire growth record (Cerde et al. 2007). The pith location can also help in determining growth ring eccentricity (Saint-André and Leban 2001), spiral grain (Sepúlveda 2001), etc. Most of the methods for pith detection from a CT scan or a photograph use the Hough Transform as a means of locating the pith position (Hough, 1962).

There are several methods published that automatically or semi-automatically detect internal log defects and other properties from CT scans or photographs of the end cross-sections, mostly relying on Hough Transform to detect centers of concentric circles.

Bhandarkar et al. (1996) detect knots and cracks on log CT scans by first identifying defect-like regions based on their connectivity and then analyzing those regions through different morphological and orientation features. The system is enhanced to do 3D reconstruction of the log and simulate board sawing and veneer slicing to help the user make decisions about the breakdown of the log (Bhandarkar et al. 1999). Andreu and Rinnhofer (2002) enhance CT scan by adaptively changing the contrast of the ridge and valley structure of the growth rings. They use the Fourier Transform to compute orientation and frequency in local neighborhoods as opposed to most methods, which use the Sobel operator to obtain the gradient of those neighborhoods. Cerde et al. (2007) developed an automatic growth ring detection algorithm using the Hough Transform both to detect the growth rings and the pith. The results are compared with the observations of an expert in dendrology. The algorithm is not tested among different species of trees so it is difficult to say what would be the general accuracy. Longuetaud et al. (2004) presented a detailed literature review on this topic and an implementation of pith detection technique working also on images with knots as such slices are interpolated from clear slices. This approach

was improved by Boukadida et al. (2012) and tested on a large number of images with high precision. Norell and Borgefors (2008) developed a method to approximate pith position in the sawmill environment from the digital photos of both ends of the log that is fast and suitable for on-line processing but without knowledge of internal structure. Schraml and Uhl (2013) used Fourier Spectrum Analysis on 36 CT images and 109 digital images of a single spruce log each to estimate pith position. Most recently, Kurdthongmee et al. (2018) proposed three fast algorithms for approximate pith location in rubberwood timber from digital images of log ends.

2. Methodology

2.1. Overview of the method

The pith location is detected by using a predictor-corrector strategy. We locate the pith only in some slices by using a computationally intensive method and in others using a fast method. The prediction is based on the pith deviation between two slices found with the fast method. If the result from the fast method diverges from the result of a previous slice in less than a predefined value, this location is kept. Otherwise, the pith location is updated by using the more computationally intensive method.

The computationally intensive method uses a parallel version of Hough Transform for circle detection to locate the pith. To compute the Hough Transform, the gradient directions and edges from an image are calculated. Let $[x, y]$ denote the coordinates of the position of the plane of a cross section. An accumulator array will be initialized and the accumulator at position $[x, y]$, will be incremented if there is a line segment that starts at an edge point with the direction of the gradient at that point. Then, the coordinate with the highest value of the accumulator will be the location where most line segments intersected and thus, where the pith is most likely to be located since it is contained inside the growth rings of the log.

In the computationally faster method, the lowest density of the input image is computed and the result is regarded as pith position, given the condition that the distance to the pith position from the previous image is no higher than a predefined value. For the very noisy data with barely visible growth ring structure, the first two methods can fail. In that case, they are replaced with a rough approximation of pith from the log outer boundary that is clearly distinguishable from the background. Pith is approximated by averaging the positions of points on the log boundary, marking the pith in the middle of the log as shown in Fig. 2.

The slices are processed preferably in the sequential order in the direction from the top of the tree to the bottom (downward), as this provides better accuracy on logs where double pith, large branches or forks are present. Processing in the upward direction, from the bottom of the tree up, could lead to unwanted detection of the secondary pith, as shown by Boukadida et al. (2012).

After the first pass, all cross-section images of the entire log are processed and pith locations are found in each image. In the second pass, a weighted average scheme is used to further improve the results by using frame-to-frame coherency. The final step is the visualization of the cross-section images of the log as well as whole log in 3D, showing the resulting pith position and the minimum volume containing the pith (Figs. 2 and 3).

The most computationally intensive part is Hough Transform. A log can be scanned into several hundred slices and the computation for the entire log could take more than one minute, making it unfeasible for real-time processing required in an industrial application at a sawmill. Our solution to this problem is two-fold. First, we accelerate the calculations by running them in parallel on the graphic processor (GPU). Second, we use predictor-corrector method – only if the error from an approximate method diverges too much, we apply the precise method.

