Automatic Wood Pith Detector: Local Orientation Estimation and Robust Accumulation

Henry Marichal[1], Diego Passarella[2], and Gregory Randall[1]

[1] Instituto de Ingeniería Eléctrica, Facultad de Ingeniería, Universidad de la República, Montevideo, Uruguay
[2] Centro Universitario Regional Noreste, Universidad de la República, Tacuarembo, Uruguay

Abstract. A fully automated technique for wood pith detection (APD), relying on the concentric shape of the structure of wood ring slices, is introduced. The method estimates the ring's local orientations using the 2D structure tensor and finds the pith position, optimizing a cost function designed for this problem. We also present a variant (APD-PCL), using the parallel coordinates space, that enhances the method's effectiveness when there are no clear tree ring patterns. Furthermore, refining previous work by Kurdthongmee, a YoloV8 net is trained for pith detection, producing a deep learning-based approach to the same problem (APD-DL). All methods were tested on seven datasets, including images captured under diverse conditions (controlled laboratory settings, sawmill, and forest) and featuring various tree species (*Pinus taeda, Douglas fir, Abies alba*, and *Gleditsia triacanthos*). All proposed approaches outperform existing state-of-the-art methods and can be used in CPU-based real-time applications. Additionally, we provide a novel dataset comprising images of gymnosperm and angiosperm species. Dataset and source code are available at `http://github.com/hmarichal93/apd`.

Keywords: Computer vision · Wood pith detection · Deep neural network object detection · Wood quality

1 Introduction

Locating the pith of tree cross-sections is essential to identify (in basal discs) the first year of growth and, therefore, the tree's age. The pith has a different type of tissue than the rest of the tree, with distinct physical-mechanical properties. Locating the pith is useful, among other reasons, to detect growing eccentricity; because in the natural process of senescence of standing trees, the fungi that degrade the wood enter through the pith or because the industry discards that part as it has different uses than the rest of the wood. Moreover, some tree ring delineation algorithms are sensitive to a precise pith location Cerda et al. [2007], Makela et al. [2020], Marichal et al. [2023b], Norell [2009], mainly when those algorithms are based on the ring structure, a concentric pattern similar to a spider web as illustrated in Figure 1. That figure shows some examples of the diversity of images of tree slices. Ideally, the intersection point between the perpendicular lines through the tree rings should be the pith (the center of the structure, located inside the medulla of the tree). The *spider web* model is only a general

approximation, as it is depicted in Figure 1.d. Real slices include ring asymmetries, cracks, knots, fungus, etc., as seen in Figure 1.b and Figure 1.c. Different species produce diverse patterns. Moreover, gymnosperm, as the ones illustrated in Figure 1, and angiosperm species produce a different wood structure, as seen in Figure 6.e. Automatic pith detection must be robust to such variations and perturbations.
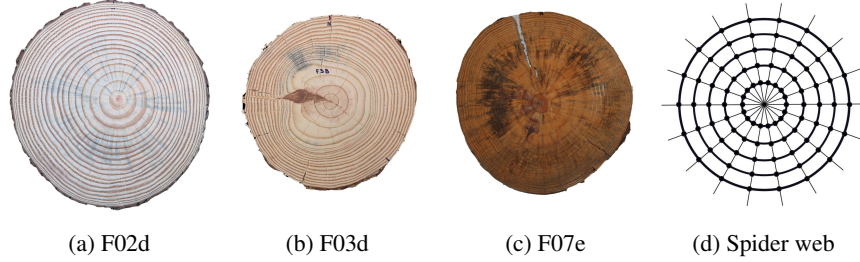

| (a) F02d | (b) F03d | (c) F07e | (d) Spider web |

Fig. 1: (a) to (c) Some examples from **UruDendro** datasetMarichal et al. [2023a], (d) The whole structure, called *spider web*, is formed by a *center* (the slice pith), *rays*, and the *rings* (concentric curves). In the scheme, the *rings* are circles, but in practice, they can be (strongly) deformed as long as they don't intersect another *ring*.

This paper presents several key contributions: the release of a new challenging dataset (UruDendro2 and UruDendro3) for wood pith detection, the development of real-time automatic detection methods (APD and APD-PCL), training of a YoloV8 net (APD-DL) for the same purpose, and rigorous comparison with state-of-the-art methods on various public datasets. These contributions enhance the field of wood pith detection, offering practical solutions and insights for real-time applications.

2   Previous work

Schraml and Uhl et al. Schraml and Uhl [2013] proposed a method (here called LFSA) that splits the wood cross-section into patches, estimating the patch's orientation by 2D Fourier Transform. They accumulate the patch's orientation using a Hough Transform approach and estimate the pith position as the maximum in the accumulation space.

Kurdthongmee et al. Kurdthongmee et al. [2018] proposed the Histogram Orientation Gradient to estimate the tree ring local orientation and proceed similarly to Schraml and Uhl [2013]. In the same line, Norell et al. Norell and Borgefors [2008] proposed two ways for estimating the local orientations: quadrature filters and a Laplacian pyramids approach. Recently, Decelle et al. Decelle et al. [2022] proposed ACO, a method based on an ant colony optimization algorithm for the local orientation accumulation step.

Deep Neural Network (DNN) methods have also been applied to solve this problem. Kurdhongmeed et al. Kurdthongmee [2020] compared the effectiveness of two DNN

object detector models (YoloV3 and SSD MobileNet) to locate the pith. They trained the models via transfer learning over 345 wood slice RGB images captured within a sawmill environment and evaluated over a separate dataset of 215 images.

3 APD: Automatic Wood Pith Detection

We propose an automatic pith detection method based on a model of the wood slice. In a gymnosperm tree cross-section, as the ones shown in Figure 1, two types of structures are present: the rings formed by (roughly) concentric curves and (in some cases) the presence of radial structures such as cracks and fungi. Both are fundamentally related to the pith. The former is due to the growing process of the tree, which forms the rings, and the latter is because the tree's anatomy leads naturally to the radial characteristic of cracks and fungus growing. From this observation derives the principal idea of the proposed method: giving an image of a tree cross-section, we can locate the pith at the intersection of the lines supported by radial structures and the perpendiculars to the rings.

