Dendrochronologia 96 (2026) 126496

Contents lists available at ScienceDirect

Dendrochronologia

journal homepage: www.elsevier.com/locate/dendro

TECHNICAL NOTE

Mist4Cores: Reliable batch image stitching for dendrochronological cores

M. García-Hidalgo [a,*], Y. Ye [a], A. García-Pedrero [b,c], J.M. Olano [a]

[a] Instituto Universitario de Gestión Forestal Sostenible (iuFOR), Escuela Universitaria de Ingeniería de la Industria Forestal, Agronómica y de la Bioenergía (EiFAB), Universidad de Valladolid, Soria, Spain
[b] Department of Computer Architecture and Technology, Universidad Politécnica de Madrid, Madrid, Spain
[c] Center for Biomedical Technology, Universidad Politécnica de Madrid, Madrid, Spain

ARTICLE INFO

Keywords:
Digitization
Mosaicking
Open-source
Stitching

ABSTRACT

MIST4Cores automates the entire workflow of image stitching for camera-based digitization systems. The software provides reliable stitching with customizable settings and directory monitoring, using overlap analysis and a proven method for scientific stitching. Each module in the software contributes to a robust, reproducible, and user-friendly image stitching. The combination of free open software tools for image acquisition, image stitching, and image analysis creates an open-source ecosystem to promote further dendrochronological research.

1. Introduction

Current developments on sample digitization are pushing dendrochronology into the digital imaging world. Flatbed scanners are widespread at many dendrochronological laboratories as affordable and easy-to-use systems for sample digitization. However, the limited resolution and pixel aberrations (e.g., smearing or blooming effects) of flatbed scanners still necessitate high dependency on the re-examination of the physical sample during the dendrochronological analysis. Recent methodologies for digitization with camera-based systems reach great levels of resolution and pixel quality, effectively incorporating dendrochronology into the digital world (García-Hidalgo et al., 2022, Griffin et al., 2021, WSL, 2023).

Image capturing with digital cameras also entails certain challenges. Unlike flatbed scanners, which generate a unique image for the whole sample, camera-based systems capture one image per shot. This requires a consequent stitching (*i.e.,* mosaicking) to generate a single image for the whole sample, which is commonly created through software developed for non-scientific purposes.

General-purpose software tools for stitching, *e.g.*, PTGui (New House Int. Serv. Rotterdam, The Netherlands), Photoshop (Adobe Inc. San Jose, CA, USA) or Image Composite Editor (Microsoft research, Redmond, WA, USA), are designed primarily for landscape panoramas, where varying camera angles to the object of interest are expected. Furthermore, stitching methods intended for general consumer use often alter individual pixel characteristics (e.g., value or bit depth) to optimize visual appearance rather the strict fidelity to the original single images. In the common usage of stitching methods, some processes as lens distortion corrections are automatically applied to reduce optical artifacts (Adobe Inc., 2025). However, different smoothing or enhancement filters or complete image resizing could also be applied by default, in many cases even beyond the users' control, which could carry to loss of image integrity. The use of this general stitching software, therefore, requires several control steps to reduce artifact occurrences in the image and to improve the reliability of the whole digitized image for scientific applications (García-Hidalgo and Sangüesa-Barreda, 2025).

The generation of panorama images (*e.g.*, Whole Slide Images) from individual shots is also a recurrent problem in different scientific areas (Kumar et al., 2020). Opposite to the case of generalist stitching methods, reliability and replicability of results from the sample to the panorama output image are critical. In fact, open-source community has explored diverse reliable stitching solutions promoting user interaction and fidelity control about the entire process (Mohammadi et al., 2024). These methods are usually focused on microscopic slides which maintain uniform focal distance and fixed displacement across single images.

Here we introduce MIST4Cores, an open-source tool designed to stitch sequential macroscopic images of tree-ring cores acquired through camera-based systems. MIST4Cores automates image stitching through an easy-to-use graphical interface, and its batch processing substantially reduces time and effort for image generation. The software ensures traceability and reliability with a log file for each stitched image.

