

Hemispherical photography for the estimation of woodland canopy traits

John L. Godlee

2019-10-22

This guide serves as an introduction to the use of hemispherical photography for estimating woodland tree canopy variables in ecological research. Much of this guide has been informed by personal fieldwork, along with scattered references found throughout the scientific and practitioner literature. This guide is an attempt to bring those ideas into a cohesive resource which assumes little prior knowledge of the technique. This guide focuses on practical application of hemispherical photography, rather than being an in-depth analysis of the science of optics. It is designed to provide novice users with the means to obtain robust results quickly for use in their research. The manual contains a number of checklists for fieldwork which the user is encouraged to adapt for their own purposes and take into the field to serve as a reminder of best practice. Analysis of hemispherical photographs is covered with an effort to stick to free, open-source, and actively maintained software in order to maximise accessibility, and to ensure that the guide remains relevant into the future.

1 What is hemispherical photography?

Hemispherical photography is a technique used to estimate plant canopy traits, using an extreme wide-angle or fisheye camera lens. Hemispherical photography is most commonly used to estimate the canopy cover of trees in a forest or woodland setting ([Seidel et al., 2011](#); [Macfarlane et al., 2014](#)), and this is the focus of the current guide. There are however, many other creative applications of this technique. For example, hemispherical photography may also be used to estimate the ground cover of agricultural crops, by pointing the camera downwards instead of upwards ([Demarez et al., 2008](#)).

2 Measurements of woodland tree canopies

Hemispherical photographs can be analysed in multiple ways to provide different metrics related to the form of the woodland canopy. All these metrics are related and rely fundamentally on the classification of pixels in the photo image into either plant material or sky. Different metrics are used depending on the convention of the chosen field of research and according to the research question being asked. Some common metrics are:

- Gap fraction (site factor) - Proportional coverage of plant canopy material as viewed from a single point with some given angular field of view (%).
- Canopy cover - Proportional coverage of plant canopy material per unit ground area covered (%).
- Leaf/Plant Area Index (LAI/PAI) - Area of leaf (LAI) or plant (PAI) per unit ground area ($\text{m}^2 \text{ m}^{-2}$).

3 Hemispherical photos in the field

3.1 Capturing informative photos

Taking hemispherical photos under field conditions which are representative of the woodland canopy requires the careful balancing of camera settings. Images must be properly exposed and without visual aberrations which would otherwise bias analysis. Confident use of the many settings available on a modern DSLR camera is necessary to adjust for environmental conditions which may change throughout the day to gather consistent and informative images.

Some common issues which create bias in results:

- Lens flare - Bright sunlight reflects within the lens mechanism due to material imperfections in the lens glass, causing haze and starburst artifacts in the image ([Figure 1](#)).
- Blooming - An oversaturated part of the image bleeds over into other pixels on the image sensor during image capture. This is the most common problem in hemispherical photography.
- Obstructions between camera and tree canopy - Understorey vegetation, man-made objects, people etc. can all contribute to an inaccurate estimate of the area of plant material in the frame.



Figure 1: An example of a hemispherical photo with excessive lens flare. In this example, the canopy gap fraction will be over-represented as the bright lens flare will be classified as sky during the binarization process. This photo should be discarded from further analysis.

Capturing the right image relies on manual adjustment of three key camera settings to balance the image exposure and sharpness:

- Shutter speed
- Lens aperture
- ISO - analogous to the sensitivity of the sensor to incoming light

Below is a checklist to guide you in taking informative hemispherical photographs:

- Take photos under a uniformly overcast sky where possible, ideally before the sun has risen too high in the sky, or just before sunset, this avoids lens flare and helps to increase the contrast between plant material

and sky. I find in the morning the photos are generally better due to the quality of the light and the generally more consistent cloud cover. At high latitudes you will have a longer suitable period to capture photos than in the tropics.

- Ensure that the camera is level on the tripod and the lens is pointing straight up. Use a spirit level attached to the camera hotshoe to do this.
- Adjust the tripod so that the top of the camera lens is 1 m above the ground, or above any understorey vegetation you wish to exclude from the analysis, whichever is higher.
- Turn the camera so the top of the camera body is facing north, bring a compass! This ensures that the top of the captured photo is also facing north, which is necessary for calculating LAI.
- Make use of the visual display on the camera, if there is one, to get a good view of the photo before you take it.
- Set the camera:
 - Manual shooting mode
 - Manual focus
 - Set the focus to infinity
 - Set the exposure compensation to -0.7. This makes thresholding the image easier later on
 - Capture fine jpeg and RAW images at the same time
 - Ensure the camera time and date is accurate (this is purely for ease of matching photos to sites during analysis)
 - Set the Aperture to about 7. This is only a guideline.
 - Adjust the ISO and shutter speed so the photo is neutrally exposed but the shutter speed is always over 1/60sec, otherwise you will introduce camera shake when you capture the photo. If it's windy check the photos for blurry tree branches and adjust the shutter speed accordingly.
 - If the camera attempts to intelligently rotate photos based on the camera orientation, disable this feature and make sure all photos are captured in landscape orientation, never portrait.
- Make sure everybody ducks down below the camera when the image is taken!
- Cover the lens with the lens cap between photos to prevent accidents.