The angiosperm tree cross-section structure is slightly different, as seen in Figure 6e. Still, it is also formed of radially organized cells, with texture patterns that appear at different radii of the pith. This produces visual macrostructures that allow a similar approach to determine the pith position as depicted in the previous paragraph.

Not always do those hypotheses stand out completely. Sometimes, the ring structure can be highly (locally) deformed, as in the presence of a knot. Sometimes, there are no cracks or fungi present. But in general, enough information is produced by the ring structure and, eventually, by the presence of cracks and fungi to estimate the pith location correctly.

Given an image of the tree cross-section, and using the *spider web* model illustrated in Figure 1.d, the APD approach pseudocode is described at Algorithm 1. The main steps are the following (see Figure 3 for more details):

1. *Local orientation detection (line 1 of Algorithm 1)*. To estimate the local orientation (LO), we compute the 2D-Structure Tensor Bigun et al. [1991] $ST[p]$ at each pixel p using a window of size $st_w \times st_w$. Pixels in the window are weighted by a Gaussian kernel w of parameter st_σ. The structure tensor is calculated as $ST[p] = \sum_r w[r]ST_{xy}[p-r]$ where $ST_{xy}[p]$ is defined as

$$ST_{xy}[p] = \begin{bmatrix} (I_x[p])^2 & I_x[p]I_y[p] \\ I_x[p]I_y[p] & (I_y[p])^2 \end{bmatrix}$$

where $I_x[p]$ and $I_y[p]$ are the first derivatives of image I in point p along x and y, respectively. For simplification, we can re-write the 2×2 structure tensor matrix at pixel p as:

$$ST[p] = \begin{bmatrix} J_{11} & J_{12} \\ J_{12} & J_{22} \end{bmatrix}$$

The local orientation at pixel p is:

$$ST_O[p] = \frac{1}{2}\arctan(\frac{2J_{12}}{J_{22} - J_{11}}) \tag{1}$$

Algorithm 1: APD

Input: Im_{in}, //RGB slice image;
Output: Pith location

1 $ST_O, ST_C \leftarrow$ local_orientation(Im_{in}, st_σ, st_w)

2 $LO_f \leftarrow$ lo_sampling(ST_O, ST_C, lo_w, $percent_{LO}$)

3 $LO_r \leftarrow LO_f$

4 **for** i **in** 1 **to** max_iter **do**

5 $\quad$ **if** $i > 1$ **then**

6 $\quad\quad$ $LO_r \leftarrow$ filter_lo_around_c_i(LO_f, r_f, c_i), // See Figure 3.e

7 $\quad$ $c_{i+1} \leftarrow$ optimization(LO_r) // Equation (4)

8 $\quad$ **if** $\|c_{i+1} - c_i\|_2 < \epsilon$ **then**

9 $\quad\quad$ $c_i \leftarrow c_{i+1}$

10 $\quad\quad$ break

11 $\quad$ $c_i \leftarrow c_{i+1}$

12 **return** c_i

The *coherence* of the LO estimation in p is given by the relative value of eigenvalues λ_1 and λ_2 (where λ_1 is the largest eigenvalue, and λ_2 is the smallest one):

$$ST_C[p] = \left(\frac{\lambda_1 - \lambda_2}{\lambda_1 + \lambda_2}\right)^2 \tag{2}$$

The outputs of this step are two matrices: one of local orientations (ST_O) and one of coherence (ST_C).

2. *Local orientation sampling (line 2 of Algorithm 1)*. The LO estimations are sampled in the following way: 1) ST_O and ST_C are divided in non-overlapping patches of size $lo_w \times lo_w$. 2) We find the pixel p^j with the highest coherence (c_{high}^j) within patch $patch_i$. A minimum patch coherence st_{th} is defined. We assign $ST_O[p^j]$ to $patch_i$ in position p^j, if $c_{high}^j > st_{th}$. To fix st_{th}, we calculate the value of ST_C such that a given percentage (parameter $percent_{LO}$) of the LO in the slice has $ST_C > st_{th}$. Each LO is a segment $lo_i = \overline{p_1^i p_2^i}$, defined by the limits p_1^i and p_2^i. p_{LO}^i is the middle point between them (p^j). Given the local orientation $\alpha_i = ST_O[p_{LO}^i]$, points p_1^i and p_2^i are computed as $p_{1,2}^i = p_{LO}^i \pm (\cos(\alpha_i), \sin(\alpha_i))$.
Suppose N patches have coherently enough LO; the output of the step is a matrix, LO_f of size $N \times 4$. In this way, lines are supported by the LO of all meaningful structures in the cross-section, such as the rings.

3. *Find the center (line 7 of Algorithm 1)*. Given the filtered local orientation matrix, LO_f, we define the following optimization problem: find c_{opt}, the geometrical position that maximizes the collinearity between the lo_i and a line passing by c_{opt} and p_{LO}^i. To this aim, we define the following cost function:

$$h(x, y) = \frac{1}{N} \sum_{i=1}^{N} \cos^2(\theta_i(x, y)) \tag{3}$$

Figure 2 illustrates the vectors involved in computing Equation (3). The angle between $\overline{p_1^i p_2^i}$ and $\overline{cp_{LO}^i}$ is θ_i. The pith position c of coordinates (x,y) is the origin of a segment $\overline{cp_c^i}$. As $\cos(\theta_i(x,y)) = \frac{<\overline{cp_{LO}^i}, \overline{p_1^i p_2^i}>}{|cp_{LO}^i||p_1^i p_2^i|}$, the optimization problem to be solved becomes:

$$c_{opt} = \max_c \quad \frac{1}{N} \sum_{i=1}^{N} \left(\frac{< \overline{cp_{LO}^i}, \overline{p_1^i p_2^i} >}{|cp_{LO}^i||p_1^i p_2^i|} \right)^2 \tag{4}$$

$$\text{s.t.} \quad c \in Slice\ Region$$

4. To find the global maximum (c_{opt}) of the former optimization problem, we use the *minimize* method from *scipy.optimize* python package with its default parameters. Problem 4 is convex if it is restricted to the region of the wood cross-section. Therefore, as initialization to the *minimize* method, we use the least squares solution of finding the point c_{ini}, which minimizes the distance to all the lines in LO_r within the slice.
5. *Refinement (lines 4 to 11 of Algorithm 1)*. Once a candidate for the pith location c_{opt} is obtained, the optimization procedure (4) is repeated using only the local orientations within a squared region of size $Size_{image}/r_f$ centered in c_{opt} (see Figure 3.e). This step is repeated until the pith location doesn't move more than a given tolerance ($\epsilon = 10^{-5}$) or the iteration counter reaches *max_iter* = 5. This approach avoids distortions introduced by a (very) asymmetric tree ring growth pattern.
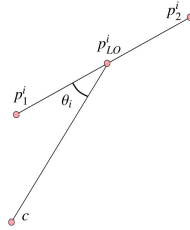

