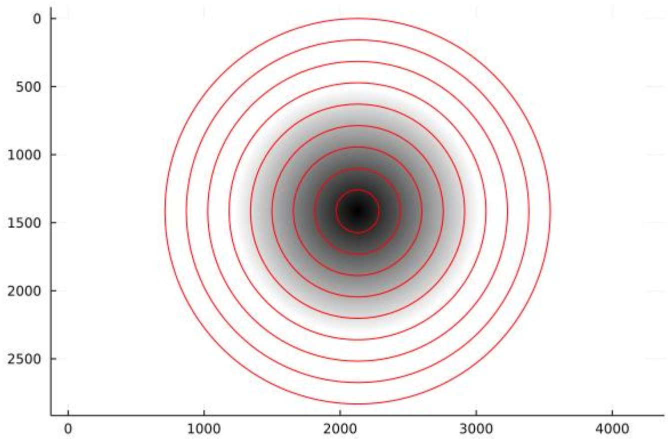
Automated canopy and hemispherical photo processing

There has been quite a bit of progress in automated image processing since we first implemented mobile phone LAI processing. Algorithms like segmentation and binarization mean that we can now batch process large image archives efficiently. I've done some testing with a hemispherical image, but the methods below would work on a mobile phone or other planar camera image.

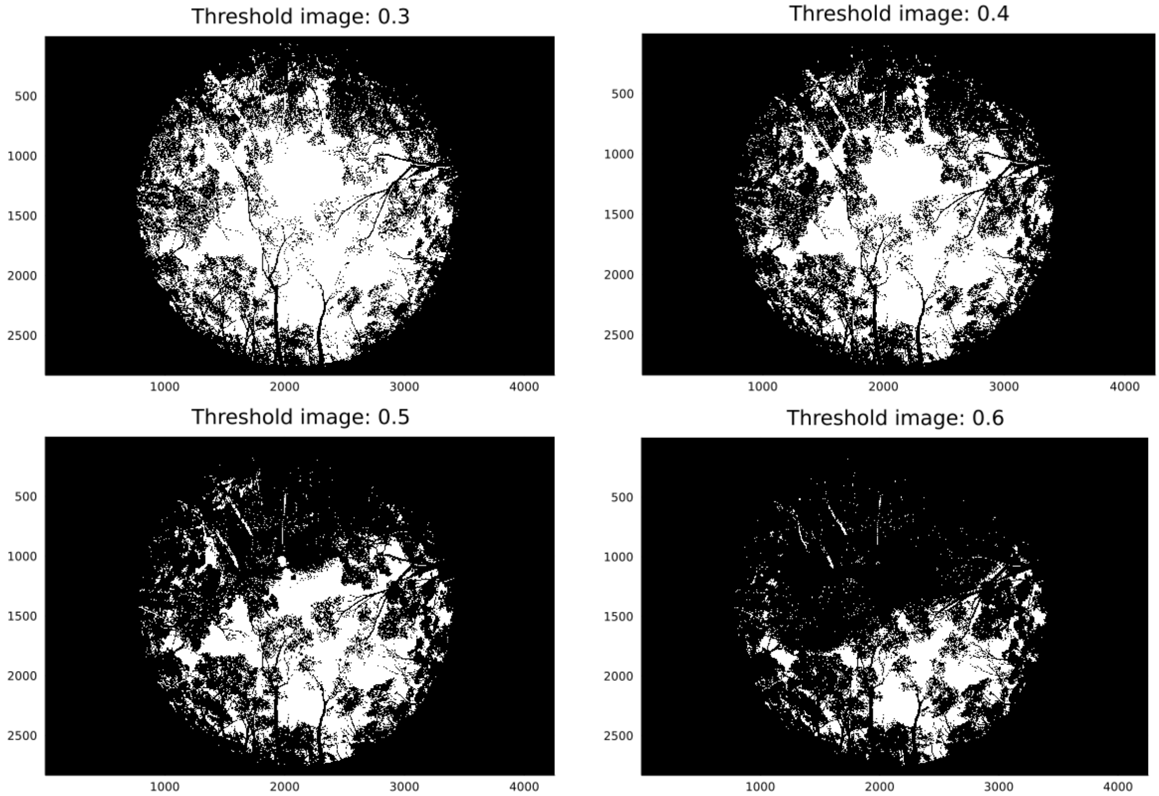
This is a poor image because the sun is within the field of view, so it disturbs the scaling (stretch) of brightness values, such that sky and vegetation become less distinct.

We need to know what zenith angle each pixel is, because this influences both the projection of vegetation elements and the path length through the canopy. For the above image I'm assuming that the shorter vertical dimension is at the horizon (zenith angle = pi/2) and that the centre of the image is at the zenith (zenith angle = 0). The background for the plot below is actually an image of zenith angles and I'm going to use nine zenith rings to calculate gap fractions (probability).