A daily kit list for taking hemispherical photos:

- Camera with appropriate fisheye lens
- Lens cap
- Lens cleaning solution and lens cloth
- Tripod
- At least 2 fully charged batteries for camera
- 2 SD cards
- Spirit level hotshoe attachment for camera
- Compass
- Notebook and pencil
- GPS unit
- Tape measure > 2 m

- Waterproof bag to cover camera

An ideal list of products for a high quality DSLR camera setup:

- Nikon D750 DSLR Camera Body
- Sigma 8 mm f3.5 Circular Fisheye EX DG For Nikon Lens
- Hotshoe Fit Spirit Level
- Integral USB SD Card Reader
- 2x Sandisk Ultra 30 MB/s SDHC Card 16 GB Class 10
- Hard peli-case to fit equipment in, e.g. Peli-1520 with foam
- A sturdy tripod

3.2 Sampling strategy

In order to estimate the mean value of various woodland canopy traits over an area of woodland, multiple photos must be taken. Determining the spatial distribution of these multiple photos is therefore necessary to ensure that a representative sample of the woodland canopy is achieved. The shape of the sampling area will somewhat determine the spatial pattern of photos, circular plots may require a different layout to rectangular plots and linear transects, for example.

During analysis it is common to restrict the field of view of the hemispherical photo to 60 degrees. See [section 4.6](#) for more information. With this in mind, a simple calculation with some sensible assumptions can estimate the minimum distance of photos to avoid pseudo-replication of the canopy area:

$$D = 2D_\theta = 2(h \tan \theta) \quad (1)$$

where θ is the angular field of view in degrees, h is the height from the camera lens to the base of the canopy, and D is the minimum distance of photos to avoid pseudo-replication of canopy area.

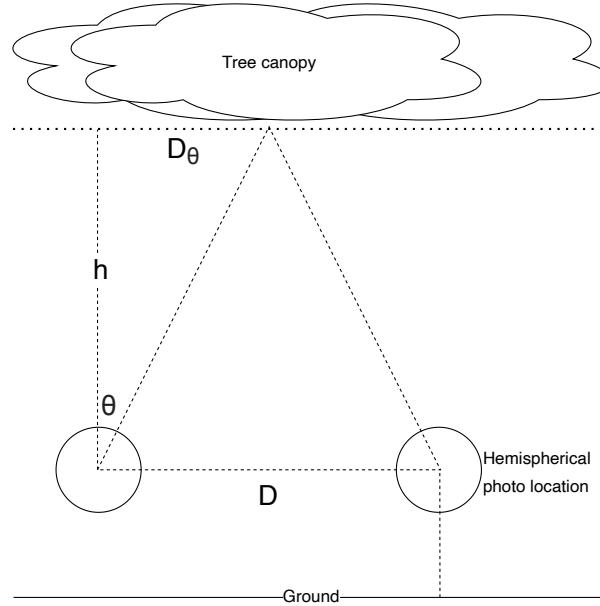


Figure 2: Schematic diagram of the measurements used when calculating the minimum distance between hemispherical photo locations.

4 Processing hemispherical photographs

There are many proprietary options for processing and analysing woodland canopy traits from hemispherical photographs, e.g. WinPhot, CAN-EYE. I think processing with freely available tools is more flexible, reproducible, and future-proof, and should be preferred for peer-reviewed research. This part of the guide demonstrates a workflow for processing hemispherical photos using ImageJ v1.x (Schneider et al., 2012).

4.1 Differentiating between plant and sky

In order to calculate various woodland canopy traits it is necessary to partition the hemispherical photo into areas of plant material and sky. A straightforward method for this relies on the assumption that the sky will be much brighter than the plant material. This is largely true for a correctly exposed hemispherical photo. The checklist below describes how to partition an image based on pixel brightness, in ImageJ. This process returns an image with white pixels for sky and black pixels for plant material. Monospace blue font refers to menu items in ImageJ to be selected.