Fig. 2: Cost function definitions

4   APD-PCL: PClines based Automatic Wood Pith Detection

The APD method described in Section 3 works fine when the ring structure gives enough information. In some (rare) cases, the ring structure is not visible due to fungi or other perturbations. In those cases, it is possible to solve the same problem using the lines supported by the radial structure of those perturbations in addition to the lines produced by the ring structure. For this reason, the APD-PCL version of the method is more
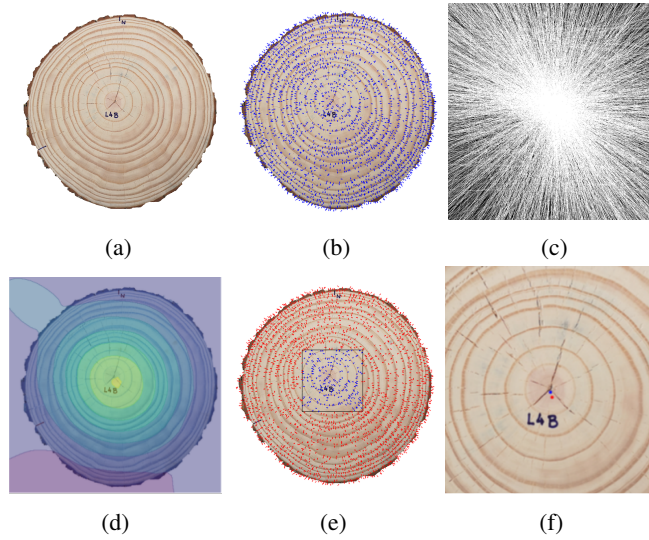
Fig. 3: Principal steps of APD method (L04d image from UruDendro2 collection). (a) Resized slice image, without background; (b) Sampled LO produced by the Structure Tensor estimation; (c) Accumulation space defined by the LO supported lines; (d) Plot of the cost function (Equation (3)), highest values in yellow; (e) Sub image built around the solution c_1 obtained after the first iteration; (f) Evolution of c_i. The final solution is in blue; previous iterations' solutions are in red.

robust and allows for the successful treatment of cross-sections with highly degraded ring patterns. The price to pay is a slower algorithm, as it includes a RANSAC-based clustering step.

The APD-PCL method selects which local orientations to consider in the optimization problem of Equation (4). In general, the estimation made by the structure tensor calculation step is determined by the rings. Different perturbations also produce some LO, but its number is minimal, and the lines they support don't converge to the pith. In some (rare) cases, the perturbations are so important that they overshadow the ring structure. In those cases, the number of LO produced by the perturbations is more significant than those produced by the ring structure. The set of perturbations-related local orientations can be of diverse origin: knots, fungi, cracks, and noise. Some of them (namely fungi and cracks) have a typical radial orientation, so the perpendicular lines to its LO converge to the pith.

Considering this, we modify Algorithm 1, by including a post-processing step over matrix LO_f, between lines 2 and 3. The rest of the algorithm is the same:

1. Use the PClines transform Dubska et al. [2011] to convert each line into a point.
2. The PClines space is formed by two sub-spaces defined by a parameter d: the *straight space* includes lines with orientations $\alpha_i \in [0, \frac{\pi}{2}]$ and the *twisted space* lines with orientations $\alpha_i \in [\frac{\pi}{2}, \pi]$. As seen in Figure 4, convergent lines in the Eu-

clidean space correspond to aligned points in the PClines spaces. This allows the following steps:

(a) Lines supported by LO produced by the ring structure converge somewhere around the pith. They produce a line-shaped cluster in the PClines spaces (figures 4b and 4c). We select the aligned points using a RANSAC Fischler and Bolles [1981] approach. We work only in the $[-d, 0]$ and $[0, d]$ ranges for the twisted and straight sub-spaces, respectively. This avoids the use of points near the infinity. We select all the converging lines in both spaces, excluding those simultaneously selected in both sub-spaces.

(b) The previous step clusters all convergent LO_f in the image producing the set LO_{ring}. We rotate by 90 degrees all the orientations in LO_f and repeat the previous procedure to detect the converging ones. These rotated converging lines cluster, LO_{radial}, is produced by cracks, fungi, or similar structures. Adding both gives the set:

$$LO_f^{PClines} = LO_{ring} + LO_{radial}$$

.

(c) To make the line segment selection method more robust, we add a third PClines transform using the lines supported by $LO_f^{PClines}$. Most ring-related LO and rotated LO generated by radial structures are expected to converge (hence, to form a line cluster in the PCline space). Therefore, most of the outliers should be removed at this step.

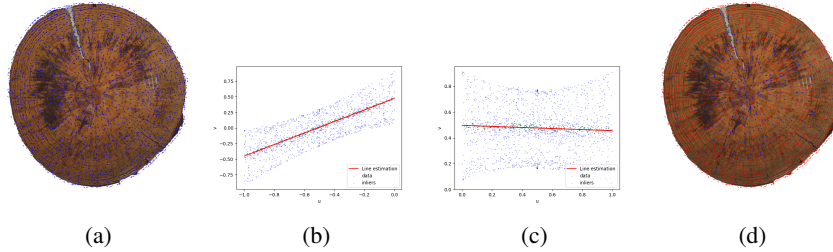(a)   (b)   (c)   (d)

Fig. 4: Use of PClines to cluster converging local orientations for slice F07e (same as Figure 1c). (a) Local orientations; (b) Selection of the converging segments in the twisted space using RANSAC to fit a line (in red). Inliers are colored in green; (c) The same procedure is applied in the straight space; (d) In blue, the converging LO (inliers from both sub-spaces) and the LO to be removed in red.