Thresholding

Conventional processing requires a user specified threshold to determine areas of sky and vegetation. These are some examples where I've used the "value" channel from a Hue-Saturation-Value (HSV) transformed image:



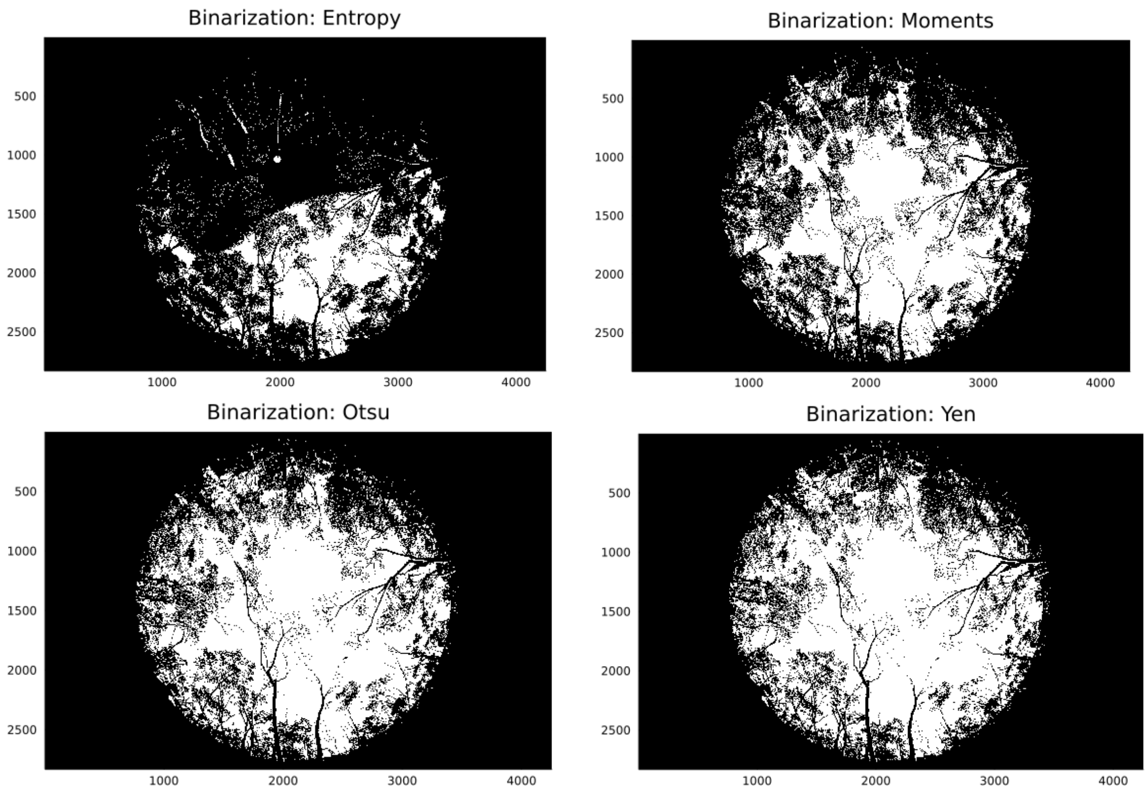
In the above case it looks like a threshold of between 0.3 and 0.4 might be the best threshold value to use. The issue here is not that thresholding doesn't do a good job, it is that the user must specify an appropriate threshold value, and this will vary with lighting conditions, image quality and the processing, such as brightness normalisation.

Binarization

Automated processing of images can be done using three methods:

1. Using a fixed threshold, which is subject to the problems mentioned above
2. Automated classification of either raw pixels or segmented image regions. Again, this is subject to lighting conditions and so training of the classification is required for every image.
3. Binarization, which uses histogram based adaptive thresholds, based on the assumption that image brightness is bi-modal and an appropriate threshold value can be determined automatically.

Binarization has been around for a long time, but the libraries to run this processing have come a long way in the past ten years. The following threshold images demonstrate some of the common algorithms, all of which require no parameter setting by the user.

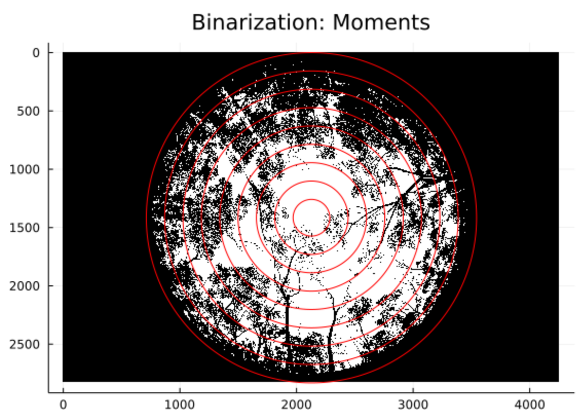


To me, the Moments algorithms looks like it is doing a pretty good job, but you would want to test this on a range of images and lighting conditions and see if you could consistently get the automatic binarization performing as well as manual thresholding.