1. Open ImageJ
2. **File → Open**, then select an image to process.
3. Visually inspect the image to see that there isn't large amounts of lens flare. See [Figure 1](#) for an example of lens flare. If you have lots of lens flare, the photo should be discarded.
4. **Image → Type → 8-bit**
5. **Image → Adjust → Threshold**, tick **Dark background** and manually adjust the image so all sky is entirely red and the branches are grey, or as near as you can get it. Click **Apply** to binarize the image.
6. Save the newly thresholded image as a **.tif** in a separate directory. The image should have black plant material and white sky.
7. Repeat for all images.

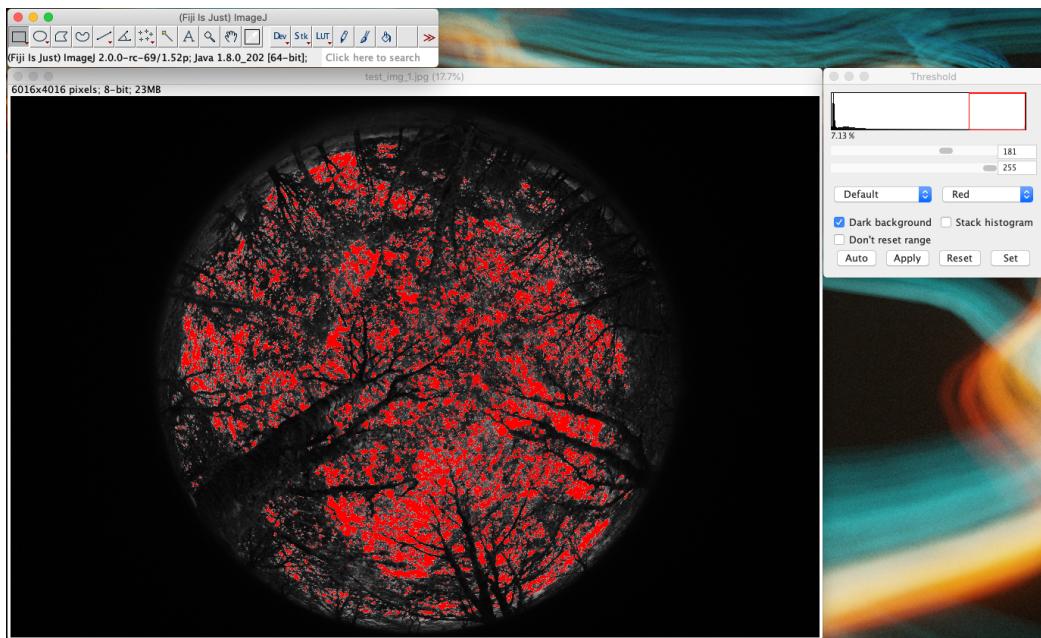


Figure 3: Screenshot of the binarization thresholding process in ImageJ.

The above process can be automated with an ImageJ macro ([Code 1](#)), but this removes the ability to manually threshold each image, relying instead on a number of binarization algorithms provided in ImageJ. This technique should only be used when you are confident that all images are consistently exposed. Macros should be saved as a `.ijm` file and called within ImageJ with `Plugins → Macros → Run...`. For a full list of macro functions available in ImageJ, consult the documentation (<https://imagej.nih.gov/ij/developer/macro/functions.html>).

```

1 // User inputs
2 /////////////////////////////////
3 input_path = "/Users/username/Desktop/input/";
4 output_path = "/Users/username/Desktop/output/";
5 algorithm = "Default";
6 ///////////////////////////////
7 // END user inputs
8
9 list = getFileList(input_path);
10
11 for (i=0; i<(list.length); i++) {
12   open(""+input_path+list[i]+"");
13   file_name = getInfo("image.filename");
14   run("8-bit");
15   setAutoThreshold(algorithm);
16   setOption("BlackBackground", false);
17   run("Convert to Mask");
18   saveAs("tif",(""+output_path+file_name+""));
19   image_id = getImageID();
20   selectImage(image_id);
21   close();
22 }
```

Code 1: ImageJ macro to binarize all images in a nominated directory. The macro can also be found in `binarize.ijm` in the supplementary material.

I find that the [Huang \(Huang and Wang, 1995\)](#) binarization algorithm normally works well for thresholding canopy photos, but you should experiment with different algorithms to find the one which works best for your data. `Default` is also widely appropriate for canopy photos.