Test data for our method were composed from 22 logs of six

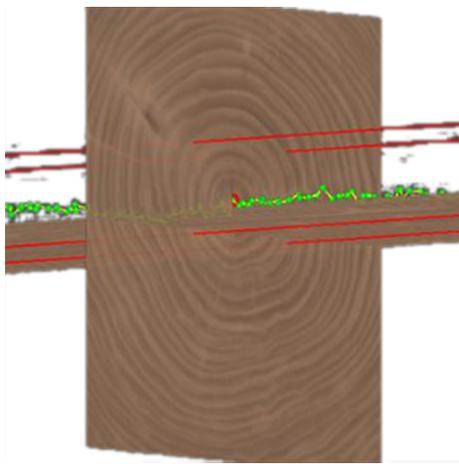


Fig. 2. Cross-section of a log. The red lines demarcate the maximum excursion of pith in X and Y throughout the entire length of the log. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)



Fig. 3. Visualization of pith in 3D showing its location along entire log.

different species (Table 1). As is customary in North American hardwood sawmills, log length is rounded down to nearest whole foot. Small and large end diameter inside bark is given in inches. Each reported log diameter is an average of smallest and largest diameter reading at each end. The sample species included black cherry (*Prunus serotina* Ehrh.), hard maple (*Acer saccharum* Marsh.), yellow poplar (*Liriodendron tulipifera* L.), black walnut (*Juglans nigra*), red oak (*Quercus rubra* L.), and white oak (*Quercus alba* L.). All logs are believed to have come from northeastern Indiana, US. Each species group consisted of logs from each of US Forest Service Log Grades 1, 2, 3, and veneer, thus providing a variety of shapes, sizes and internal characteristics. The logs were scanned with a GE HiSpeed CT/I medical X-ray scanner temporarily installed in a sawmill. Each log contained up to 1,000 image slices with resolution 512×512 (0.93 mm per pixel) and slice spacing of 1 mm. Data consisted of 256 shades of gray (8-bit) and data size was 200 MB on average. As we wanted to test the algorithm robustness, the data contained many common defects like knots, cracks and decay, not just clear wood.

The pith detection algorithm is divided into two main parts: the *detection* and the *correction*. First, the detection is carried out over the entire set of cross-section images of the log and then correction of the results is done, along with the calculation of the maximum excursion of pith in X and Y throughout the entire length of log. The detection stage itself is divided into detection by using the Hough Transform, detection through the search of the lowest value on the input image, and

detection from the outer boundary.

In the context of this work, the method for detecting the pith location through the Hough Transform will be referred to as *Method HT* (*Hough Transform*), the method for finding the lowest value from the cross-section will be referred to as *Method MIN*, and the method to roughly approximate the pith from outer boundary will be referred to as *Method BA* (*boundary approximation*).

The outline of the algorithm is as follows:

1. Use precise *Method HT* to detect pith on image (will serve as reference)
2. Use approximate *Method MIN* to detect pith on the next image
3. If the distance between the results from steps 1 and 2 is higher than threshold, use *Method HT* on image
4. Repeat steps 1, 2 and 3 for the entire log
5. Correct results by weighted average
6. Compute minimum volume containing pith
7. Visualize

This outline is valid if data is of good quality. If the quality is poor (e.g., no growth rings are visible), steps 1–3 will be replaced by using *Method BA* only, and this method will be used for the entire log. The slices are processed from the top to the bottom of the log, as this order is more accurate when tree forks and double pith is present.

2.2. Pith detection by using the modified Hough Transform for circles (*Method HT*)

Hough Transform for circles has been successfully used for accurate pith detection and our approach is similar to Longuetaud et al. (2004) with necessary modifications for parallel GPU implementation using CUDA programming interface. First, the data needs to be uploaded into GPU memory. In case the data are too large and might not fit into GPU memory, we divide it into sections perpendicular to the main log direction and process them independently. After the upload, all slices and all pixels in each slice are processed in parallel. Although the images may appear to have low visual contrast, it does not affect the pith detection algorithm that deals with gradients and edge detection.

Running in parallel in CUDA means that all pixels of the slice are processed at once on multiple CUDA threads. Each thread will process one pixel and for each pixel, we first compute gradient information from the input gray-scale image and then apply a Sobel filter in horizontal and vertical direction that requires two passes with thread synchronization. The Sobel filter is a convolutional detector that generates a gradient of the input image as shown in image in Fig. 4.

The next step is to compute the edges of the gray scale image. For this project, an implementation of the Canny edge (Canny 1986) detector on CUDA was selected. While the Sobel operator generates a smooth gradient, the Canny edge detector provides a binary image with edges. Fig. 5 shows the result of the edge detection on one of the gray-scale images.