Figure 5 illustrates the considered lines and the accumulation space of Equation (3) without and with the PClines step. Note how the method filters out many non-convergent lines and regularizes the cost function.

The APD-PCL method is similar to the APD one, but the PClines-based filtering step diminishes the number of considered lines, filtering out many non-convergent ones.
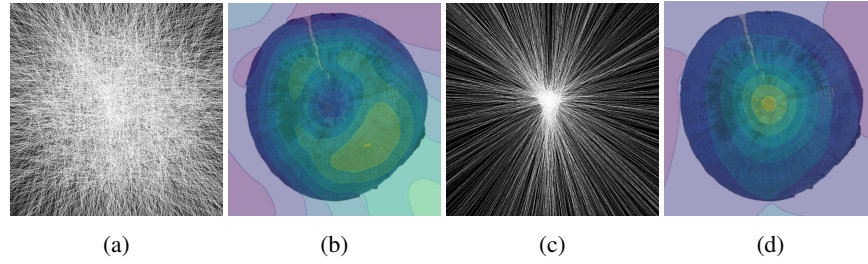
Fig. 5: LO Accumulation space and cost function for slice F07e with and without applying the PClines filtering method. (a) LO Accumulation space with no filtering; (b) cost function of LO with no filtering; (c) LO Accumulation Space with PClines filtering; (d) cost function of LO with PClines filtering

5   APD-DL: Deep Learning based Automatic Wood Pith Detection

In Sections 3 and 4, we tackle the pith detection problem using a (*spider web*) model, as in the "classic" image processing times. Now, we present a Deep-learning approach that learns the model from the data. To this aim, Kurdthongmee et al. Kurdthongmee and Suwannarat [2019] used a YoloV3 model. Inspired by them, we train a YoloV8 Jocher et al. [2023] network using the datasets described in Section 6. This is an architecture tailored for object detection and segmentation. To train it, we must provide a set of data formed by wood cross-section images labeled with the ground truth position of the pith as a bounding box.

We use a five-fold cross-validation technique. We divide the data into five sets. In each fold i, we use one set ($test_i$) for testing and the other four for training. The training process in each fold is done as usual, and we use the produced model to label the data in $test_i$. The process is repeated for all the folds. In the end, we have predictions for all the data, and in each case, the used model was generated without the influence of the $test_i$ data. With the predictions for all images produced in this manner, we can deliver the metrics to determine the method's performance.

Table 1 show the results using normalized errors (see Section 7.2). Training with such a high diversity of data produces state-of-the-art results. Results over each row (collection) are calculated using the predictions produced during the five-fold cross-validation with all the images in the seven datasets. In some (rare) cases, this approach doesn't give a prediction (hence a false negative). In those situations, the method gives the center of the image as the pith position. This explains the (relatively) large value of the Maximum error and the differences between the Mean and Median errors. Besides the rare false negatives, the results are excellent. The last row depicts the results for all the collections.

Hyperparameters The algorithm was trained with images of width size 640 (keeping the aspect ratio), using a batch size of 16, for 100 epochs, with the optimizer AdamW

Collection	Mean (Std)	Median	Max	FN
Uru2	0.55 (1.45)	0.18	11.32	2
Uru3	0.13 (0.06)	0.13	0.27	0
Kennel	0.14 (0.07)	0.13	0.24	0
Forest	0.45 (1.85)	0.12	13.91	0
Logyard	0.52 (1.29)	0.27	7.51	0
Logs	0.22 (0.46)	0.13	4.42	1
Discs	0.23 (0.54)	0.14	5.67	0
All	0.33 (1.01)	0.14	13.91	3

Table 1: Prediction results of 5-fold cross-validation for the APD-DL. The second to fourth columns show the Mean (and standard deviation in parenthesis), Median, and Maximum normalized error (defined in Section 7.2) values. The last column shows the false negatives. We use all the datasets together for train the model and calculate the performance within each dataset.

Loshchilov and Hutter [2019] ($lr = 0.002$, $momentum = 0.9$) and yolov8n as pre-trained weights. All the network was re-trained.

6 Datasets description

We use the following datasets:

- UruDendro. We introduce here a new public dataset with two collections of wood cross-section samples with experts annotated ground truth Marichal et al. [2023a]:
    - UruDendro2: 119 RGB images of *Pinus taeda* slices. This collection includes 64 images taken under different illumination conditions and cameras, published in 2022 in our website Marichal et al. [2023a], increased with 55 new images. The new images were taken in laboratory conditions, with an iPhone 6S phone (12 Mpx camera) at a distance between 43 and 51 cm from the slice, under controlled illumination with a led ring of 35 W. Size images range between 1000 and 3000 pixels in width.The surface of the slices presented different conditions: some were cut by chainsaw, smoothed by a handheld planner, and polished with a rotary sander. All these images are annotated by at least one expert with the position of the pith.
    - UruDendro3: 9 RGB images of *Gleditsia triacanthos*, an angiosperm, acquired in a laboratory, without illumination-controlled conditions, using a Huawei P20 Pro smartphone (24 Mpx camera) at a distance of approximately 1 meter from the slice. Size images range between 1000 and 2000 pixels in width. All the slices were polished. All these images are annotated by at least one expert with the position of the pith.
- Kennel Kennel et al. [2015]. A public dataset with 7 RGB 1280 pixels squared images of *Abies alba*, polished and acquired in controlled illumination laboratory conditions. The pith pixel location is provided as metadata.
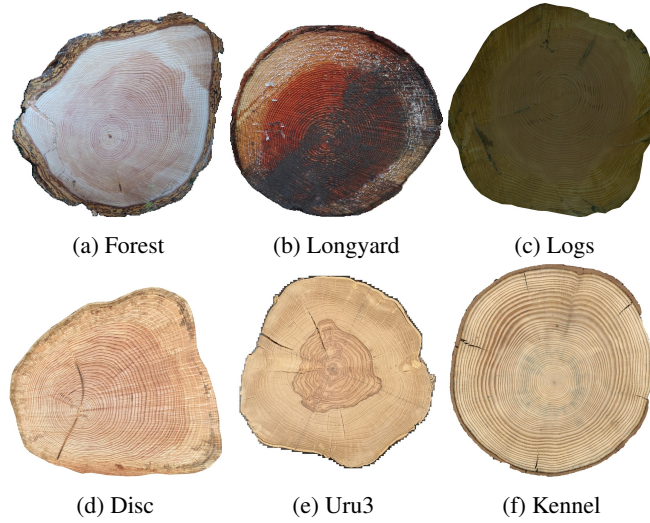
(a) Forest   (b) Longyard   (c) Logs