An alternative to simple thresholding of 8-bit greyscale images is to use a colour thresholding technique. Plant material often reflects very little blue, while the sky generally has much more, so one can threshold using only the blue channel of the image ([Brusa and Bunker 2014, Code 2](#)).

```

1 // User inputs
2 ///////////////////////////////
3 input_path = "/Users/johngodlee/Desktop/input/";
4 output_path = "/Users/johngodlee/Desktop/output/";
5 algorithm = "Default";
6 ///////////////////////////////
7 // END user inputs
8
9 list = getFileList(input_path);
10
11 for (i=0; i<(list.length); i++){
12   open(""+input_path+list[i]+"");
13   file_name = getInfo("image.filename");
14   run("Split Channels");
15   selectWindow(file_name+" (blue)");
16   setAutoThreshold(algorithm);
17   run("Convert to Mask");
18   saveAs("tif",(""+output_path+"blue_"+file_name+""));
19   close("*");
20 }
```

Code 2: ImageJ macro to binarize images by the blue colour channel. The macro can also be found in `binarize_blue_channel.ijm`.

4.2 Cropping a circular image

Sometimes, it's desirable to crop a hemispherical photo to a smaller circle with a known angle of view (zenith angle). A common convention is to crop an image to a 60 degree angular field of view. See [section 4.6](#) for more information. Fisheye lenses have different projection functions which map the curved image onto a flat surface. Here is a list of common projection functions for different lenses:

- Equisolid (equal area) - $R = 2f \sin(\theta/2)$
- Equidistant - $R = f\theta$
- Orthographic - $R = f \sin(\theta)$
- Thoby fisheye - $R = 1.47f \sin(0.713\theta)$

where R is the radial position of a point on the image on the sensor, f is the focal length of the lens, and θ is the angle in radians of the desired angular radius of the cropped image. Here is a diagram I made of what those values equate to on the camera lens.

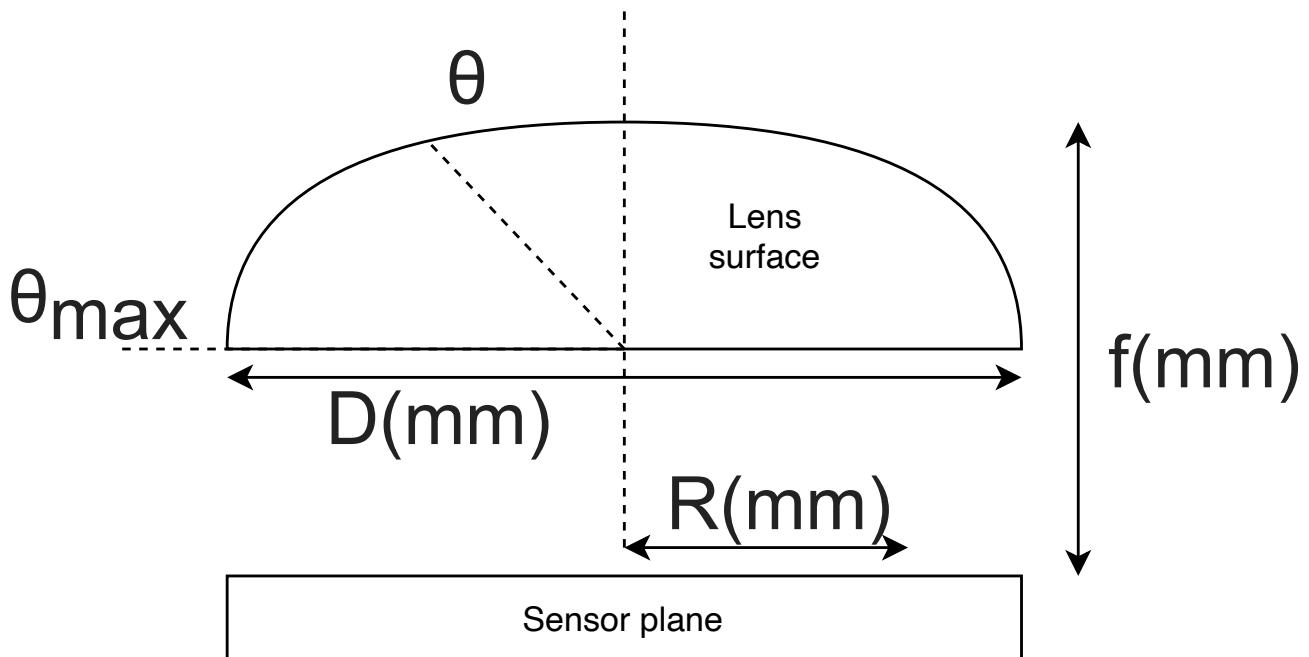


Figure 4: Schematic diagram of a camera lens and sensor, demonstrating the measurements used when cropping an image to a given angular field of view.