* Corresponding author.
E-mail address: miguel.garcia.hidalgo@uva.es (M. García-Hidalgo).

https://doi.org/10.1016/j.dendro.2026.126496
Received 17 October 2025; Received in revised form 6 February 2026; Accepted 23 February 2026
Available online 24 February 2026

2. Application description

MIST4Cores automates the stitching of macroscopic images using a combination of software methods in just a one-task tool. In a first step, OpenCV (Bradski and Kaehler, 2000) automatically detects the overlapping percentage in consecutive images. Afterwards, MIST (Chalfoun et al., 2017) is responsible to stitch the pairs of consecutive images through a background process within the Fiji/ImageJ environment (Schindelin et al., 2012). In a third step, the fully stitched image is exported according to user requirements. MIST4Cores is written in Python (van Rossum and Drake, 2009). Finally, the software is also compiled as an executable for its use for only-users. The main steps in the pipeline and the role of each function are outlined below:

3. Computing environment

During the first run of the software, MIST4Cores configures system dependencies and prompts the user to specify the Fiji/ImageJ installation path, the source directory of single images (the *monitoring folder*), and the desired output directory and format: TIFF or OME-TIFF (Linkert et al., 2010), Fig. 1. The software groups images according to their parent folder structure, enabling an organized workflow. Users may also define available RAM, estimated DPI to include in the metadata of the resulting image. These values are saved for future sessions in the *configuration file*. This file collects also the stitching settings for MIST (e.g. file pattern, input image path, output image path) which can be customized by the user, see Supplementary materials for a detailed description.

4. Processing

MIST4Cores monitors user-specified directories for either pre-existing or newly created image folders which are then queued for processing. Within each folder, MIST4Cores identifies and sorts images matching the expected filename pattern, ensuring only files of interest are processed.

4.1. Overlap determination

The software estimates pairwise overlaps using feature-based image registration (ORB keypoints and homography in OpenCV), generating the parameters required for MIST stitching.

4.2. Stitching execution and output

The MIST plugin in Fiji then assembles the composite image using calculated overlaps and user-defined settings. Output images are produced in TIFF and/or OME-TIFF format, with the estimated DPI included metadata. The spatial calibration of the image is compulsory in the subsequent analysis, this value is saved only to prevent possible crashes with analysis software (e.g. CooRecorder®) but it is not the real value.

4.3. Logging and reporting

Each processing run is documented with a summary log and metadata summaries, recording processing outcomes, image lists, overlap metrics, and run configurations for transparency and reproducibility of the entire stitching process.

All this pipeline is run in the backend, so the OpenCV or Fiji/ImageJ based methods are automatically managed and do not require any interaction by the user. The user is informed via terminal window to track the complete batching process in real time, in addition to the logging and reporting files generated at the end of each stitching process.

5. Sample and source images

To maintain focus distance across the entire sample length, a totally flat surface is required for MIST4Cores adequate operation. A preprocessing step is thus critical to reduce variations of focal distance among single images. To ensure a flat surface, a precise sanding or horizontal cutting across the entire surface is required. This guarantees that the basal mounting face and the surface of the core are completely parallel. For the initial core preparation, either a drum sander machine (also known as a calibrating polishing machine and commonly used in woodworking) or a cut with a microtome is essential. Successive finer grain polishing can be done with manual or mechanical alternatives, since it has little effect on surface depth.

