Contents lists available at ScienceDirect

SoftwareX

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

CoverPy: Automated estimates of plant area index, vegetation cover, crown cover, crown porosity, and uncertainties from digital cover photography in Python

Luke A. Brown[a,*], Sylvain G. Leblanc[b]

[a] School of Science, Engineering & Environment, University of Salford, Manchester M5 4WT, United Kingdom
[b] Canada Centre for Remote Sensing, Natural Resources Canada, Ottawa, Ontario K1A 0E4, Canada

ARTICLE INFO

Keywords:
Fraction of vegetation cover (FCOVER)
Leaf area index (LAI)
Plant area index (PAI)
Uncertainty

ABSTRACT

Implemented in Python, CoverPy enables automated estimation of plant area index (PAI), the fraction of vegetation cover (FCOVER), crown cover (CC), and crown porosity (CP) from digital cover photography (DCP). When compared to available alternatives, a key strength of CoverPy is the incorporation of end-to-end uncertainty propagation, enabling uncertainties due to within-plot variability and the user-specified extinction coefficient to be quantified. CoverPy is made available to the community on an open-source basis via GitHub, and should prove useful for researchers and citizen scientists interested in quantifying vegetation structure using inexpensive, non-specialist hardware.

Metadata

Nr	Code metadata description	
C1	Current code version	v0.1.0
C2	Permanent link to code/repository used for this code version	https://github.com/luke-a-brown/coverpy
C3	Permanent link to reproducible capsule	–
C4	Legal code license	MIT
C5	Code versioning system used	Semantic versioning
C6	Software code languages, tools and services used	Python
C7	Compilation requirements, operating environments and dependencies	CoverPy makes use of the `imageio`, `numpy`, `rawpy`, `scikit-image`, and `uncertainties` modules, in addition to the `glob` module included in the Python Standard Library.
C8	If available, link to developer documentation/manual	README at https://github.com/luke-a-brown/coverpy
C9	Support email for questions	l.a.brown4@salford.ac.uk

1. Motivation and significance

Owing to its low cost and wide accessibility, digital cover photography (DCP) is an increasingly popular method for estimating canopy biophysical variables such as plant area index (PAI), the fraction of vegetation cover (FCOVER), crown cover (CC), and crown porosity (CP). In contrast to alternative techniques requiring costly and specialist hardware, such as digital hemispherical photography (DHP), ceptometry, the LI-COR LAI-2000 series of instruments, and terrestrial laser scanning [1,2], the key advantage of DCP is that it requires only a standard, inexpensive digital camera with a 15° to 30° field-of-view [3–6]. Further advantages include increased spatial resolution, reduced sensitivity to illumination conditions and exposure, and a narrower rectangular footprint that is well-suited to experimental plots [3,5,6].

Images acquired using DCP must subsequently be processed to estimate gap fraction (and the proportion of large, between-crown, and small, within-crown gaps), from which PAI, FCOVER, CC, and CP can be determined. Whilst several software packages have been made available to facilitate DCP processing, few have been provided on an open-source basis. One popular commercially available solution is WinSCANOPY [7], and an example of an open-source solution is the recently released coveR2 package [8]. Another freely available solution is CAN-EYE, which allows the processing of digital images acquired at nadir, but only for the derivation of FCOVER [9]. Although undoubtedly useful, these packages do not provide for the quantification or propagation of uncertainties in the derived canopy biophysical variables, despite the

* Corresponding author.
E-mail address: l.a.brown4@salford.ac.uk (L.A. Brown).

https://doi.org/10.1016/j.softx.2024.101767
Received 12 April 2024; Received in revised form 13 May 2024; Accepted 17 May 2024
2352-7110/© 2024 The Authors. Published by Elsevier B.V. This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/).

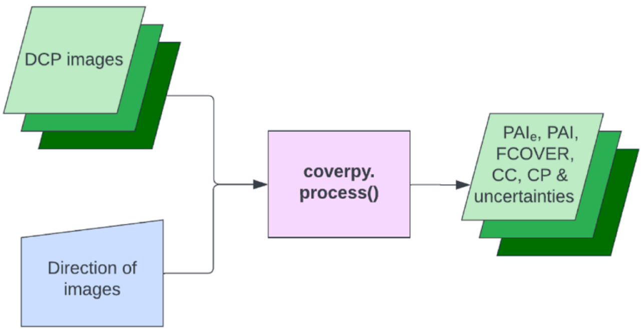


Fig. 1. Overview of the workflow for processing a set of DCP images with CoverPy.