(d) Disc   (e) Uru3   (f) Kennel

Fig. 6: Examples of the used datasets. Species are (a-d) *Douglas fir*, (e) *Gleditsia tri-acanthos*, (f) *Abies alba*. Acquisition conditions: (a-c) in the field, with a smartphone camera; (d-e) in the laboratory, with controlled illumination. The samples of images (d-e) were previously sanded and polished. The samples of images (a-c) didn't have any special treatment.

– TreeTrace Longuetaud et al. [2022]. A public dataset, with samples of *Douglas fir* taken at different stages of the wood process chain. The pith pixel location is provided as metadata. Each image has several wood slices. To build the collections, we extract sub-images containing one slice each, producing almost squared images between 1000 and 3000 pixels in width. This dataset includes the following collections:
  • Forest, 57 RGB images taken from the freshly cut logs with a digital camera.
  • Logyard, 32 RGB images of the same log ends, acquired with a smartphone in the sawmill courtyard several days after the cutting.
  • Logs, 150 RGB images acquired in the sawmill with a smartphone.
  • Discs, 208 RGB images acquired with a 400 dpi scanner from sanded and polished slices after several weeks of air-drying.

Table 2 summarize the used datasets. Figure 1 show images from the UruDendro2 dataset, and Figure 6 show examples from the other collections. These datasets convey a high degree of variability. It includes examples of gymnosperm (*Pinus taeda*, *Abies alba* and *Douglas fir*) as well as angiosperm (*Gleditsia triacanthos*). Acquisition conditions are also diverse, including images obtained with a smartphone in the forest or the sawmill. Samples were acquired in the field with dirt, sap, or saw marks, and others were obtained in controlled illumination conditions in the laboratory from polished samples. The samples include perturbations as the presence of fungi, cracks, and knots,

Collection	Size	Specie
UruDendro2	119	*Pinus taeda*
UruDendro3	9	*Gleditsia triacanthos*
Kennel	7	*Abies alba*
Forest	57	*Douglas fir*
Logyard	32	*Douglas fir*
Logs	150	*Douglas fir*
Discs	208	*Douglas fir*

Table 2: Dataset description.

as can be seen in Figure 1 and sap and saw marks, as can be seen in Figure 6.b and Figure 6.d. All have the ground truth position of the pith. Considering all datasets, we work with 582 images.

7   Results and discussion

	UruDendro2	UruDendro3	Kennel	Forest	Logyard	Logs	Discs
LFSA Schraml and Uhl [2013]	1.03 (0.85)	1.46 (0.97)	0.42 (0.18)	0.80 (0.36)	1.02 (0.62)	0.80 (0.46)	0.72 (0.43)
ACO Decelle et al. [2022]	2.23 (6.64)	4.52 (11.96)	0.2 (0.06)	0.24 (0.24)	0.60 (1.11)	0.46 (0.45)	0.24 (0.35)
APD-PCL	**0.42 (0.34)**	0.74 (0.54)	0.19 (0.10)	0.81 (0.98)	0.82 (0.84)	0.52 (0.47)	0.46 (0.57)
APD	1.02 (2.45)	0.55 (0.30)	**0.14 (0.06)**	**0.22 (0.18)**	**0.35 (0.17)**	0.29 (0.33)	0.26 (0.42)
APD-DL	0.55 (1.45)	**0.13 (0.06)**	0.14 (0.07)	0.45 (1.85)	0.52 (1.29)	**0.22 (0.46)**	**0.23 (0.54)**

Table 3: Results on all the datasets. Normalized errors. We show the mean error and the standard deviation between parenthesis.

7.1   Preprocessing

Sometimes, the images are acquired in the field with a smartphone camera, and one image can contain more than one cross-section. Regardless of the method used, all images are preprocessed in such a way as to standardize the image input:

1. *Background substraction*. Produce a new image limited to one slice. To this aim, when possible, we filter out the background using the mask provided in the datasets. If the mask is not provided, we use a deep learning-based method Qin et al. [2020], which uses an U^2Net to segment salient objects.
2. *Resize the image*. This step, not strictly necessary, allows us to fix the algorithm's parameters once and for all. All images are resized to 640 pixels width, respecting the original image's aspect ratio.

7.2   Normalized errors

Given the cross-sections' diverse dimensions, presenting the errors in pixels is not informative. Additionally, not all the datasets provides the millimeter pixel relation. We use the percentage of the equivalent slice radius. Given a prediction P_i and a Ground Truth GT_i, this error is calculated as follows:

$$Err_i = \frac{100 \times Dist(P_i, GT_i)}{Equivalent\_radio(image_i)}$$

Where $Dist(P_i, GT_i)$ is the Euclidean distance, in pixels, between the prediction and the Ground Truth. Remember that within this work, the pith is modeled as a point in the image. Therefore $Dist(P_i, GT_i)$ is a distance between points. $Equivalent\_radio(image_i)$ (in pixels) is half the biggest horizontal or vertical image size that circumscribes the slice, without background, generated by the preprocessing.

7.3   Experiments

The method to fine-tune the APD-DL method is explained in Section 5. To determine the best parameters' values for LFSA, APD and APD-PCL we minimize overall used datasets the average of Euclidean distances between ground truth and predictions.