Most of these edges represent the growth rings of the log that can be thought of as a set of concentric circles or ellipses and the pith is contained by the growth rings. The key purpose of our detector is to generate perpendicular lines to the growth rings in the direction of the gradient and find their intersections. To increase the speed of computation and to make it more robust, we analyze only a small portion of the image around the previously detected center of pith. We incremented the value from zero by a fixed step and we found that the portion within the half of the log diameter proved to give satisfactory results.

Although the intersection could be generated analytically, we use a faster approach that is based on image processing. For each of the edge points, a line is produced with the gradient information into the accumulator. The accumulator is a 2D image initialized with zeros and line segments produced from the gradient information at every edge

Table 1

Overview of species, logs, average errors and processing time. Downward refers to images processed in the order from top of the tree down to the bottom. Upward refers to the opposite direction.

Species/Log	Length [feet]	Small End Diameter Inside Bark [inch]	Large End Diameter Inside Bark [inch]	Log Grade	Number of Slices	Average Error Downward [pixel]	Average Error Upward [pixel]	Average Error Downward [mm]	Average Error Upward [mm]	Processing Time [seconds/ image]
Black Cherry 1	16	16.75	20	1	980	3.9	3.9	3.63	3.63	0.003490
Black Cherry 2	16	16.5	17	2	1000	2.9	3.1	2.70	2.88	0.002960
Black Cherry 3	10	10.25	11.75	3	640	1.8	1.8	1.67	1.67	0.002563
Black Cherry 4	16	15.5	18	1	1000	2.8	2.7	2.59	2.51	0.002950
<i>Black Cherry average</i>						<i>2.8</i>	<i>2.9</i>	<i>2.65</i>	<i>2.67</i>	<i>0.002991</i>
Black Walnut 1	9	17	18	veneer	600	3.7	3.7	3.44	3.44	0.003367
Black Walnut 2	9	17	18	veneer	580	2.7	2.7	2.51	2.51	0.003155
Black Walnut 3	14	14	16	veneer	720	2.7	2.1	2.51	1.95	0.003167
Black Walnut 4	10	15.5	17	veneer	660	2.1	2.3	1.95	2.14	0.003030
<i>Black Walnut average</i>						<i>2.8</i>	<i>2.7</i>	<i>2.60</i>	<i>2.51</i>	<i>0.003180</i>
Hard Maple 1	16	16	20.5	1	985	6.3	6.0	5.86	5.58	0.002985
Hard Maple 2	10	14.5	14.75	3	660	2.8	3.3	2.60	3.07	0.002848
Hard Maple 3	14	14	16.5	2	880	3.9	5.1	3.63	4.74	0.002830
Hard Maple 4	10	14.25	18.5	1	630	2.4	2.9	2.23	2.70	0.002810
<i>Hard Maple average</i>						<i>3.9</i>	<i>4.3</i>	<i>3.58</i>	<i>4.02</i>	<i>0.002868</i>
Red Oak 1	10	16.25	22.5	1	660	11.3	9.2	10.51	8.56	0.003379
Red Oak 2	10	15.25	16	3	640	3.0	3.2	2.79	2.98	0.002969
Red Oak 3	12	16.5	18.25	1	735	5.5	5.8	5.12	5.39	0.003483
Red Oak 4	10	13.5	13.5	2	680	3.3	3.6	3.07	3.35	0.002779
<i>Red Oak average</i>						<i>5.8</i>	<i>5.5</i>	<i>5.37</i>	<i>5.07</i>	<i>0.003152</i>
White Oak 1	16	14	20.5	1	960	14.2	23.8	13.21	22.13	0.003042
White Oak 2	12	13	13	3	780	3.3	26.5	3.07	24.65	0.002974
White Oak 3	10	15.5	17.25	2	615	4.9	15.7	4.56	14.60	0.003252
<i>White Oak average</i>			<i>22.5</i>			<i>7.5</i>	<i>22.0</i>	<i>6.94</i>	<i>20.46</i>	<i>0.003089</i>
Yellow Poplar 1	14	15	20.25	2	900	3.6	3.4	3.35	3.16	0.003444
Yellow Poplar 2	14	19	16.25	1	880	3.6	3.8	3.35	3.53	0.003761
Yellow Poplar 3	12	15.5		3	780	5.4	3.3	5.02	3.07	0.002962
<i>Yellow Poplar average</i>						<i>4.2</i>	<i>3.5</i>	<i>3.91</i>	<i>3.26</i>	<i>0.003389</i>
Species average						4.5	6.8	4.18	6.33	0.003112