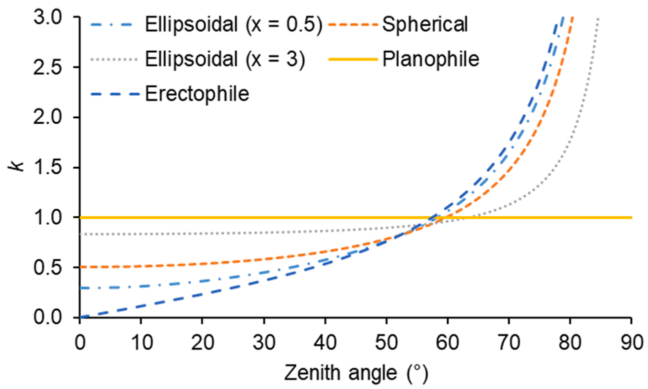


Fig. 2. Extinction coefficient (k) associated with various leaf angle distributions as a function of zenith angle (near-nadir values are applicable to DCP). Note that the planophile and erectophile values in this figure correspond to completely horizontal and vertical leaves, so represent extreme cases.

Table 1
Equations to compute extinction coefficient (k) as a function of zenith angle (θ) for planophile, spherical, erectophile, and ellipsoidal leaf angle distributions [26–29]. In the case of the ellipsoidal leaf angle distribution, x corresponds to the ratio of the horizontal to vertical semi-axis, where $x < 1$ implies a more planophile canopy and $x > 1$ a more erectophile canopy.

Leaf angle distribution	k
Planophile	1
Spherical	$\frac{1}{2\cos(\theta)}$
Erectophile	$\frac{2\tan(\theta)}{\pi}$
Ellipsoidal	$\frac{\sqrt{x^2+\tan(\theta)^2}}{x+1.744\,(x+1.182)^{-0.733}}$

```
#import required modules
import coverpy

#run the main function on example data
results = coverpy.process('example_data', direction = 'up')
```

Fig. 3. Illustrative example of how CoverPy can be used to process the example dataset provided in the GitHub repository.

increasing importance of uncertainty estimates in assessing measurement quality and fitness-for-purpose [10–12].

For the estimation of PAI, a primary source of uncertainty in DCP is the specification of the extinction coefficient (k), which is dependent on the leaf angle distribution of the canopy in question, necessitating ancillary data collection or assumptions [13–15]. Whilst many previous studies have assumed a spherical leaf angle distribution ($k = 0.5$) [4–6, 16], previous work has shown that this assumption is often violated in broadleaf canopies [17]. As such, accounting for the uncertainty associated with the extinction coefficient is a key, yet underexplored, issue.

This original software paper describes CoverPy (https://github.com/luke-a-brown/coverpy), an open-source Python module for automated estimation of PAI, FCOVER, CC and CP, and uncertainties from DCP.

2. Software description

2.1. Software architecture

CoverPy is implemented in Python, and is closely related to the HemiPy module [18], which is designed for processing DHP (i.e. images acquired with a fisheye lens) as opposed to DCP data. The key differences between CoverPy and HemiPy are related to post-classification analysis:

- HemiPy computes multi-angular estimates of gap fraction (based on the optical centre and projection function of the adopted camera and fisheye lens), whereas CoverPy requires no information on the properties of the camera used;
- Once the classification is performed, CoverPy identifies clusters of pixels, in order to identify small, within-crown, and large, between-crown gaps. A distinction between types of gaps is not made in HemiPy;
- Because HemiPy makes use of multi-angular information, its PAI derivation methods do not require assumptions on canopy leaf angle distribution or the specification of an extinction coefficient, which is the case for CoverPy.

CoverPy makes use of the `imageio`, `numpy`, `rawpy`, `scikit-image`, and `uncertainties` modules [19–22], in addition to the `glob` module included in the Python Standard Library. It consists of a single main function: `coverpy.process()`, and the workflow for processing a set of DCP images from a single measurement plot is to pass this function the directory of images to be processed, along with the direction of the images (i.e. upwards- or downwards-facing) (Fig. 1). As in HemiPy, all images within a directory are processed together to provide a single value and uncertainty for each canopy biophysical variable, meaning that each directory should contain multiple images corresponding to the same measurement plot [18]. Processing of a directory containing a single image is possible, but discouraged, as a less reliable biophysical variable and overoptimistic estimate of its uncertainty will be obtained.