For the APD and APD-PCL methods, the parameters st_σ, r_f and $ransac\_outlier\_th$ were set after experiments over a few images. Therefore, they are considered as fixed. The rest of the parameters, $percent_{LO}$, st_w and lo_w were set following the procedure described in the former paragraph. The search of the minimum was over the following grid: $percent_{LO}$ in $[0.3, 0.5, 0.7, 0.9]$, st_w in $[3, 7, 9, 11]$ and lo_w in $[3, 7, 9, 11]$.

Inferences were made using an Intel Core i5 10300H workstation with 16GB and a GPU GTX1650 (when needed).

7.4   Results

In this section, a performance comparison is made between the different methods. Table 3 show the performance of the proposed methods and two state-of-the-art ones Decelle et al. [2022], Schraml and Uhl [2013], over the datasets presented in Section 6. We use the mean error and standard deviation, using normalized errors, to compare different-size wood cross-sections. The performance of the methods differs for each collection due to its specific characteristics regarding species, acquisition conditions, etc. Note that ACO was developed (and tailored) for the TraceTree collections. Its performance degrades when tried on other species (such as UruDendro collections). LFSA performance is more regular across collections. The three methods proposed in this paper outperform ACO and LFSA on all collections. APD and APD-DL perform better for almost all collections. APD outperforms APD-PCL for all collections except UruDendro2, which has some images with fungi and cracks overshadowing the ring structure. Note that in all the cases, the precision of the pith detection is very high.

Table 4 compares the performance of all tested methods using the 582 images of all collections. All methods presented in this paper outperform LSFA and ACO methods. The APD performance is surpassed only by the APD-DL method but at the cost of some

Method	Mean	Median	Max	FN	Time
LFSA Schraml and Uhl [2013]	0.83	0.72	5.03	0	627
ACO Decelle et al. [2022]	0.79	0.21	36.39	2	918
APD-PCL	0.52	0.34	4.33	0	2339
APD	0.42	0.19	15.44	0	784
APD-DL	0.33	0.14	13.91	3	209

Table 4: Results of all the methods over the whole set of images, i.e., merging all collections. Normalized errors, number of false negatives, and execution time in milliseconds.

false negatives: images in which APD-DL didn't find a solution. We can see that APD slightly outperforms the APD-PCL method. This is due to the RANSAC algorithm used to cluster points in the PClines space. When there is no clear clustering of points around a line, RANSAC tries to fit a line anyway, selecting a wrong set of LO and producing a wrong pith localization. This situation sometimes appears in the TreeTrace dataset. To consider the mean processing time per image for each method, it must be considered that APD-DL and ACO methods run on GPU, while APD, APD-DL, and LFSA run on a CPU machine. Note that APD is roughly three times faster than APD-PCL. All in all, it is remarkable that the "classic style" model-based proposed methods (APD and APD-PCL) and a Deep Learning one (APD-DL) have similar performance and execution times, allowing real-time applications with a CPU in the APD and APD-PCL cases. In the supplementary material, we add showcases illustrating how the different methods work under extreme conditions.

8   Conclusions and future work

This paper addresses the wood pith detection on tree slices problem using classic image processing and machine learning-based approaches. Both approaches are determined by the characteristics of the data used to tune the algorithm. In search of a more general solution, we use a set of diverse datasets, which spans different species, acquisition conditions, and perturbations (from cracks and knots to saw marks and dirt for images acquired on the field).

We proposed three real-time methods. The first two are based on a *spider web* model in a classic image processing approach, and the third one is a Deep Learning method. The former has excellent performance, runs in real-time on a CPU-based machine, and the model allows a clear comprehension of the approach. The limited number of parameters is understandable and can be fixed once and for all. The latter has better (although similar) performance but has some false negatives and is more opaque concerning the meaning of its millions of parameters. Moreover, it runs on a GPU based machine.

The UruDendro dataset, with annotated images of *Pinus taeda* (a gymnosperm) and *Gleditsia triacanthos* (an angiosperm), are presented and can be used by the community to test other approaches to this problem.

9   Supplementary material

9.1   APD Parameters

The number of parameters in the machine learning approaches is huge, and they don't have a clear physical meaning. In the classic approaches, such as APD and APD-PCL methods, a limited number of parameters are included, and they have a physical or algorithmic meaning. The APD method has the following parameters, fixed once and for all after a grid search with all the datasets. The default values are in parentheses:

- st_σ: Structure tensor Gaussian σ (1.2).
- st_w: Structure tensor Gaussian kernel size (3 for APD and 7 for APD-PCL).
- $percent_{LO}$: to fix st_{th}, the minimum coherence value to consider valid an LO (0.7).
- lo_w: LO sampling window size (3 for APD and 7 for APD-PCL).
- r_f: a factor used to estimate the side size of the search region for the iteration (7).

The APD-PCL method adds the following:

- $ransac\_outlier\_th$: RANSAC residual threshold defining the width of the line cluster (0.03).

9.2   Methods comparison: difficult cases

Figure 7 illustrates how all the presented methods perform over the difficult cases in the UruDendro2 collection. Figure 7.a to 7.c illustrates cases where the ACO method (red marker) performs poorly. In one case, the method even predicts outside the disc region (Figure 7.a). In this cases, the APD, APD-PCL and APD-DL (blue, yellow, and green markers, respectively) perform similarly, all of them close enough to the ground truth pith (the prediction is within the first tree ring region). On the other hand, the LFSA method (purple marker) performs slightly worse.

Figure 7.d to 7.f show discs with a strong fungus presence around the pith position. In this case, the APD-PCL (yellow marker) performs considerably better than the other methods (in all cases, the prediction is within the first tree ring region). When there is no tree ring information, the tree ring-based methods (LFSA, ACO and APD) fail and do not converge near the pith position as illustrated in Figure 7.e and 7.f. This is the critical scenario for the ACO and APD methods. In the last stages, they search for the local minimum in a region centered around the previous pith prediction. The ring information will be very limited if the region is too small, which produces a method divergence. In the other hand, the APD-PCL method integrates radial structural insights derived from cracks and fungus in addition to the ring structure. Thereby enhancing its performance significantly in this particular scenario. Finally, the APD-DL method (green marker) also performs poorly in this scenario.