The lens I use (Sigma 8 mm) uses an equisolid projection, so I can use the equation above to create a function in the R programming language which will give me the radius of the circle in pixels for a desired angular field of view ([Code 3](#)).

```

1  fov.px <- function(deg_theta, focal_length_mm, pixel_pitch_um){
2    # Convert degrees of theta to radians
3    rads_theta <- NISTunits::NISTdegToRadian(deg_theta)
4
5    # Calculate radius of circle drawn by angle of view
6    # (rads_theta and max_rads_theta) in mm projected onto the sensor plane
7    R <- 2 * focal_length_mm * sin(rads_theta / 2)
8
9    # Calculate the px per mm on the sensor, i.e. the pixel pitch
10   sensor_px_per_mm_flat <- 1/pixel_pitch_um * 1000
11
12   # Multiply the mm radius of the desired circle by the number of
13   # pixels per mm on the sensor, to get the number of pixels radius
14   # of the desired circle
15   pixels_for_theta <- round(R * sensor_px_per_mm_flat, 0)
16
17   return(pixels_for_theta)
18 }
19
20 # Run the function
21 fov.px(90, 8, 5.95)

```

Code 3: R function to calculate the pixel radius of a circle with a given zenith angle. Also found in [fov_func.R](#) in the supplementary materials.

The pixel pitch of the sensor is the real distance (in μm) from the centre of one pixel on the sensor to the centre of the next, in my case it's 5.95 μm . This information can generally be found by querying the technical specifications for the camera, which may be available from the manufacturer or possibly found on online forums.

Similarly, I made a function to calculate the angular field of view of an already cropped circular image, by solving the equation for θ (Code 4).

```

1  fov.theta <- function(prop_crop, full_circle_radius_px, focal_length_mm, pixel_pitch_um){
2    # Calculate number of pixels in the radius of the crop
3    px_crop <- full_circle_radius_px * prop_crop
4
5    # Calculate radius
6    theta <- 2 * asin((pixel_pitch_um * px_crop) / (2 * focal_length_mm * 1000)))
7
8    deg_theta <- round(NISTRadianT0deg(theta), 2)
9
10   return(deg_theta)
11 }

```

Code 4: R function to calculate the zenith angle from a cropped circular image. Also found in [fov_func.R](#) in the supplementary materials.

4.3 Calculating gap fraction with ImageJ

The process for calculating gap fraction is fairly simple. Basically, from the binarized [.tif](#) images you made earlier, you just count the number of pixels in the image which are sky, i.e. white in the thresholded image, then divide by the total number of pixels in the image. It's slightly more complicated when the image is a circle within a larger frame, but not much.

To calculate gap fraction:

1. Open Image
2. [File → Open](#) then select a binarized [.tif](#) image
3. [Edit → Invert](#) to invert the colours
4. [Analyze → Analyze Particles..](#)
 - (a) [Size \(pixel²\) = 0-infinity](#)

- (b) [Circularity](#) = 0-1
 - (c) [Show](#) = Nothing
 - (d) Check [Summarize](#)
5. The results should appear in a table, the gap fraction value is [%Area](#)
6. Rinse and repeat for all images

[Code 5](#) performs the same process but for a folder of images and exports a [.csv](#) spreadsheet file of the results. Remember to change the user inputs to point to where you would like images to be opened from and the spreadsheet saved to.

```

1 // User inputs
2 /////////////////
3 input_path = "/Users/username/Desktop/input/";
4 output_path = "/Users/username/Desktop/output/";
5 binarize_first = "TRUE";
6 // Only set to "FALSE" if a binarized '.tif' is used
7 algorithm = "Default";
8 /////////////////
9 // END user inputs
10
11 list = getFileList(input_path);
12
13 for (i=0; i<(list.length); i++) {
14   open(""+input_path+list[i]+");
15   if (binarize_first=="TRUE"){
16     run("8-bit");
17     setAutoThreshold(algorithm);
18     setOption("BlackBackground", false);
19     run("Convert to Mask");
20   }
21   run("Invert");
22   file_name = getInfo("image.filename");
23   run("Analyze Particles...", "summarize");
24   image_id = getImageID();
25   selectImage(image_id);
26   close();
27   roiManager("reset");
28 }
29
30 selectWindow("Summary");
31 saveAs("Results",(""+output_path+"results.csv"));
32 run("Close");

```

Code 5: ImageJ macro to calculate the gap fraction of a full image. The macro can also be found in [gap_frac_image.ijm](#).