2.2. Software functionality

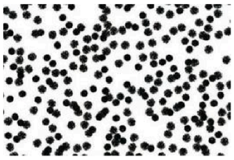
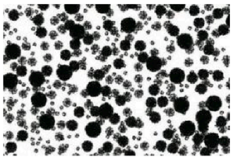
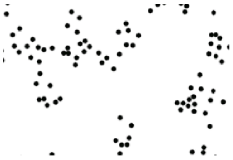
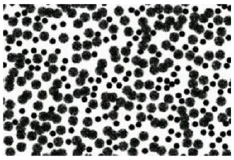
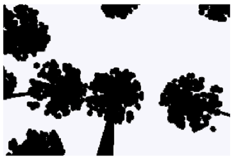
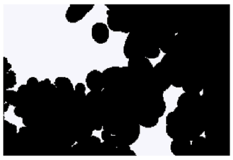
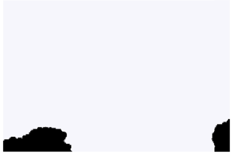
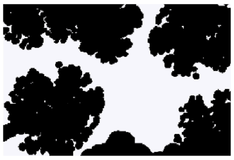
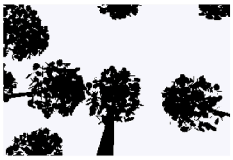
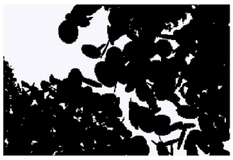
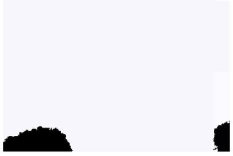
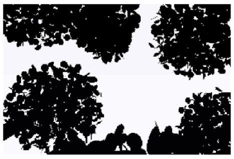
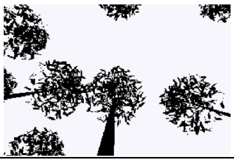
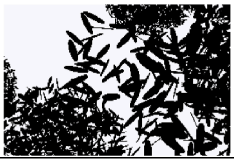
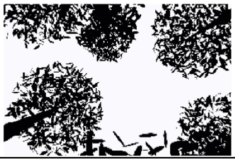
Images are read into memory as `numpy` arrays using either `imageio` (for 8-bit formats, including JPEG, PNG, GIF, BMP and TIFF), or `rawpy` (for RAW formats, including NEF, CR2, CR3 and PEF). In the latter case, by default, gamma correction is applied and contrast stretching is carried out so that 1% of pixels at the high and low ends of the histogram are saturated. The result is stored in 8-bit form for further analysis. This procedure corresponds to the recommendations of Macfarlane et al. [23]. Alternatively, RAW images may be processed in their full bit-depth by setting the `pre_process_raw` parameter of `coverpy.process()` to `False`. To reduce computation time, downsampling, which makes use of the `measure.block_reduce()` function of `scikit-image`, is possible by setting the `down_factor` parameter of `coverpy.process()` to greater than one. A default value of 3 is assumed.

Once loaded into memory, pixels are classified as belonging to the canopy or background. The classification approach is dependent on whether the `direction` parameter of `coverpy.process()` has been set to `'up'` (as is default) or `'down'`, reflecting upwards- or downwards-

```
{'paie': 1.3650264364222338+/-0.5594233011666294,
 'cc': 0.6790182637262555+/-0.04615407729369503,
 'cp': 0.26807838801210454+/-0.015366531769920201,
 'pai': 1.787822289984458+/-0.7295454264405354,
 'clumping': 0.7635134901657928+/-0.09185464414530152,
 'fcover': 0.49465464806141046+/-0.030766861785177055}
```

Fig. 4. Results obtained from processing the example dataset with CoverPy. Note that results are reported in default 64-bit 'double' precision, but are not considered reliable to this level of precision.

Table 2
Example orthographic and DCP images simulated by POV-Ray for four virtual forest scenes according to a planophile, random, and erectophile leaf angle distribution. Note that the FCOVER values represent the mean FCOVER derived from orthographic images corresponding to the three leaf angle distributions.

	Scene 11	Scene 31	Scene 38	Scene 60
PAI	2.80	3.56	2.32	3.91
FCOVER	0.29	0.41	0.05	0.44
Example orthographic image				
Example DCP image (planophile)				
Example DCP image (random)				
Example DCP image (erectophile)				

facing images, respectively. In the former case, Ridler and Calvard's [24] clustering algorithm is adopted to separate the sky from the canopy on the basis of the blue band of the image, whereas in the latter case, green vegetation is distinguished from the soil by Meyer and Neto's [25] colour index based approach.

After classification, gap fraction is calculated as

$$P = \frac{n_{background}}{n} \tag{1}$$

where $n_{background}$ and n are the number of pixels classified as the background, and the total number of pixels in the image, respectively, as determined using the `numpy` statistics functions. From gap fraction, FCOVER and effective PAI (PAI$_e$) (i.e. assuming a random distribution of plant material) can be directly determined as

$$FCOVER = 1 - \overline{P} \tag{2}$$

and

$$PAI_e = \frac{-\ln(\overline{P})}{k} \tag{3}$$

where $\overline{P}$ is the mean gap fraction over all images, and k is the extinction coefficient provided as an input to the `coverpy.process()` function, which is set to 0.5, corresponding to a spherical leaf angle distribution, by default.

Clusters of pixels classified as gaps are determined using the `measure.label()` function of `scikit-image`, according to 8-connectivity. Each cluster is designated as a small, within-crown or large, between-crown gap according the criterion proposed by Macfarlane et al. [5,6] (i.e. gaps larger than 1.3% of the total image area are designated as large, between-crown gaps). For each image, this enables CC and CP to be determined as