Figure 8 illustrates cases of off-center pith in Forest, Logyard and Disc collections and the presence of (strong) saw marks. As seen in Figure 8.a to 8.c, the APD-DL method performs considerably worse than the other methods. These disks belong to the Forest and Logyard collections. Both collections have 89 images, a small data size for
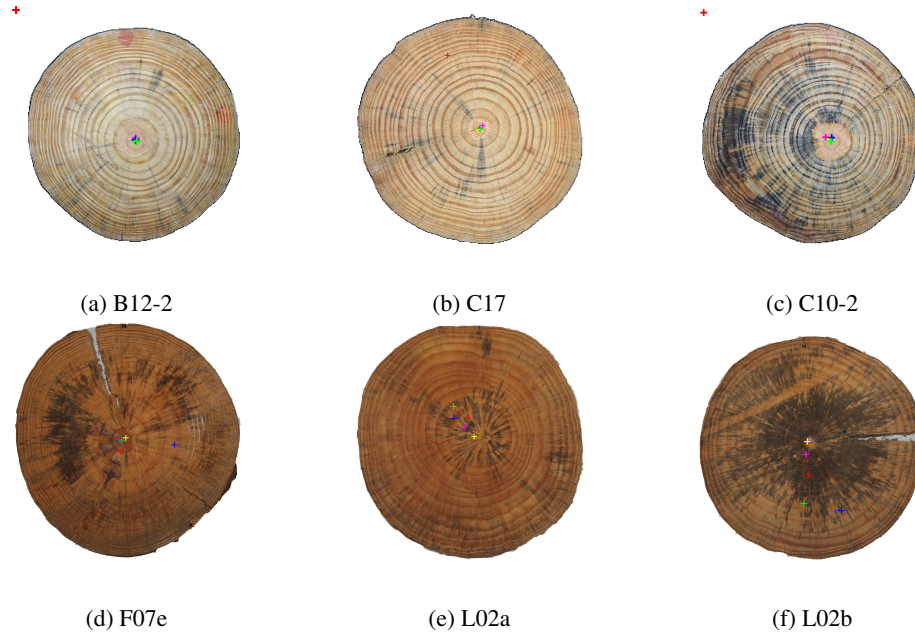
(a) B12-2   (b) C17   (c) C10-2

(d) F07e   (e) L02a   (f) L02b

Fig. 7: Results over difficult cases, e.g cracks, fungus presence in Uru2 collection. Purple, LFSA; Red, ACO; Blue, APD; Yellow, APD-PCL and Green, APD-DL

training a deep learning method. As shown in the article *Results* section, both APD and ACO outperform the APD-DL method on this dataset.

Figure 8.d to 8.f, show images from the Discs collection in which the pith is strongly off-centered. A zoom-in is shown for each disc in Figure 8.g to 8.i. On this scenario, the LFSA method (purple marker) is less precise (in all the cases, it is further from the ground truth pith), whereas APD and APD-DL perform similarly.

10   Acknowledgments

(a) D02-L1-9

(b) D01-L4-BBF-2

(c) D03-S-4

(d) D13a

(e) C05a

(f) B07a

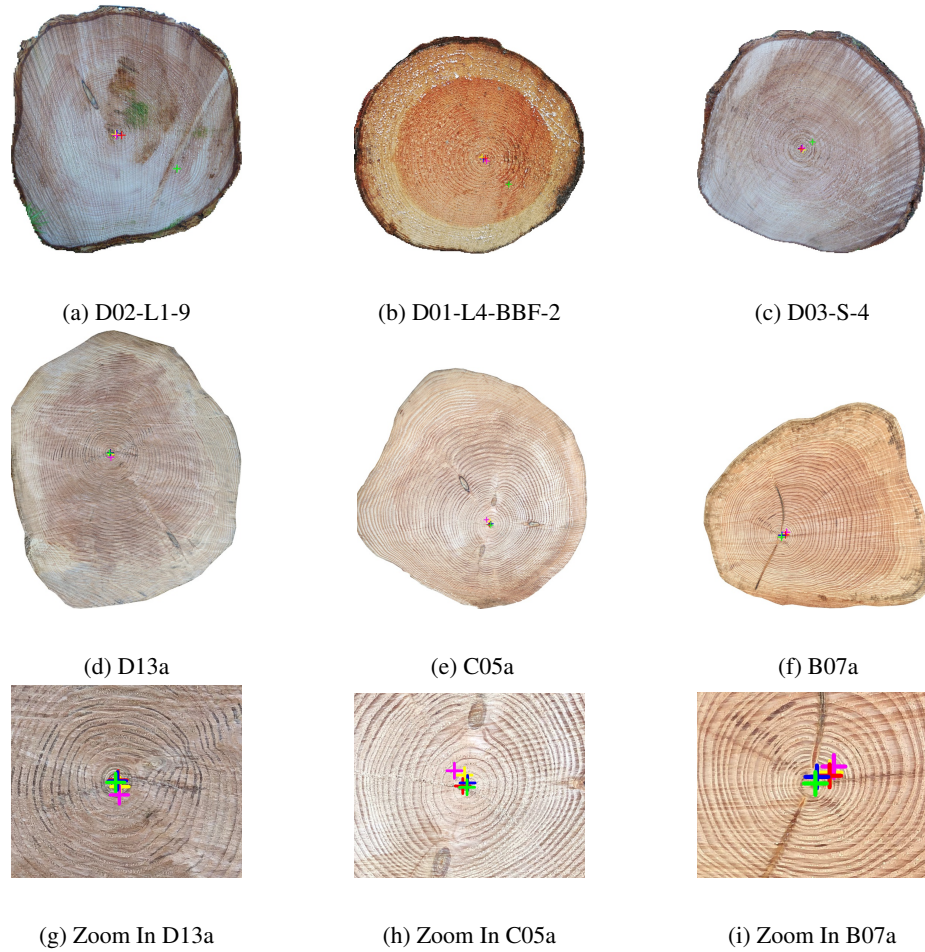(g) Zoom In D13a

(h) Zoom In C05a

(i) Zoom In B07a

Fig. 8: Results over difficult cases, e.g fungus presence, no pith eccentricity (uncentred pith position) in Forest, Logyard and Disc collections. Purple, LFSA; Red, ACO; Blue, APD; Yellow, APD-PCL and Green, APD-DL

Bibliography

J. Bigun, G.H. Granlund, and J. Wiklund. Multidimensional orientation estimation with applications to texture analysis and optical flow. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(8):775–790, 1991. `https://doi.org/10.1109/34.85668`.