The process is similar for a circular image such as that taken with a full frame camera and a fisheye lens, except you draw a circle selection to exclude the black uninformative bits of the image before running [Analyze Particles....](#) See [Code 6](#) for the macro.

```

1 // User inputs
2 /////////////////
3 input_path = "/Users/johngodlee/Desktop/input/";
4 output_path = "/Users/johngodlee/Desktop/output/";
5 circle_diam = 3925;
6 binarize_first = "TRUE";
7 // Only set to "FALSE" if a binarized '.tif' is used
8 algorithm = "Default";
9 /////////////////
10 // END user inputs
11
12 list = getFileList(input_path);
13
14 for (i=0; i<(list.length); i++) {
15   open(""+input_path+list[i]+");
16   if (binarize_first=="TRUE"){
17     run("8-bit");
18     setAutoThreshold("Default");
19     setOption("BlackBackground", false);
20     run("Convert to Mask");
21   }
22   run("Invert");
23   makeOval((getWidth/2) - (0.5 * circle_diam),
24             (getHeight/2) - (0.5 * circle_diam),
25             circle_diam,
26             circle_diam);
27   file_name = getInfo("image.filename");
28   run("Analyze Particles...", "summarize");
29   image_id = getImageID();
30   selectImage(image_id);
31   close();
32   roiManager("reset");
33 }
34
35 selectWindow("Summary");
36 saveAs("Results",(""+output_path+"results.csv"));
37 run("Close");

```

Code 6: ImageJ macro to calculate the gap fraction of a circular selection of an image. The macro can also be found in [gap_frac_circle.ijm](#).

The circular diameter of the image to fill `circle_diam` can be measured in ImageJ by selecting [Straight Line](#) from the toolbar then drawing a straight line across the centre of the circular image. Then select [Analyze → Measure](#) to get the length of the line in the Results table. Alternatively, use the methods presented above in section 4.2 to crop the circular image to a particular angle of view, to exclude parts of the image closer to the ground.

4.4 Calculating Leaf Area Index

This part of the guide relies mostly on code written by Hans ter Steege's HemiPhot package ([ter Steege, 2018](#)), which ports WinPhot into the R language. Winphot was originally written by Hans ter Steege in 1997, and Hemiphot can be seen as the natural progression of this software. Winphot is now obsolete and has been unmaintained since Windows XP. The code includes functions for thresholding and binarizing images, but I prefer to do this step in ImageJ prior to analysing the images in R, because I like I have more control over how the images are thresholded this way.

The workflow below assumes you are using RStudio, but you can use the scripts in any R programming environment.

1. Open RStudio
2. Open a new script ([File → New File → R Script](#))
3. Save the script in a folder above the images folder

4. Enter the following preamble into the R script

```
1 # Set working directory to location of thresholded images
2 setwd("LOCATION_OF_ANALYSIS")
3
4 # Source the functions used to calculate stuff
5 source("hemiphot.R")
```

5. Read in all the previously thresholded and binarized `.tif` images and create an empty data frame which will later be filled with canopy trait statistics like LAI and canopy openness.

```
1 # Load images from directory
2 img_file_list <- list.files("test_img", pattern = ".tif", full.names = TRUE)
3
4 img_list <- lapply(img_file_list, readTIFF)
5
6 # Create empty dataframe, 6x7 and fill it with zeroes
7 all_data = data.frame(matrix(data = 0, nrow = length(img_file_list), ncol = 7))
8 names(all_data) = c("File", "CanOpen", "LAI", "DirectAbove",
9 "DiffAbove", "DirectBelow", "DiffBelow")
10 # Fill first column with image names
11 all_data[,1] = img_file_list
```

6. Set some parameters for the location the photos are being taken. Approximate location (0.1 degrees latitude) is good enough for our purposes. Note that the values below are for somewhere in Africa and should be changed:

```
1 location.latitude      = -15
2 location.altitude      = 200
3 location.day           = 30
4 location.days          = seq(15,360,30)    # roughly the middle of each month of the year
```

7. Set some parameters for the images, cropping them to a circle and setting the threshold. Even if the images have been thresholded already, thresholding them again won't change anything. These parameters are ones I have used on my camera (Nikon D750 with Sigma 8 mm equisolid fisheye lens), so may need to be changed depending on your equipment:

```
1 location.cx <- dim(img_list[[1]])[2] / 2    # x coordinate of center
2 location.cy <- dim(img_list[[1]])[1] / 2    # y coordinate of center
3 location.cr <- fov.px(60, 8, 5.95)    # Circle radius of interest
```