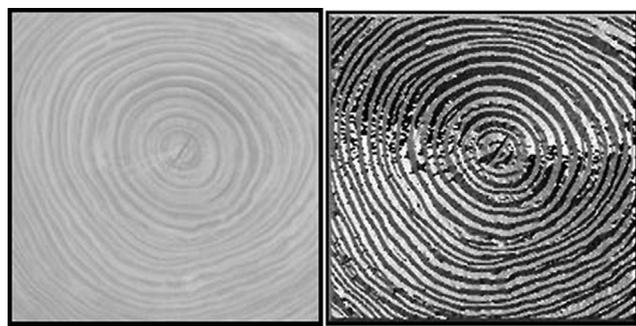


Fig. 4. (a) Original gray-scale image. (b) Gradient image computed with the Sobel filter.



Fig. 5. Canny filter applied to input gray-scale image.

location and rasterized into the accumulator. Every point that the line segments intersect is recorded in the accumulator image by increasing the image value at that point. Inspired by the Hough Transform, where the increase at the point would be by one, we weight the linearly increment according to the distance from the outer edge, so that the external growth rings will not have as much weight as the internal ones in the final result. External growth rings may be more distorted than the inner ones and do not point directly to the pith. A second heuristics uses density information and lines going through areas with lower density, where pith is more likely to be located, are given higher weight. In Fig. 6 an accumulator image obtained from a cross-section image of the log is shown.

A GPU implementation must take care during the parallel computing with race conditions that occur when rasterizing the lines into an accumulator. It introduces issue with shared memory writes, as each

thread is writing into a different part of the accumulator. The solution was to use atomic reads and writes, assuring that threads will not access the same location at once. A disadvantage of atomic operations is a performance slowdown but this solution is still ten times faster than the equivalent CPU implementation.

The accumulator image is then smoothed by applying a Gaussian filter and the highest value from the resulting image is taken as the pith position on the input image. The highest value is obtained using parallel reduction on the GPU for faster calculation (see nVIDIA CUDA Toolkit (2014) for details about parallel reduction).

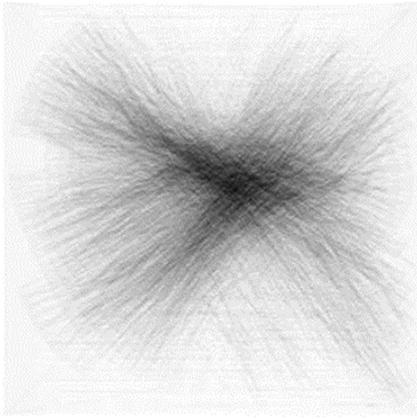


Fig. 6. Accumulator image obtained from a cross-section image of the log. Intersection of the gradient edges approximates the pith location.

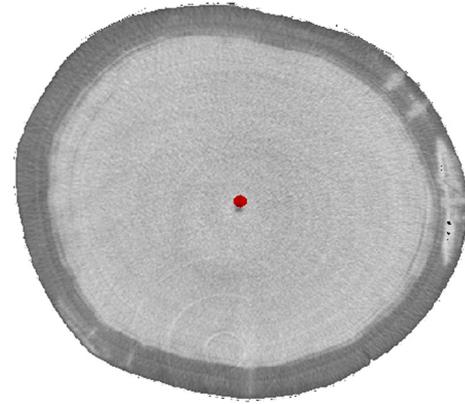


Fig. 7. Very rough approximation of pith position from detected outer boundary on noisy data where ring structure is not clearly visible.

2.3. Lowest density cross-section log image (Method MIN)

Finding the precise pith location using the Hough Transform is computationally intensive despite the use of GPU. Method MIN avoids the computation of the Hough Transform because in CT scan cross-sections of many logs, the pith can be located by the lowest density location of the image. It is worth noting that this method may not be used when many knots are present or with certain tree species (e.g., Norway spruce, as shown by Longuetaud and Caraglio (2009)).

No smoothing is performed on the original image prior to obtaining the lowest value. This decision seems counter-intuitive, but when testing the algorithm, better results were obtained when the image was not smoothed. This indicated that, while in some cases the pith is represented by the lowest value in the image, the average values around the pith are not always the lowest. The lowest value of the cross-section image is again found using parallel reduction on the GPU.