$$CC = 1 - \frac{n_{large}}{n} \tag{4}$$

and

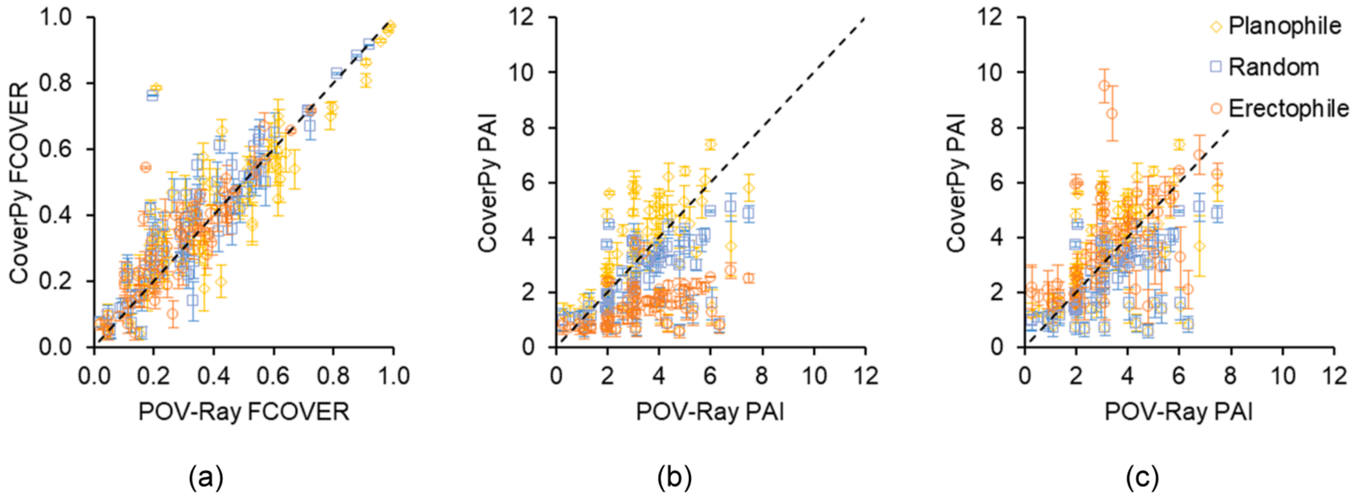


Fig. 5. Verification of CoverPy's FCOVER (a) and PAI outputs for simulated DCP images generated with POV-Ray using an extinction coefficient corresponding to a spherical leaf angle distribution ($k = 0.5$) (b), in addition to an alternative extinction coefficient ($k = 0.2$) for the erectophile cases (c). Error bars represent standard uncertainties.

Table 3
Total processing time for 75 virtual forest scenes (20 images per scene) using CoverPy and coveR2's default settings, for planophile, random, and erectophile leaf angle distributions.

	Processing time (h:mm)	
Leaf angle distribution	CoverPy	coveR2
Planophile	0:41	2:28
Random	0:47	2:32
Erectophile	1:01	2:35

$$CP = 1 - \frac{1 - \overline{P}}{CC} \tag{5}$$

where n_{large} represents the number of pixels labelled as belonging to large gaps. The mean CC and CP values over all images are reported. From these values, it is possible to correct for the effects of canopy clumping, enabling PAI as opposed to PAI$_e$ to be derived as

$$PAI = -\overline{CC}\,\frac{\ln(\overline{CP})}{k} \tag{6}$$

where $\overline{CC}$ and $\overline{CP}$ are the mean CC and CP values over all images. The extinction coefficient (k) should be set according to leaf angle distribution of the canopy being analysed. Values of k for various leaf angle distributions are provided in Fig. 2, and can be computed according to the equations detailed in Table 1.

The clumping index, Ω, is calculated as the ratio of PAI$_e$ to PAI, which is equivalent to

$$\Omega = \frac{(1 - CP)\ln(\overline{P})}{\ln(CP)(1 - \overline{P})} \tag{7}$$

To account for within-plot variability, uncertainties in gap fraction, CC, and CP are determined as the standard error of the mean over all images being analysed. For FCOVER, PAI$_e$, and PAI, the `uncertainties` module is used to automatically and analytically propagate these uncertainties (as well as the uncertainty associated with the extinction coefficient for the latter two variables, which is set to 0.2 by default) through Eqs. (2), (3) and (6). This achieves accordance with the International Standards Organisation (ISO) Guide to the Expression of Uncertainty in Measurement (GUM) [30], in addition to the recommendations of the Fiducial Reference Measurements for Vegetation (FRM4VEG) project [10]. CoverPy returns results as a dictionary of `ufloat` values (i.e. as defined by the `uncertainties` module).

3. Illustrative examples

3.1. Example dataset

To enable users to check that CoverPy is installed and working correctly, an example dataset for a single measurement plot containing upwards-facing images is provided in the GitHub repository. Fig. 3 demonstrates the processing of this dataset, whilst Fig. 4 demonstrates the results obtained. Note that by embedding this simple example in a loop (or nested loop), multiple plots (and sub-plots) could easily be processed in batch (an example of this is provided in the README file).