MIST4Cores depends on established open-source tools for image processing (OpenCV) and scientific research (MIST), ensuring the fidelity among pixel values from sequential images for the stitching process. However, certain limitations appear in the longitudinal borders due to inherent methodological limitations of lens distortions and ambiguous correlations (Ghosh, Kaabouch, 2016). In areas near the top or bottom edges of the image, the number of pixels in common between two consecutive images is reduced and, therefore, the stitching may

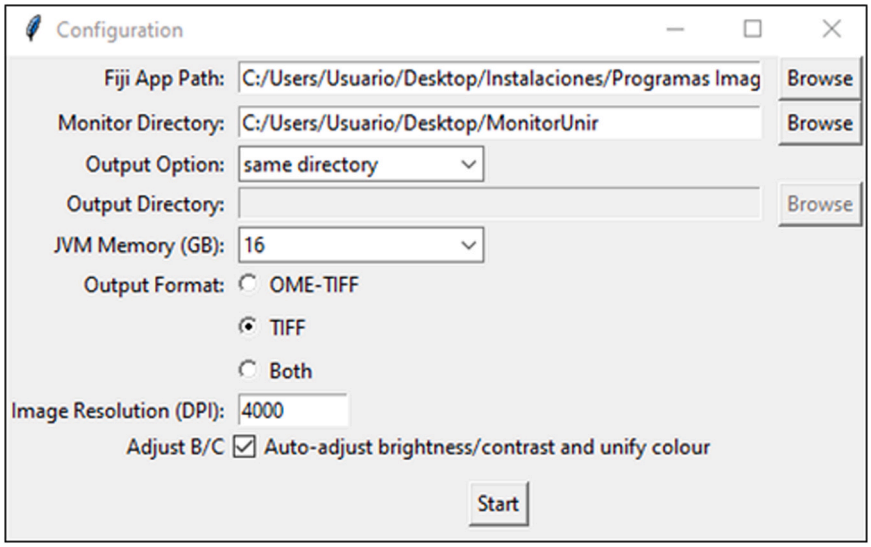

Fig. 1. Configuration window of MIST4Cores.

suffer from slight blurring or displacement. To reduce the concomitant ghosting and border artifacts in image stitching we highly recommend adapting the field of view in the shooting process.

6. Comparison with PTGui stitching

We compared the use of MIST4Cores with PTGui (New House Int. Serv. Rotterdam, The Netherlands), the dominant stitching software in dendrochronology, for the entire stitching process in 10 complete cores digitized with CaptuRING (García-Hidalgo et al., 2022) at 3.7 µm/px. We followed the standard guide by von Arx (2017) in sequential stitching in PTGui as a reference.

MIST4Cores operates in a fully continuous workflow, completing the entire stitching process in 14 min and 2 s. This included user intervention at program launching (43 s), automatic stitching process (10 min 11 s), and again user intervention at image checking (3 min 9 s). PTGui required 28 min and 47 s to complete stitching and needed user intervention at multiple intermediate steps. Thus, MIST4Cores performed faster and with a substantially lower demand for the user. Additionally, MIST4Cores produced linear images without curvature, whereas this misalignment was detected in PTGui outputs. While manual adjustment of control points could resolve this issue, the process would demand considerable user intervention. All tests were performed on an Intel® Core™ i7–7700 CPU at 3.60 GHz with 24 GB of RAM.

7. Computing requirements

MIST4Cores operates via the Java Virtual Machine (JVM), where users can specify the maximum RAM allocation. For stable performance, allocating approximately half of the system's RAM to the JVM is recommended. Systems with at least 32 GB of RAM are ideal, especially when running no other intensive applications.

CRediT authorship contribution statement

M. García-Hidalgo: Writing – original draft, Validation, Supervision, Methodology, Funding acquisition, Conceptualization; **Y. Ye:** Software, Methodology; **A. García-Pedrero:** Writing – original draft, Supervision, Software, Methodology; **J.M. Olano:** Writing – original draft, Supervision, Funding acquisition, Conceptualization.

Funding information

This research was funded by the project GIANTS (PID2023-147214NB-I00) funded by MICIU/AEI/10.13039/501100011033, and Prémio CEI-IIT Investigação, inovação e Território by Centro de Estudios Ibéricos.

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.

Acknowledgements

We gratefully acknowledge the sincere thanks to David Álvarez, Luz Laguna, and José Serra for their valuable assistance during image acquisition, software testing, and workflow optimization.