10. Set some atmospheric parameters. The metrics affected by these parameters are: `DirectAbove`, `DiffAbove`, `DirectBelow` and `DiffBelow`. If these metrics are of no interest to you, just fill these parameters with random decimals between zero and one.

```
1 # Atmospheric transmissivity
2 # Normally set at 0.6, but can vary between 0.4-0.6 in the tropics
3 location.tau = 0.6
4
5 # Amount of direct light that is used as diffuse light in the Uniform Ovecast Sky (UOC)
6 location.uoc = 0.15
```

11. Run a big for loop to calculate the statistics for each photo

```

1  for(i in 1:length(img_file_list)){
2    # read file
3    image <- readTIFF(img_file_list[i])
4
5    # convert to Hemi image
6    image <- Image2Hemiphot(image)
7
8    # set circle parameters
9    image <- SetCircle(image, cx = location.cx, cy = location.cy, cr = location.cr)
10
11   gap.fractions <- CalcGapFractions(image)
12   all_data[i,2] = CalcOpenness(fractions = gap.fractions)
13
14   # calculate LAI according to Licor's LAI Analyzer
15   all_data[i,3] = CalcLAI(fractions = gap.fractions)
16
17   # Photosynthetic Photon Flux Density (PPDF, umol m-1 s-1) P
18   rad <- CalcPAR.Day(im = image,
19     lat = location.latitude, d = days,
20     tau = location.tau, uoc = location.uoc,
21     draw.tracks = F, full.day = F)
22
23   all_data[i,4] = rad[1]
24   all_data[i,5] = rad[2]
25   all_data[i,6] = rad[3]
26   all_data[i,7] = rad[4]
27 }
```

12. Finally, look at the output, which is stored in `all_data`.

There are many other functions in the source code for Hemiphot and I recommend reading through them along with the documentation to see what is right for your needs. A copy of the script above is included in [hemiphot_example.R](#).

4.5 Setting the focus to infinity

My assumption has always been that when taking hemispherical photos of forest canopies, the focus of the lens should always be set to infinity. While this knowledge is repeated throughout the literature, it is difficult to find much discussion on *why* this is the case. What follows is a series of quotes from the literature relating on the discussion of setting the lens focus to infinity.

We used a Nikon MF-16 camera and a Nikkor 8-mm fish-eye lens with TriX ASA 400 film, a red filter to increase sharpness of leaf edges, and the *focus set to infinity*.

– Englund et al. 2000

The lens was set to a small aperture and *focused on infinity* (Frazer et al. 2001; Zhu et al. 2003)

– Hu and Zhu 2009

Exposure settings were selected to obtain the best contrast between foliage and sky and making the last one appear white (cloudy sky offers the best condition in this context). The camera was used in automatic mode using the parameters fixed in FISHEYE1 lens mode (*focus set to infinity*, widest zoom, metering center-weighted), and the shutter speed was varied automatically by the camera.

– Paletto and Tosi 2009

This next quote comes from a paper which many others reference when describing proper protocol for taking hemispherical photos:

Unlike the Nikon F, the digital camera did not allow full manual control of both the shutter speed and lens aperture. We therefore *set the autofocus, exposure mode, and f-stop of the digital camera to infinity*, aperture priority (shutter speed is set automatically by the camera), and f/2.6, respectively.

– Frazer et al. 2001

While this paper is one of the most respected guides on hemispherical photography, it references an unconventional source when describing the practice of setting the focus to infinity. A downloadable tutorial series on taking high-quality digital photos, called 123di (). Most of this resource is behind a paywall, so I can't read the applicable bit. Importantly however, this paper references *why* the authors set the focus to infinity, because the depth of field is practically infinite under these conditions.

Photographic images were recorded using a Sigma 8 mm f/4 fisheye lens (Sigma Corporation, Tokyo, Japan) at the highest possible resolution (3040-2008 pixels) with highest ISO setting (ISO 200). Moreover, *the focus ring was set to infinity when using the fisheye lens, as depth of field is practically infinite and focusing was not required (Bockaert, 2004)*.

– Jonckheere et al. 2005

4.6 Zenith angle for LAI calculations

This is the only reference I have found in the literature which discusses why the angle of view should be limited from a full 180 degree hemispherical image. It can be supposed that below 60 degrees, in most woodland canopies, variation in tree canopy density does little to affect sunlight penetration, due to the greater depth of canopy at these angles. Additionally, for many fisheye lenses, angles below 60 degrees introduce significant visual artefacts and distortion, which may lead to biased results.

In the range from zero to about 60 zenith angle the canopy effects are more significant. This is the useful working range of most hemispherical photograph data.