3.2. Verification against simulated images

To verify CoverPy's outputs, we adopted a three-dimensional simulation-based approach. Using the Persistence of Vision Raytracer (POV-Ray) [31], we simulated 20 DCP images for 75 virtual forest scenes with a PAI ranging from 0.3 to 7.5. Scenes could represent a Neyman, plantation, or random stem distribution, and for each scene, three sets of simulations were conducted with planophile, random, and erectophile leaf angle distributions. The parameters used to generate the virtual forest scenes are described in detail by [32]. Simulations were carried out at a resolution of 4500×3000 pixels, with a 30° field-of-view across the diagonal. It is worth noting that of the variables derived by CoverPy, only PAI was an explicit input to the POV-Ray simulations. Nevertheless, reference FCOVER values were determined from very high spatial resolution (1 cm) orthographic images simulated by POV-Ray for a 30 m x 40 m area of each virtual forest scene (Table 2).

The images simulated with POV-Ray were processed with CoverPy using an extinction coefficient corresponding to a spherical leaf angle distribution ($k = 0.5$), and agreement with the reference values was quantified using the root mean square error (RMSE) and bias. Good overall agreement was observed between CoverPy's FCOVER outputs and the reference FCOVER values derived from orthographic images (RMSE = 0.10, bias = 0.03) (Fig. 5a), with little difference in performance observed between the planophile (RMSE = 0.12, bias = 0.02), random (RMSE = 0.11, bias = 0.04), and erectophile (RMSE = 0.08, bias = 0.02) leaf angle distributions.

CoverPy's PAI outputs demonstrated more moderate agreement with the PAI values used to simulate the images (RMSE = 1.88, bias = −0.91) (Fig. 5b). Particularly poor agreement was observed in the case of the

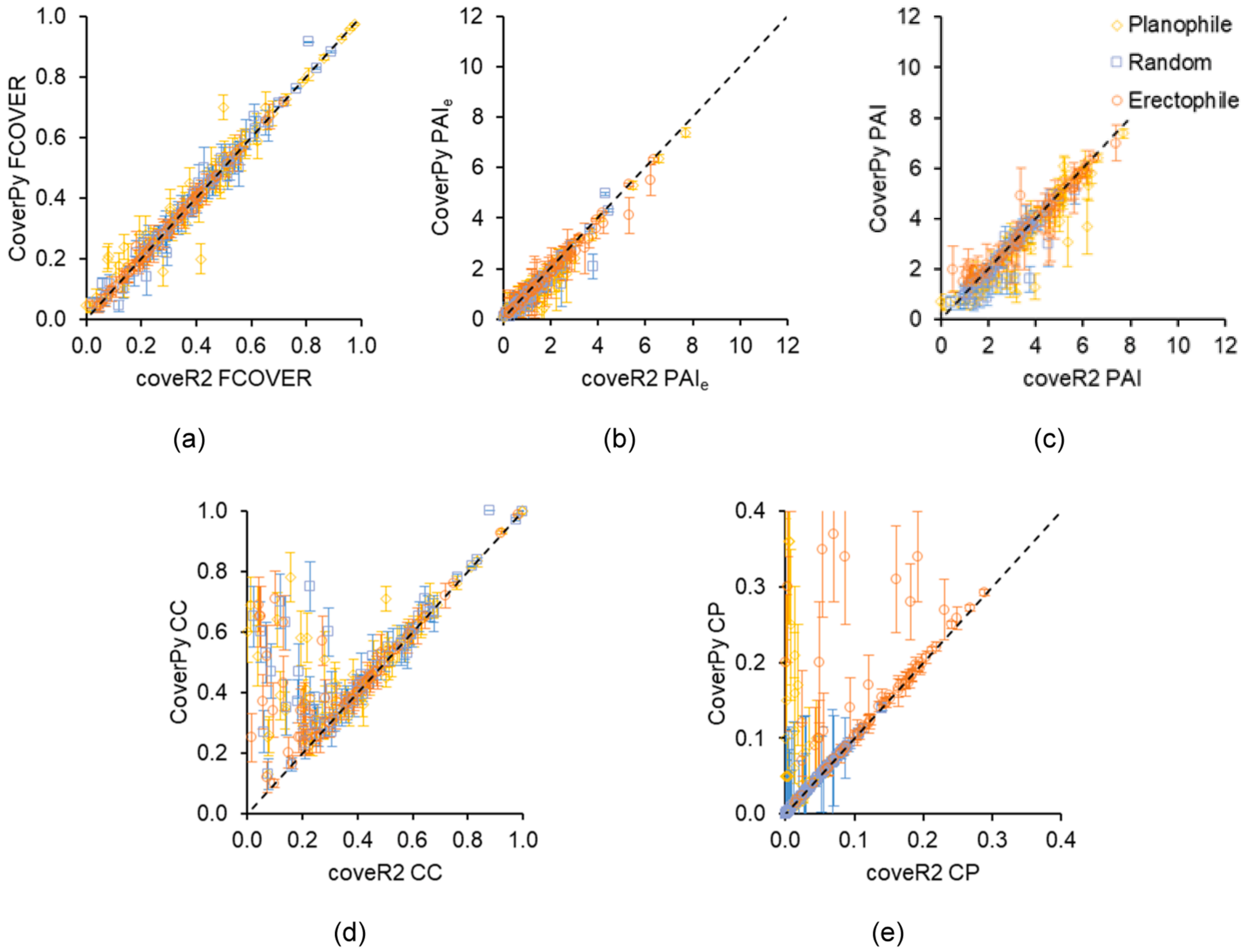


Fig. 6. Comparison between CoverPy and coveR2 FCOVER (a), PAI_e (b), PAI (c), CC (d), and CP (e) outputs. Error bars represent standard uncertainties.