Appendix A. Supporting information

Supplementary data associated with this article can be found in the online version at doi:10.1016/j.dendro.2026.126496.

Data Availability

Application code/data and compiled release are available at GuitHub, Supplementary materials with comparison recording in Zenodo. Data of stitching comparison is available under reasonable request.

References

Adobe Inc. (2025). Correct image distortion and noise. ⟨https://helpx.adobe.com/photoshop/using/correcting-image-distortion-noise.html⟩ Accessed on 15/07/2025.
von Arx (2017). Stitching distortion-free mosaic images for QWA using PTGui. ⟨https://roxas.wsl.ch/fileadmin/user_upload/WSL/Microsite/ROXAS/ptgui_quickguide.pdf⟩ Accessed on 04/02/2026.
Bradski, G., & Kaehler, A. (2000). OpenCV. Dr. Dobb's journal of software tools, 3(2).
Chalfoun, J., Majurski, M., Blattner, T., et al., 2017. MIST: accurate and scalable microscopy image stitching tool with stage modeling and error minimization. Sci. Rep. 7, 4988. https://doi.org/10.1038/s41598-017-04567-y.
García-Hidalgo, M., Sangüesa-Barreda, G., 2025. Considerations on Image Resolution, measurement, accuracy and crossdating in the digital era. Tree-Ring Res. 81 (2), 87–90. https://doi.org/10.3959/TRR2024-10.
García-Hidalgo, M., García-Pedrero, Á., Colón, D., Sangüesa-Barreda, G., García-Cervigón, A.I., López-Molina, J., Alonso-Gómez, V., 2022. CaptuRING: a do-it-yourself tool for wood sample digitization. Methods Ecol. Evol. 13 (6), 1185–1191. https://doi.org/10.1111/2041-210X.13847.
Ghosh, D., Kaabouch, N., 2016. A survey on image mosaicing techniques. J. Vis. Commun. Image Represent. 34, 1–11. https://doi.org/10.1016/j.jvcir.2015.10.014.
Griffin, D., Porter, S.T., Trumper, M.L., Carlson, K.E., Crawford, D.J., Schwalen, D., McFadden, C.H., 2021. Gigapixel macro photography of tree rings. Tree-Ring Res. 77 (2), 86–94. https://doi.org/10.3959/TRR2021-3.
Kumar, N., Gupta, R., Gupta, S., 2020. Whole slide imaging (WSI) in pathology: current perspectives and future directions. J. Digit. Imaging 33 (4), 1034–1040. https://doi.org/10.1007/s10278-020-00351-z.
Linkert, M., Rueden, C.T., Allan, C., Burel, J.-M., Moore, W., Patterson, A., Loranger, B., Moore, J., Neves, C., MacDonald, D., Tarkowska, A., Sticco, C., Hill, E., Rossner, M., Eliceiri, K.W., Swedlow, J.R., 2010. Metadata matters: access to image data in the real world. J. Cell Biol. 189 (5), 777–782. https://doi.org/10.1083/jcb.201004104.
Mohammadi, F.S., Mohammadi, S.E., Mojarad Adi, P., Mirkarimi, S.M.A., Shabani, H., 2024. A comparative analysis of pairwise image stitching techniques for microscopy images. Sci. Rep. 14 (1), 9215. https://doi.org/10.1038/s41598-024-59626-y.
van Rossum, G., Drake, F.L., 2009. Python 3 reference manual. Python Softw. Found. Available at ⟨https://docs.python.org/3/reference/⟩.
Schindelin, J., Arganda-Carreras, I., Frise, E., Kaynig, V., Longair, M., Pietzsch, T., Cardona, A., 2012. Fiji: an open-source platform for biological-image analysis. Nat. Methods 9, 676–682. https://doi.org/10.1038/nmeth.2019.
WSL (2023). Skippy: The new high-resolution image capturing system for tree rings developed at WSL. ⟨https://www.wsl.ch/it/services-produkte/skippy/⟩ Accessed on 04/02/2026.