2.4. Pith detection from outer boundary (Method BA)

Using the Hough Transform on noisy images without clearly visible growth ring structure leads to incorrect detection, as this method requires extracted growth rings to compute gradients of lines. In this case, a simpler, more robust, albeit less precise algorithm has been proposed to get at least a general approximation of pith in low quality scans. Extraction of the outer boundary works as follows. For each pixel in the slice:

- Select 3×3 surrounding pixels
- Add to boundary pixel list if at least three of its neighbors are empty (=background)

Output of this step is a list of pixels on the outer boundary. Pith is found by a simple averaging of all pixel coordinates and the result is a point in the middle of the log boundary (Fig. 7). Note that this method will not find the pith for eccentric logs.

To decide which method to use, several random slices (3 slices provide sufficient results in our implementation) are selected from the data and both Methods HT and BA are used to find the pith. If the difference between detected pith positions is larger than the threshold, we assume that Method HT has failed, as it is more sensitive to poor data quality. Then, Method BA will be selected as the last resort to perform pith detection of the entire log.

2.5. Pith position correction

When the pith position (located in the $[x, y]$ plane perpendicular to the main log direction) for the whole set of cross-section images of the

log has been computed, a correction stage takes place to further improve the results. We took a similar smoothing approach as Boukadida et al. (2012). The pith positions are stored in an array and for each value; the previous 10 elements and the next 10 elements are averaged with weight. The closer the array index is to the current index, the greater the weight it will have.

After obtaining the weighted average for the position, the distance between this point and the average is calculated. If the distance is low (less than 5 pixels in the actual implementation), then the current position is not modified. For higher values, the current position is replaced by the average. This test helps to filter out the outliers that would otherwise not be removed by smoothing. Fig. 8 shows an example of some corrected values.

A software implementation of the proposed pith detection algorithm is presented. The Hardwood Scanning Center at Purdue University previously developed a database of CT scans of 65 hardwood logs of 6 species and 3 log grades within each species (Gazo and Chang 2010). A random subset from this database was used for verification. The program was tested on 22 logs with an average of 768 cross-section images each.

The testing was done on an Intel™ i7 920 CPU clocked at 2.67 GHz with 12 GB of the main memory and a nVIDIA™ Quadro K5000 with 4 GB of memory. The log data were read from an Intel SSD drive. The input images were in the resolution of 512×512 pixels.

3. Results and discussion

On average, the GPU implementation took 0.003 s per slice, or 2.3 s per log consisting of 768 image slices to process the pith. Compared to the CPU implementations running by Boukadida et al. (2012), our accelerated solution was 17 times faster and it allows for real-time processing. In an industrial setting, when higher precision can be sacrificed for speed, the detection can run even faster by not analyzing every slice.

To test the algorithm, an expert manually annotated 22 logs from six different species: Black Cherry, Black Walnut, Hard Maple, Red Oak, White Oak, and Yellow Poplar (Fig. 9). The pith in hardwood logs, as defined by NHLA grading rules (NHLA, 2019), is a circular area in cross-sectional image that is approximately 20 pixels in diameter. A single pixel nearest the center of this area was manually annotated as “pith” on each slice for study comparison purposes. To demonstrate the robustness of our solution, the algorithm was tested on data that in addition to pith contained a variety of defects such as knots, cracks and decay. This was compared to the algorithm, and the pixel distance from manually annotated to automatically detected pith on each slice for each log was considered the significant variable. Results of the comparison of the automatic and manual methods is shown in Table 1.

The average distance between algorithm-found and manually marked pith center was 4.5 pixels. It is important to note that pith in

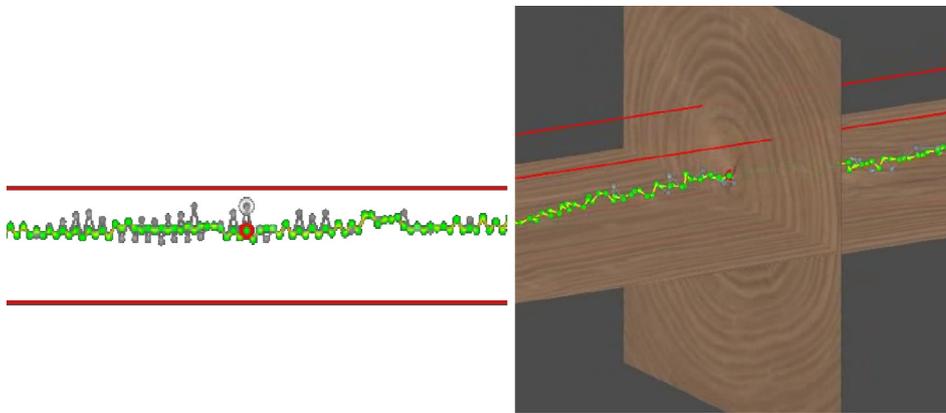


Fig. 8. Example of corrected values. Green dots represent the actual pith positions for a number of consecutive cross-sections and the gray dots represent the positions before being corrected. The red dot represents the position of the current image and the white dot is the value before being corrected. This image only shows the correction in the y axis to illustrate the idea. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