erectophile leaf angle distribution (RMSE = 2.27, bias = −1.82) when compared to the planophile (RMSE = 1.73, bias = −0.07) and random (RMSE = 1.58, bias = −0.85) leaf angle distributions, indicating that the assumed spherical leaf angle distribution was ill-suited to erectophile canopies. Indeed, when an alternative extinction coefficient corresponding to a more erectophile canopy was used to calculate PAI for the erectophile cases ($k = 0.2$), agreement was substantially increased, both for these cases (RMSE = 1.66, bias = 0.43), and overall (RMSE = 1.65, bias = −0.16) (Fig. 5c).

3.3. Benchmarking against the coveR2 package

To further asses CoverPy, we also processed the simulated images described in Section 3.2 with the coveR2 package [8], allowing a direct comparison between CoverPy and coveR2's outputs to be made. Default settings were used, with the exception that an extinction coefficient (k) of 0.2 was set for the erectophile cases (as was the case with the CoverPy processing). In addition to assessing the agreement between outputs, we also calculated the time taken to process each batch of images (i.e. 75 virtual forest plots, with 20 images per plot) on a personal computer with an Intel i7–6500U 2.5 GHz processor and 8 GB of memory. Unlike CoverPy, which provides a single value and uncertainty per measurement plot, coveR2 processes each image individually. Therefore, CoverPy outputs were compared against the mean of the coveR2 outputs from all 20 images in each plot.

In all cases, coveR2 processing was substantially slower than CoverPy processing (Table 3). This is likely because CoverPy downsamples images by a factor of three in order to speed up computation by default (Section 2.2), whereas images are processed in their native resolution by coveR2. Despite differences in processing time, overall agreement between CoverPy and coveR2 outputs was high for FCOVER (RMSE = 0.03, bias = 0.01), PAI_e (RMSE = 0.31, bias = −0.17), and PAI (RMSE = 0.57, bias = −0.20) (Fig. 6). For FCOVER, little difference in performance was observed between the planophile (RMSE = 0.05, bias = 0.01), random (RMSE = 0.03, bias = 0.00), and erectophile (RMSE = 0.03, bias = 0.00) leaf angle distributions (Fig. 6a). Meanwhile, for PAI, greater disagreement was observed for the planophile leaf angle distribution (RMSE = 0.73, bias = −0.35) than for the random (RMSE = 0.49, bias = −0.21) and erectophile (RMSE = 0.46, bias = −0.05) leaf angle distributions (Fig. 6c).

Some disagreement between CoverPy and coveR2 was observed in the case of CC (Fig. 6d) and CP (Fig. 6e), where CoverPy overestimated coveR2's outputs in a number of cases. This is likely due to differences in the way clusters of pixels are identified as gaps by coveR2 (8-connectivity is used by CoverPy, meaning larger gaps would be identified than if using 4-connectivity). It is worth noting that these outputs were identified as having a higher uncertainty. Despite this, overall agreement was reasonable (RMSE = 0.18, bias = 0.08 for both CC and CP).

4. Impact and conclusions

A principal strength of CoverPy is that it is well-suited to batch processing, meaning it should prove useful for processing long time-series of DCP observations from automated digital cameras and routine sampling. Indeed, several authors have demonstrated the use of timelapse camera systems for capturing DCP images, which, when processed, can provide rich information on the temporal dynamics of vegetation structure [16,33,34].

It is worth noting that DCP is a primary data collection method adopted by environmental monitoring networks including the Terrestrial Ecosystem Research Network (TERN). TERN routinely collects DCP at 10 'SuperSites' spread across Australia, and over 12,000 unprocessed DCP images acquired between 2013 and 2023 are currently available via the TERN EcoImages Portal [35]. Under the European Space Agency (ESA) funded 'Ground Reference Observations Underlying Novel Decametric Vegetation Data Products from Earth Observation (GROUNDED EO)' project [36], these images are being processed using CoverPy, and the results will be made available to the community in the near future.

Since CoverPy provides quantitative estimates of uncertainty in accordance with ISO GUM and FRM4VEG standards [10,30], it is very well-suited to deriving reference data for the calibration and validation of remotely sensed data products [37]. Owing to the narrow rectangular footprint associated with DCP, CoverPy should prove particularly useful for calibration/validation of high spatial resolution estimates from unoccupied aerial vehicle (UAV) and satellite sensors [3].