– Jupp et al. (2008)

5 Alternatives to hemispherical photography

There are a number of other ways to measure the same canopy traits for which hemispherical photography is commonly used, such as gap fraction. Additionally, some tree canopy traits are best measured with something other than hemispherical photography, or are impossible to measure with hemispherical photography.

5.1 LiCOR LAI-2000

A device operating on a similar principle to hemispherical photography, which processes an image and provides a percentage canopy cover estimate. Measurements are paired with replicates taken simultaneously under an open sky to normalise changing light conditions.

5.2 LiDAR

Terrestrial or airborne Light-Detection And Ranging equipment can be used to estimate canopy cover and the vertical distribution of canopy cover. Equipment is expensive and processing is complex, but this method is currently the most precise and accurate (Seidel et al., 2011).

5.3 Periscope densitometer

A simple periscope tube is marked with a cross. The periscope is pointed upwards and the operator records the presence or absence of plant material in the centre of the cross. A dense grid of measurements is taken which can then be converted to a percentage canopy cover.

5.4 Reflective dish densitometer

A concave mirror surface is divided into a grid to facilitate the estimation of the percentage of sky covered by plant material. This method, while previously commonplace in ecology and forestry, is now not recommended, as it has been shown that large recording biases can occur, even when measurements are taken by a trained operator.

5.5 Leaf collection

A leaf trap of known dimensions is left on the forest floor for a known period of time to capture falling leaves. The area of these individual leaves is measured to estimate the total leaf area of the canopy, assuming a constant rate of leaf fall.

5.6 Manual harvesting

Branches within a known area of canopy are harvested and the leaves removed. The area of these individual leaves is measured to directly measure the total leaf area of the canopy.

References

- Brusa, A. and Bunker, D. E. (2014), ‘Increasing the precision of canopy closure estimates from hemispherical photography: Blue channel analysis and under-exposure’, *Agricultural and Forest Meteorology* **195-196**, 102–107.
- Demarez, V., Duthoit, S., Baret, F., Weiss, M. and Dedieu, G. (2008), ‘Estimation of leaf area and clumping indexes of crops with hemispherical photographs’, *Agricultural and Forest Meteorology* **148**(4), 644–655.
- Englund, S. R., O’Brien, J. J. and Clark, D. B. (2000), ‘Evaluation of digital and film hemispherical photography and spherical densiometry for measuring forest light environments’, *Canadian Journal of Forest Research* **30**, 1999–2005.
- Frazer, G. W., Fournier, R. A., Trofymow, J. and Hall, R. J. (2001), ‘A comparison of digital and film fish-eye photography for analysis of forest canopy structure and gap light transmission’, *Agricultural and Forest Meteorology* **109**(4), 249–263.
- Hu, L. and Zhu, J. (2009), ‘Determination of the tridimensional shape of canopy gaps using two hemispherical photographs’, *Agricultural and Forest Meteorology* **149**(5), 862–872.
- Huang, L. and Wang, M. J. (1995), ‘Image thresholding by minimizing the measures of fuzziness’, *Pattern Recognition* **28**(1), 41–51.
- Jonckheere, I., Nackaerts, K., Muys, B. and Coppin, P. (2005), ‘Assessment of automatic gap fraction estimation of forests from digital hemispherical photography’, *Agricultural and Forest Meteorology* **132**(1-2), 96–114.
- Jupp, D. L. B., Culvenor, D. S., Lovell, J. L., Newnham, G. J., Strahler, A. H. and Woodcock, C. E. (2008), ‘Estimating forest lai profiles and structural parameters using a ground-based laser called ‘echidna’’, *Tree Physiology* **29**, 171–181.
- Macfarlane, C., Ryu, Y., Ogden, G. N. and Sonnentag, O. (2014), ‘Digital canopy photography: exposed and in the raw’, *Agricultural and Forest Meteorology* **197**, 254–253.
- Paletto, A. and Tosi, V. (2009), ‘Forest canopy cover and canopy closure: comparison of assessment techniques’, *European Journal of Forest Research* **128**(3), 265–272.
- Schneider, C. A., Rasband, W. S. and Eliceiri, K. W. (2012), ‘NIH image to ImageJ: 25 years of image analysis’, *Nature Methods* **9**(7), 671–675.

Seidel, D., Fleck, S. and Leuschner, C. (2011), ‘Review of ground-based methods to measure the distribution of biomass in forest canopies’, *Annals of Forest Science* **68**, 225–244.

ter Steege, H. (2018), *Hemiphot.R: Free R scripts to analyse hemispherical photographs for canopy openness, leaf area index and photosynthetic active radiation under forest canopies*, Leiden, The Netherlands. Unpublished report.