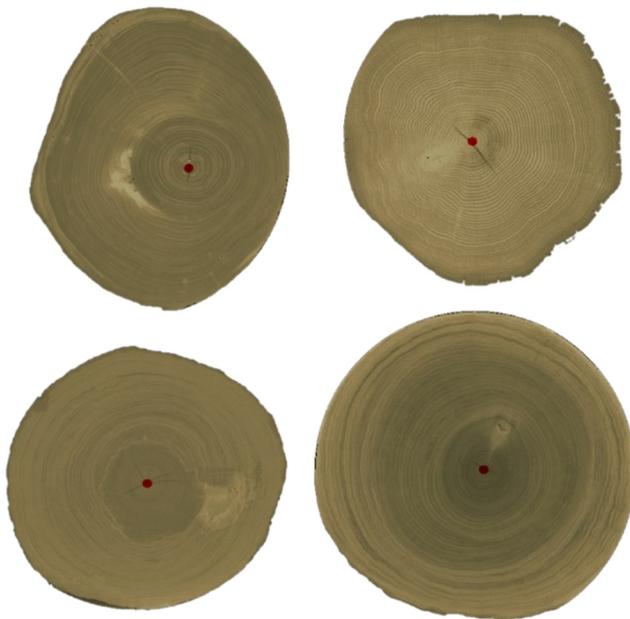


Fig. 9. Examples of detected pith (red dot) on various species (CT scans with false colors). Black Cherry (top left), Red Oak (top right), Hard Maple (bottom left), and Yellow Poplar (bottom right). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

both manual annotation and algorithmic solution was marked by a single pixel. In the context of hardwood lumber grading, the pith defect can have a diameter of approximately 10–20 pixels, thus when the algorithm-found pith center was, on average, 4.5 pixels away from the manually annotated center of the pith, it was empirically observed in majority of cases still within the pith defect area.

An example of comparison between the manual and the algorithm detection can be seen in Fig. 10 showing pith from the top and side view. As can be seen, most of the error is at the beginning of the log, where it is actually difficult to decide if the actual feature is a crack, pith, or an air pocket. However, since the logs are several meters long, this error is proportionally less important. In most of the internal log structure, the automated method provides reliable and precise results.

The precision varies between species. For example, Red Oak detection was less precise due to barely visible growth ring structure near the center at 512×512 resolution. In addition, the growth rings have similar colors and the color intensity has smaller difference. White Oak samples contained radial rays present on the cross-section, generating wrong gradients for Hough Transform. These samples also contained many slices with double pith that could lead in some cases to incorrect

detection of the primary pith (Fig. 11). The algorithm uses localized small windows that choose one of the two piths at the split point, the entire log would need to be taken into consideration to process such cases correctly. The other selected tree species showed better results due to more regular growth ring structure and the results were only affected by other natural defects.

Objective comparison with other solutions is problematic as we are using a different set of scanned logs. There are no two trees that are exactly the same. Additionally, there are anatomical differences between hardwoods and softwoods. Generally, hardwoods have higher density of wood, and a narrower pith. Difference in density of pith vs. surrounding wood in hardwoods is not as pronounced as in the softwoods, making the scanner setup and detection more challenging. However, compared to *PithExtract* from Boukadida et al. (2012) it appears that even though the precision of our solution is lower (4.18 mm compared to 1.69 mm), it is 17-times faster.

For reference, the early work of Norell and Borgefors (2008) reported the average processing time of 5,946.93 s and average error of 137.26 mm. Schraml and Uhl (2013) used two pith estimation methods with three configurations each to test CT images. The best algorithm/configuration combination resulted in mean accuracy of 4.76 mm and processing time of 16.64 s. From numerous algorithm/configuration combinations for rough log end digital images, the best reported method achieved the mean accuracy of 18 mm with processing time reported as “slow”. The most accurate of the three methods proposed by Kurdthongmee et al. (2018) had the mean error of 71.55 mm and processing time of 1.79 s. Their fastest method with processing time of 1.09 s has the mean error of 93.48 mm.

4. Conclusions & future work

Development of LogView™ real-time log processing optimization system requires rapid recognition and identification of various internal log features in CT scanned logs. This paper presents our efforts to speed up automatic identification of pith in hardwood logs. We have achieved this by developing GPU-accelerated automatic method that uses a two-pass algorithm. We verified our algorithm by comparing its results with manually detected pith from annotated images. We learned that our method satisfies the requirements of our industrial application and that it compares favorably to work done by others in terms of processing speed and accuracy.