Mauricio Cerda, Nancy Hitschfeld-Kahler, and Domingo Mery. Robust tree-ring detection. In Domingo Mery and Luis Rueda, editors, *Advances in Image and Video Technology, Second Pacific Rim Symposium, PSIVT 2007, Santiago, Chile, December 17-19, 2007, Proceedings*, volume 4872 of *Lecture Notes in Computer Science*, pages 575–585. Springer, 2007. `https://doi.org/10.1007/978-3-540-77129-6_50`. URL `https://doi.org/10.1007/978-3-540-77129-6_50`.

Rémi Decelle, Phuc Ngo, Isabelle Debled-Rennesson, Frédéric Mothe, and Fleur Longuetaud. Ant Colony Optimization for Estimating Pith Position on Images of Tree Log Ends. *Image Processing On Line*, 12:558–581, 2022. `https://doi.org/10.5201/ipol.2022.338`.

Marketa Dubska, Adam Herout, and Jiri Havel. Pclines — line detection using parallel coordinates. In *Proceedings / CVPR, IEEE Computer Society Conference on Computer Vision and Pattern Recognition. IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 1489 – 1494, 07 2011. `https://doi.org/10.1109/CVPR.2011.5995501`.

Martin A. Fischler and Robert C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, 24(6):381–395, jun 1981. ISSN 0001-0782. `https://doi.org/10.1145/358669.358692`. URL `https://doi.org/10.1145/358669.358692`.

Glenn Jocher, Ayush Chaurasia, and Jing Qiu. Ultralytics yolov8, 2023. URL `https://github.com/ultralytics/ultralytics`.

Pol Kennel, Philippe Borianne, and Gérard Subsol. An automated method for tree-ring delineation based on active contours guided by DT-CWT complex coefficients in photographic images: Application to abies alba wood slice images. *Comput. Electron. Agric.*, 118:204–214, 2015. `https://doi.org/10.1016/j.compag.2015.09.009`. URL `https://doi.org/10.1016/j.compag.2015.09.009`.

Wattanapong Kurdthongmee. A comparative study of the effectiveness of using popular dnn object detection algorithms for pith detection in cross-sectional images of parawood. *Heliyon*, 6(2):e03480, 2020. ISSN 2405-8440. `https://doi.org/https://doi.org/10.1016/j.heliyon.2020.e03480`. URL `https://www.sciencedirect.com/science/article/pii/S240584402030325X`.

Wattanapong Kurdthongmee and Korrakot Suwannarat. Locating wood pith in a wood stem cross sectional image using yolo object detection. In *2019 International Conference on Technologies and Applications of Artificial Intelligence (TAAI)*, pages 1–6, 2019. `https://doi.org/10.1109/TAAI48200.2019.8959823`.

Wattanapong Kurdthongmee, Korrakot Suwannarat, Praepaka Panyuen, and Naruedom Sae-Ma. 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), pages 1–6, 2018. `https://doi.org/10.1109/JCSSE.2018.8457375`.

Fleur Longuetaud, Guillaume Pot, Frédéric Mothe, Alexis Barthelemy, Rémi Decelle, Florian Delconte, Xihe Ge, Grégoire Guillaume, Théo Mancini, Tojo Ravoajanahary, Jean-Claude Butaud, Robert Collet, Isabelle Debled-Rennesson, Bertrand Marcon, Phuc Ngo, Benjamin Roux, and Joffrey Viguier. Traceability and quality assessment of Douglas fir (Pseudotsuga menziesii (Mirb.) Franco) logs: the TreeTrace_Douglas database. *Annals of Forest Science*, 79(46), November 2022. `https://doi.org/10.1186/s13595-022-01163-7`. URL `https://hal.science/hal-03658479`.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization, 2019.

Kayla Makela, Tim Ophelders, Michelle Quigley, Elizabeth Munch, Daniel Chitwood, and Asia Dowtin. Automatic tree ring detection using jacobi sets, 2020. URL `https://arxiv.org/abs/2010.08691`.

Henry Marichal, Diego Passarella, Christine Lucas, Ludmila Profumo, Veronica Casaravilla, Maria Noel Rocha Galli, Serrana Ambite, and Gregory Randall. UruDendro: An Uruguayan Disk Wood Database For Image Processing, 2023a. URL `https://iie.fing.edu.uy/proyectos/madera`.

Henry Marichal, Diego Passarella, and Gregory Randall. CS-TRD: a cross sections tree ring detection method. *arXiv:2305.10809*, 2023b.

Sergio Nesmachnow and Santiago Iturriaga. Cluster-uy: Collaborative scientific high performance computing in uruguay. In Moisés Torres and Jaime Klapp, editors, *Supercomputing*, pages 188–202, Cham, 2019. Springer International Publishing. ISBN 978-3-030-38043-4.

Kristin Norell. An automatic method for counting annual rings in noisy sawmill images. In Pasquale Foggia, Carlo Sansone, and Mario Vento, editors, *Image Analysis and Processing – ICIAP 2009*, pages 307–316, Berlin, Heidelberg, 2009. Springer Berlin Heidelberg. ISBN 978-3-642-04146-4.

Kristin Norell and Gunilla Borgefors. Estimation of pith position in untreated log ends in sawmill environments. *Computers and Electronics in Agriculture*, 63(2):155–167, 2008. ISSN 0168-1699. `https://doi.org/https://doi.org/10.1016/j.compag.2008.02.006`. URL `https://www.sciencedirect.com/science/article/pii/S016816990800077X`.

Xuebin Qin, Zichen Zhang, Chenyang Huang, Masood Dehghan, Osmar R. Zaïane, and Martin Jägersand. U^2-net: Going deeper with nested u-structure for salient object detection. *CoRR*, abs/2005.09007, 2020. URL `https://arxiv.org/abs/2005.09007`.

Rudolf Schraml and Andreas Uhl. 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*, volume 10, 2013.