Cover and Leaf Area Calculation

The calculation of Cover and LAI is relatively straight forward once you have a thresholded image. That is to say that the theory is well established. However, for many reasons the true relationship between the physical structure and the estimates may vary. That said, the key benefit of using photography is that you have an efficient and objective measure of structure, so that you are better placed to compare across space and time.

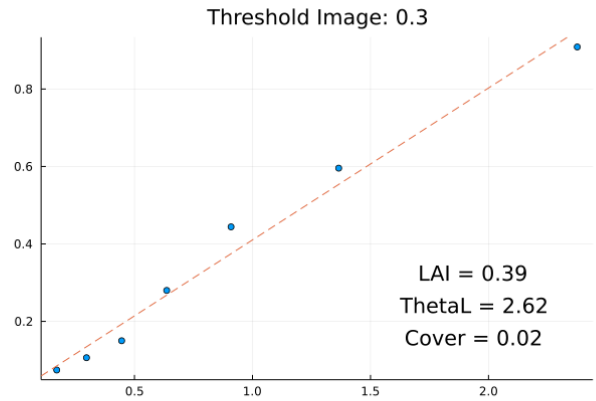
I'm going to reference the image below with zenith rings overlayed to describe the cover and LAI estimation methodology.

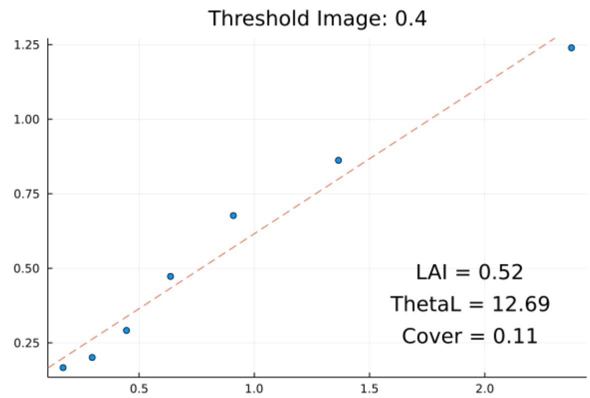


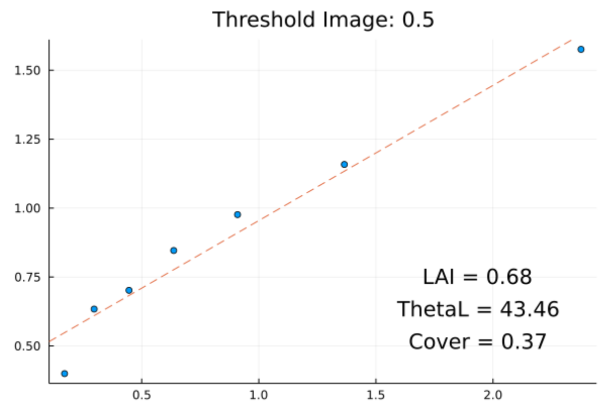
The processing follows the following steps:

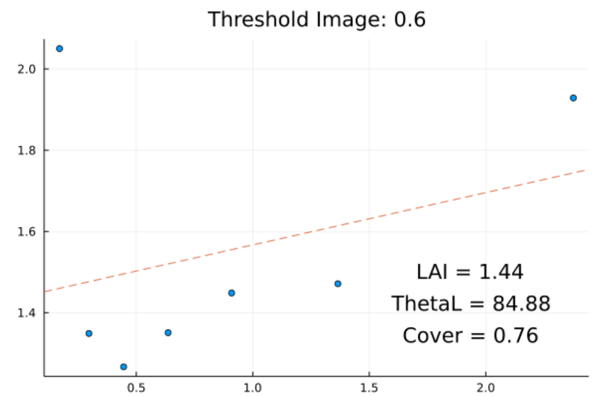
1. Estimate the gap probability (P_{gap}) for each zenith ring above by counting the total number of pixels (n_{pixels}) and the total number of gap pixels that are shown in white (n_{gaps}):
$$P_{gap}(\theta) = n_{gaps}/n_{pixels}$$
2. Calculation the horizontal (L_H) and vertical (L_V) projection of the leaf area using the equation:
$$P_{gap}(\theta) = exp(L_H + 2\ L_V tan(\theta)/\pi)$$
3. Calculate the total leaf area index as the geometric sum of the projected components:
$$LAI = \sqrt{L_H^2 + L_V^2}$$
4. Calculate the mean projection angle (thetaL):
$$\tan(\theta_L) = \frac{L_H}{L_V}$$
5. Calculate cover based on only the horizontally projected component:
$$Cover = exp(-L_H)$$

The graphs below show the linear model (i.e., step 2 above) for several thresholded and binarized images. Note that only the estimates are shown in the graphs, but these are derived from a linear model where standard error (uncertainty) is known.