While the presented algorithm meets the needs of LogView™ system very well, there are several avenues for future work. The method could be made faster by using a more advanced algorithm for the precise pith detection. It may be possible to use some pyramidal scheme with an approximation on low-resolution images. Moreover, although our results allow for an industrial application of our algorithm, it would be beneficial to verify our algorithm on a larger sample. Future investigation could also include the potential of the average gradient

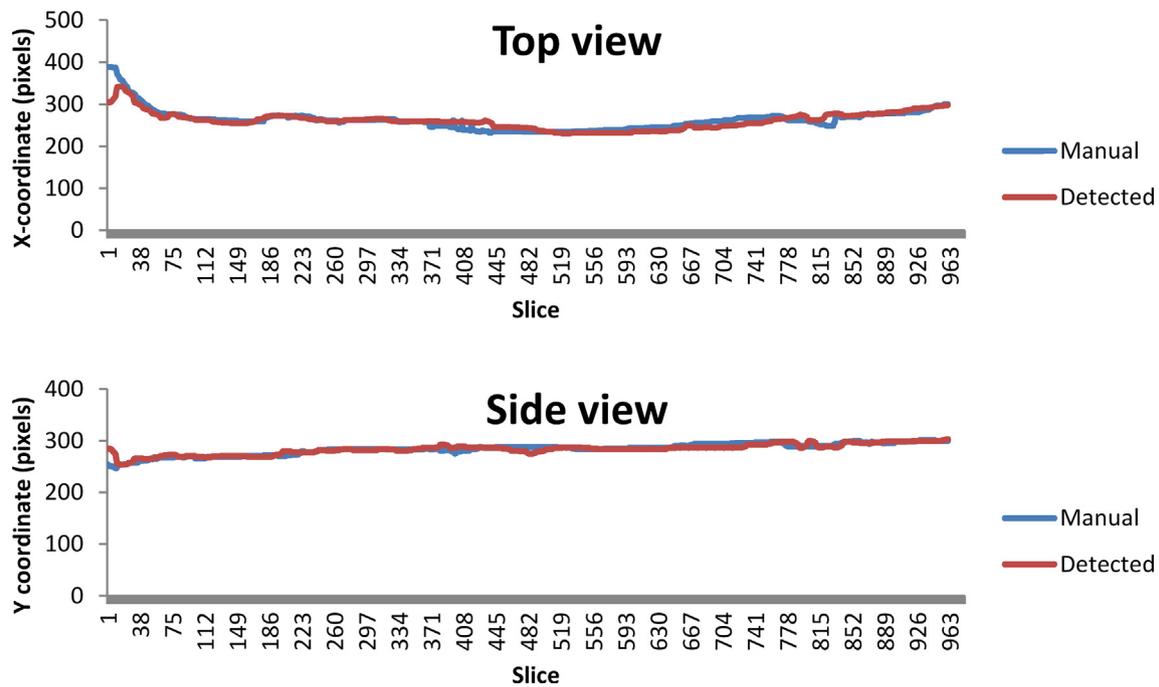


Fig. 10. Difference between manually annotated and automatically detected pith on an example log scan.

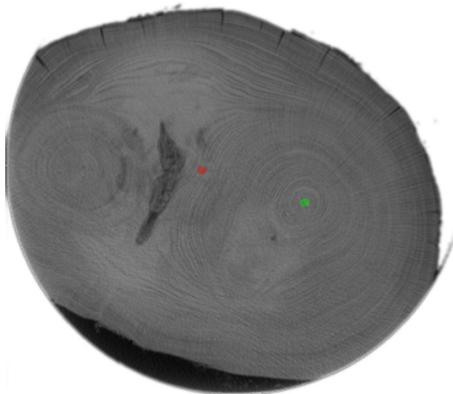


Fig. 11. Failure case with double pith; detector cannot decide which one to select.

strength of small areas of the image to predict pith location in pre-processing stage.

Funding
USDA Forest Service

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Appendix A. Supplementary material

Supplementary data to this article can be found online at <https://doi.org/10.1016/j.compag.2019.105107>.

References

Andreu, J.P., Rinnhofer, A., 2002. Enhancement of annual rings on industrial CT images of logs. *Comput. Electron. Agric.* 44, 107–119.

Bhandarkar, S., Faust, T., Tang, M., 1996. A system for detection of internal log defects by computer analysis of axial CT images. In: *Third IEEE Workshop on Applications of*

Computer Vision (WACV '96), pp. 258–263.