Finally, because DCP images can be acquired with low-cost digital cameras and even smartphones [38], there is high potential for researchers and citizen scientists interested in quantifying vegetation structure using inexpensive, non-specialist hardware to adopt the module for DCP processing [39].

CRediT authorship contribution statement

Luke A. Brown: Writing – review & editing, Writing – original draft, Visualization, Validation, Supervision, Software, Resources, Project administration, Methodology, Investigation, Funding acquisition, Formal analysis, Data curation, Conceptualization. **Sylvain G. Leblanc:** Writing – review & editing, Visualization, Software, Methodology, Investigation, Formal analysis.

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.

Data availability

The CoverPy module is available via GitHub (https://github.com/luke-a-brown/coverpy).

Acknowledgments

This activity was carried out under the Living Planet Fellowship, a programme of and funded by the European Space Agency. The view expressed in this publication can in no way be taken to reflect the official opinion of the European Space Agency.

References

[1] Garrigues S, Shabanov NV, Swanson K, Morisette JT, Baret F, Myneni RB. Intercomparison and sensitivity analysis of leaf area index retrievals from LAI-2000, AccuPAR, and digital hemispherical photography over croplands. Agric For Meteorol Jul. 2008;148(8–9):1193–209. https://doi.org/10.1016/j.agrformet.2008.02.014.

[2] Yan G, et al. Review of indirect optical measurements of leaf area index: recent advances, challenges, and perspectives. Agric For Meteorol Feb. 2019;265:390–411. https://doi.org/10.1016/j.agrformet.2018.11.033.

[3] Pekin B, Macfarlane C. Measurement of crown cover and leaf area index using digital cover photography and its application to remote sensing. Remote Sens Dec. 2009;1(4):1298–320. https://doi.org/10.3390/rs1041298.

[4] Chianucci F, Cutini A. Estimation of canopy properties in deciduous forests with digital hemispherical and cover photography. Agric For Meteorol Jan. 2013;168:130–9. https://doi.org/10.1016/j.agrformet.2012.09.002.

[5] Macfarlane C, et al. Estimation of leaf area index in eucalypt forest using digital photography. Agric For Meteorol Apr. 2007;143(3–4):176–88. https://doi.org/10.1016/j.agrformet.2006.10.013.

[6] Macfarlane C, Grigg A, Evangelista C. Estimating forest leaf area using cover and fullframe fisheye photography: thinking inside the circle. Agric For Meteorol Sep. 2007;146(1–2):1–12. https://doi.org/10.1016/j.agrformet.2007.05.001.

[7] Regent Instruments, "WinSCANOPY," 2020. https://regentinstruments.com/assets/winscanopy_about.html (accessed Jun. 28, 2021).

[8] Chianucci F, Ferrara C, Puletti N. coveR: an R package for processing digital cover photography images to retrieve forest canopy attributes. Trees Dec. 2022;36(6):1933–42. https://doi.org/10.1007/s00468-022-02338-5.

[9] Weiss M, Baret F. CAN-EYE V6.4.91 user manual. Avignon, France: Institut National de la Recherche Agronomique; 2017.

[10] Brown LA, et al. Fiducial reference measurements for vegetation bio-geophysical variables: an end-to-end uncertainty evaluation framework. Remote Sens 2021;13(16):3194. https://doi.org/10.3390/rs13163194.

[11] Raupach MR, et al. Model-data synthesis in terrestrial carbon observation: methods, data requirements and data uncertainty specifications. Glob Chang Biol Mar. 2005;11(3):378–97. https://doi.org/10.1111/j.1365-2486.2005.00917.x.

[12] Richardson AD, Dail DB, Hollinger DY. Leaf area index uncertainty estimates for model–data fusion applications. Agric For Meteorol Sep. 2011;151(9):1287–92. https://doi.org/10.1016/j.agrformet.2011.05.009.

[13] Pisek J, Ryu Y, Alikas K. Estimating leaf inclination and G-function from leveled digital camera photography in broadleaf canopies. Trees Oct. 2011;25(5):919–24. https://doi.org/10.1007/s00468-011-0566-6.

[14] Kattenborn T, Richter R, Guimarães-Steinicke C, Feilhauer H, Wirth C. AngleCam: predicting the temporal variation of leaf angle distributions from image series with deep learning. Methods Ecol Evol Nov. 2022;13(11):2531–45. https://doi.org/10.1111/2041-210X.13968.

[15] Stovall AEL, Masters B, Fatoyinbo L, Yang X. TLSLEAF: automatic leaf angle estimates from single-scan terrestrial laser scanning. New Phytol Jul. 2021;17548. https://doi.org/10.1111/nph.17548.

[16] Toda M, Richardson AD. Estimation of plant area index and phenological transition dates from digital repeat photography and radiometric approaches in a hardwood forest in the Northeastern United States. Agric For Meteorol Sep. 2017:1–10. https://doi.org/10.1016/j.agrformet.2017.09.004.