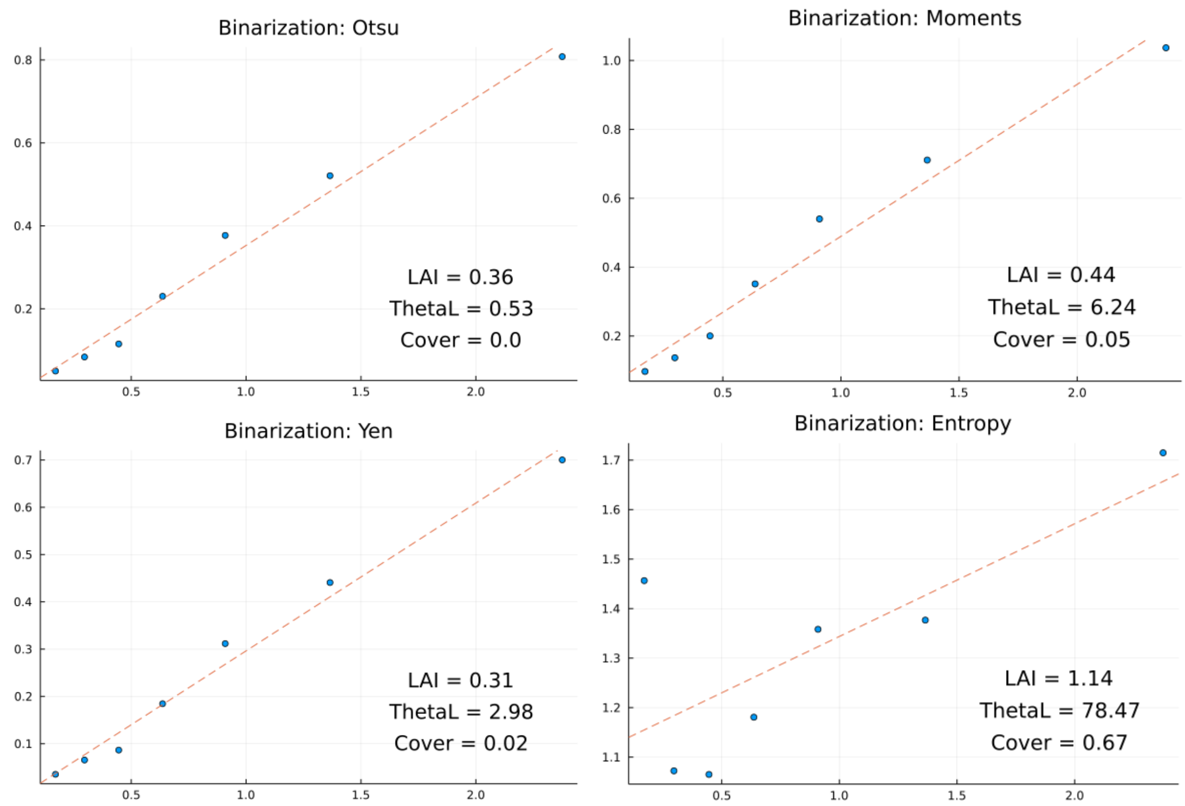






Understandably the Cover and LAI increase as you increase the manual threshold. The very low cover in the 0.3 and 0.4 threshold graphs is due to the fact that the corresponding thresholded images are almost completely gap for the zenith rings near the zenith.

The results for the binarized images shown below are more consistent except for the “Entropy” algorithm. All of these are close to the estimates using manual thresholds of 0.3 and 0.4. Again, the objective is not absolute accuracy, it is stability of the measure so that values are comparable across space and time. This is what will make the measurement useful in downstream modelling of light interception, primary productivity, biomass etc.



Where to from here?

The nice thing about these binarization algorithms is that they make estimation of cover and LAI a fully automated process. In its current form the code could be used to process large archives of image data in seconds. It could also be packaged into a citizen science app (by someone other than me :-) so that anyone can record and upload estimates of cover and height.

The initial job would be to ensure stability of the estimates across different illumination conditions and image quality.

These approaches aren’t new and have been tested by others. There are some interesting recent papers such as the following:

Díaz, G. M., Negri, P. A., & Lencinas, J. D. (2021). Toward making canopy hemispherical photography independent of illumination conditions: A deep-learning-based approach. *Agricultural and Forest Meteorology*, *296*, 108234.