Bhandarkar, S., Faust, T., Tang, M., 1999. CATALOG: a system for detection and rendering of internal log defects using computed tomography. *Mach. Vis. Appl.* 11, 171–190.

Boukadida, H., Longuetaud, F., Colin, F., Constant, T., Mothe, F., 2012. PithExtract: A robust algorithm for pith detection in computed tomography images of wood – Application to 125 logs from 17 tree species. *Comput. Electron. Agric.* 85, 90–98.

Cerda, M., Hirschfeld-Kahler, N., Mery, D., 2007. Robust tree-ring detection. In: *IEEE Pacific Rim Symposium on Image Video and Technology – PSIVT. Santiago Chile*, pp. 575–585.

Chang, J., Gazo, R., 2009. Measuring the effect of internal log defect scanning on the value of lumber produced. *Forest Products J.* 59 (11/12), 56–59.

Gazo, R., Chang, J. (2010) Hardwood Log CT scanning – Proof of Concept. In: *Proceedings of Joint UNECE Timber Committee Session and Society of Wood Science and Technology International Convention: Innovative Wood Products are the Future*. United Nations, Geneva, Switzerland.

Gazo, R., J. Chang. 2012. Hardwood Log CT scanning. In: *Proceedings of the 2012 IUFRO D5 Conference Forest Products*. Lisbon, Portugal.

Gazo, R., Vanek, J., Abdul-Masih, M., B. Benes. 2014. CT scanning of logs – analysis and optimization for better utilization of hardwoods. In: *Proceedings of Society of Wood Science and Technology 2014 International Convention - Sustainable Resources and Technology for Forest Products*. Zvolen, Slovakia.

Gazo, R., Vanek, J., Abdul-Masih, M., Haviarova, E., and B. Benes. 2013. An Efficient Pith Detection for Computer Tomography Scanned Logs Using CUDA. *International IUFRO*. In: *Proceedings of MeMo Wood – Measurement methods and modeling approaches for predicting desirable future wood properties*. Nancy, France. Electronic.

Gazo, R., Benes, B., 2013. Computed tomography log scanning: an industrial application. In: *Proceedings of ISCHP 2013–4th International Scientific Conference on Hardwood Processing*, pp. 140–147.

Hough, P.V.C. Method and means for recognizing complex patterns, U.S. Patent 3,069,654, Dec. 18, 1962.

Kurdthongmee, W., Suwannarat, K., Panyuen, P., Sae-Ma, N. 2018. A Fast Algorithm to Approximate the Pith Location of Rubberwood Timber from a Normal Camera Image. In: *2018 15th International Joint Conference on Computer Science and Software Engineering (JCSSE)* (pp. 1–6). IEEE.

Longuetaud, F., Leban, J., Mothe, F., Kerrien, E., Berger, M., 2004. Automatic detection of pith on CT images of spruce logs. *Comput. Electron. Agric.* 44, 107–119.

Longuetaud, F., Caraglio, Y. 2009. Pith: a marker of primary growth in *Picea abies* (L.) Karst. *Trees – Structure and Function*, 23, 325–334.

Canny, J., 1986. A (1986) computational approach to edge detection. *IEEE Trans. Pattern Anal. Mach. Intell.* 8 (6), 679–698.

National Hardwood Lumber Association, 2019. *Rules for the measurement and inspection of hardwood and cypress*. NHLA, Memphis, TN.

Norell, K., Borgefors, G., 2008. Estimation of pith position in untreated log ends in sawmill environments. *Comput. Electron. Agric.* 63, 155–167.

nVIDIA Corporation (2014) *CUDA Toolkit Documentation* <http://docs.nvidia.com/cuda/index.html>.

Saint-André, L., Leban, J., 2001. A model for the position and ring eccentricity in transverse sections of Norway spruce logs. *Eur. J. Wood Wood Prod.* 59, 137–144.

Sepúlveda, P., 2001. Measurement of spiral grain with computed tomography. *J. Wood Sci.* 47, 289–293.

Schraml, R., Uhl, A. 2013. Pith estimation on rough log end images using local Fourier spectrum analysis. In: *Proceedings of the 14th Conference on Computer Graphics and Imaging (CGIM'13)*, Innsbruck, AUT.

Zhu, D., Connors, R., Schmoltd, D., Araman, P., 1996. A prototype vision system for analyzing CT imagery of hardwood logs. *IEEE Trans. Syst. Man Cybern. – Part B: Cybern.* 26, 522–532.