[17] Pisek J, Sonnentag O, Richardson AD, Mõttus M. Is the spherical leaf inclination angle distribution a valid assumption for temperate and boreal broadleaf tree species? Agric For Meteorol Feb. 2013;169:186–94. https://doi.org/10.1016/j.agrformet.2012.10.011.

[18] Brown LA, et al. HemiPy: a Python module for automated estimation of forest biophysical variables and uncertainties from digital hemispherical photographs. Methods Ecol Evol Sep. 2023;14(9):2329–40. https://doi.org/10.1111/2041-210X.14199.

[19] Harris CR, et al. Array programming with NumPy. Nature Sep. 2020;585(7825):357–62. https://doi.org/10.1038/s41586-020-2649-2.

[20] M. Rierchert, "rawpy: RAW image processing for Python, a wrapper for libraw," 2021. https://github.com/letmaik/rawpy (accessed Jun. 29, 2021).

[21] van der Walt S, et al. scikit-image: image processing in Python. PeerJ Jun. 2014;2(1):e453. https://doi.org/10.7717/peerj.453.

[22] E.O. Lebigot, "Uncertainties: a Python package for calculations with uncertainties," 2017. http://pythonhosted.org/uncertainties (accessed Jun. 29, 2021).

[23] Macfarlane C, Ryu Y, Ogden GN, Sonnentag O. Digital canopy photography: exposed and in the raw. Agric For Meteorol Oct. 2014;197:244–53. https://doi.org/10.1016/j.agrformet.2014.05.014.

[24] Ridler TW, Calvard S. Picture thresholding using an iterative selection method. IEEE Trans Syst Man Cybern 1978;8(8):630–2. https://doi.org/10.1109/TSMC.1978.4310039.

[25] Meyer GE, Neto JC. Verification of color vegetation indices for automated crop imaging applications. Comput Electron Agric Oct. 2008;63(2):282–93. https://doi.org/10.1016/j.compag.2008.03.009.

[26] Campbell GS. Extinction coefficients for radiation in plant canopies calculated using an ellipsoidal inclination angle distribution. Agric For Meteorol Apr. 1986;36(4):317–21. https://doi.org/10.1016/0168-1923(86)90010-9.

[27] Jones HG, Vaughan RA. Remote sensing of vegetation: principles, techniques, and applications. Oxford, United Kingdom: Oxford University Press; 2010.

[28] Meter Group. AccuPAR PAR/LAI ceptometer model LP-80 operator's manual. Pullman, Washington, United States: Meter Group; 2018.

[29] Jones HG. Plants and microclimate. 2nd ed. Cambridge: Cambridge University Press; 1992.

[30] Working Group 1 of the Joint Committee for Guides in Metrology. Evaluation of measurement data — guide to the expression of uncertainty in measurement. Paris, France: Bureau International des Poids et Mesures; 2008.

[31] Persistence of Vision Raytracer Pty. Ltd., "POV-Ray: the Persistence of Vision Raytracer," 2021. http://www.povray.org/ (accessed May 03, 2024).

[32] Leblanc SG, Fournier RA. Hemispherical photography simulations with an architectural model to assess retrieval of leaf area index. Agric For Meteorol Aug. 2014;194:64–76. https://doi.org/10.1016/j.agrformet.2014.03.016.

[33] Chianucci F, Bajocco S, Ferrara C. Continuous observations of forest canopy structure using low-cost digital camera traps. Agric For Meteorol Sep. 2021;307: 108516. https://doi.org/10.1016/j.agrformet.2021.108516.

[34] Ryu Y, et al. Continuous observation of tree leaf area index at ecosystem scale using upward-pointing digital cameras. Remote Sens Environ Nov. 2012;126:116–25. https://doi.org/10.1016/j.rse.2012.08.027.

[35] TERN, "EcoImages Portal." https://ecoimages.tern.org.au/ (accessed Jul. 29, 2023).

[36] ESA, "Ground Reference Observations Underlying Novel Decametric Vegetation Data Products from Earth Observation – GROUNDED EO," 2023. https://eo4society.esa.int/projects/grounded-eo/ (accessed Jul. 29, 2023).

[37] Fernandes R, et al. "Global leaf area index product validation good practices," in Best practice for satellite-derived land product validation, 2.0. In: Fernandes R, Plummer S, Nightingale J, editors. Land Product Validation subgroup (Committee on Earth Observation Satellites Working Group on Calibration and Validation); 2014. https://doi.org/10.5067/doc/ceoswgcv/lpv/lai.002.

[38] Fang H, Ye Y, Liu W, Wei S, Ma L. Continuous estimation of canopy leaf area index (LAI) and clumping index over broadleaf crop fields: an investigation of the PASTIS-57 instrument and smartphone applications. Agric For Meteorol May 2018; 253–254:48–61. https://doi.org/10.1016/j.agrformet.2018.02.003.

[39] Newman G, Wiggins A, Crall A, Graham E, Newman S, Crowston K. The future of citizen science: emerging technologies and shifting paradigms. Front Ecol Environ Aug. 2012;10(6):298–304. https://doi.org/10.1890/110294.