

Hemispherical photography for the estimation of woodand canopy traits

John L. Godlee

Contents

1	Introduction	2
2	What is hemispherical photography?	2
3	Measurements of woodland tree canopies	2
3.1	Gap fraction	2
3.2	Direct/indirect/global site factor	2
3.3	Leaf/Plant Area Index (LAI/PAI)	2
3.4	Canopy cover	2
4	Hemispherical photos in the field	2
4.1	Capturing informative photos	2
4.2	Sampling strategy	4
5	Processing hemispherical photographs	4
5.1	Creating black and white thresholded images manually	5
5.1.1	A note on .tif files	6
5.2	Calculating gap fraction with ImageJ	6
5.3	Cropping a circular image	8
5.4	Calculating Leaf Area Index	9
5.5	Setting the focus to infinity	11
5.6	Zenith angle for LAI calculations	12
6	Alternatives to hemispherical photography	12
6.1	LiCOR LAI-2000	12
6.2	LiDAR	12
6.3	Periscope densitometer	12
6.4	Reflective dish densitometer	12
6.5	Leaf collection	12
6.6	Manual harvesting	12

1 Introduction

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 conducted by the authors, along with scattered references found throughout the scientific and practitioner literature. This guide is an attempt to bring those ideas into a cohesive teaching 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.

2 What is hemispherical photography?

Hemispherical photography is a technique used to estimate plant canopy geometry, 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, but there are many creative applications of this technique. For example, hemispherical photography may be used to estimate the ground cover of agricultural crops, by pointing the camera downwards instead of upwards.

3 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 or non-plant material. 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
- Direct/indirect/global site factor
- Leaf/Plant Area Index (LAI/PAI)
- Canopy cover

3.1 Gap fraction

3.2 Direct/indirect/global site factor

3.3 Leaf/Plant Area Index (LAI/PAI)

3.4 Canopy cover

4 Hemispherical photos in the field

4.1 Capturing informative photos

Taking hemispherical photos under field conditions requires the balancing of many different requirements. Images must be properly exposed and without visual aberrations which would otherwise make analysis difficult. Confident use of the many settings available on a modern DSLR camera is necessary to adjust for changing environmental conditions to gather images which are consistent, which makes further analysis easier and the results more robust.

Some common issues:

- 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.
- Blooming - An oversaturated part of the image bleeds over into other pixels of the image sensor during image capture. This is a very common problem in hemispherical photography, requiring careful adjustment of the image exposure to limit its impact.
- Obstructions between camera and tree canopy - Herbaceous understorey material, manmade objects, people etc. can all contribute to an inaccurate estimate of the amount of plant material in the frame.

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

Below is a checklist for taking hemispherical photographs in the field:

- Take photos under a uniformly overcast sky, 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. At high latitudes you will have more time 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, whichever is higher. I usually find that adjusting the legs of the tripod is easiest, but every tripod is different.
- 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)
 - 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 press the button
 - Take all photos in landscape dimensions, never portrait
- Make sure everybody ducks down below the camera when the image is being taken!
- Make sure there is a battery and you have a spare battery.

- Make sure there is an SD card in the camera, and take a spare.
- 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 for camera
- Lens cleaning solution and lens cloth
- Tripod
- Fully charged battery 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

4.2 Sampling strategy

In order to estimate the mean value of various woodland canopy traits over an area of woodland like a survey plot requires multiple photos to 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 woodland plot will somewhat determine the spatial pattern of photos, circular plots may require a different layout to rectangular plots and linear transects, for example.

Diagram of common plot layouts and sampling strategy

A grid sample strategy works well for woodland survey plots. Photo locations should be spaced far enough apart so that no two scans sample the same patch of canopy.

5 Processing hemispherical photographs

Many proprietary options for estimating LAI etc. e.g. WinPhot, CAN-EYE. I think manual processing is more flexible and reproducible.

5.1 Creating black and white thresholded images manually

1. Open ImageJ
2. File → Open, then select an image to process
3. Visually inspect the image to see that there isn't massive amounts of lens flare. See Figure x for an example of lens flare. If you have lots of lens flare, the photo should be thrown out.
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 branches and white sky
7. Repeat for all images

The above process can be automated with a macro (Code 1), but this assumes removes the ability to manually threshold each image, relying instead on a number of binarization algorithms provided in ImageJ. Macros should be saved as a .ijm file and called within ImageJ with Plugins → Macros → Run... The macro can also be found in `binarize.ijm` in the supplementary material.

```
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("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.

I find that the **Huang** 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 has very little blue in it, while the sky generally has much more, so one can threshold using only the blue channel of the image (Code 2), also found in `binarize_blue_channel.ijm`:

```
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

5.1.1 A note on .tif files

5.2 Calculating gap fraction with ImageJ

Gap fraction is the percentage of the image covered by sky.

This process is fairly simple. Basically, from the binarized '.tif' images you made earlier, you just count the number of pixels which are sky in the image, 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 do this manually:

1. Open ImageJ
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

This macro 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. The macro can also be found in `gap_frac_image.ijm`:

```

1 // User inputs
2 ///////////////////////////////////////////////////
3 input_path = "/Users/johngodlee/Desktop/input/";
4 output_path = "/Users/johngodlee/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 3: ImageJ macro to calculate the gap fraction of a full image

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...** The macro looks like this and can also be found in `gap_frac_circle.ijm`:

```

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");
38 }
39
40 saveAs("Results", ""+output_path+"gap_frac_results.xls");

```

Code 4: ImageJ macro to calculate the gap fraction of a circular selection of an image

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, read below if you would like to crop the circular image to a particular angle of view, to exclude parts of the image closer to the ground.

5.3 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). Fisheye lenses have different projection functions which map the curved image onto a flat surface, much like a map projection. Here is a list of common projection functions for different lenses:

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

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

fov_diagram.png

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 ?? (Code 5).

```

1 fov.px <- function(deg_theta, focal_length_mm, pixel_pitch_um){
2   require(NISTunits)
3
4   # Convert degrees of theta to radians
5   rads_theta <- NISTdegToRadian(deg_theta)
6
7   # Calculate radius of circle drawn by angle of view
8   # (rads_theta and max_rads_theta) in mm projected onto the sensor plane
9   R <- 2 * focal_length_mm * sin(rads_theta / 2)
10
11  # Calculate the px per mm on the sensor, i.e. the pixel pitch
12  sensor_px_per_mm_flat <- 1/pixel_pitch_um * 1000
13
14  # Multiply the mm radius of the desired circle by the
15  # number of pixels per mm on the sensor, to get the number
16  # of pixels radius of the desired circle
17  pixels_for_theta <- R * sensor_px_per_mm_flat
18
19  # Print result
20  print(paste("Radius of circle:", round(pixels_for_theta, digits = 0), "px"))
21 }
22
23 # Run the function
24 fov.px(deg_theta = 90, focal_length_mm = 8, pixel_pitch_um = 5.95)

```

Code 5: R function to calculate the pixel radius of a circle with a given zenith angle

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 degrees ' \hat{I}_c ' of an already cropped circular image, by solving the equation for ' \hat{I}_c ' (Code 6). Both functions can also be found in fov_func.R.

```

1 fov.theta <- function(prop_crop, full_circle_radius_px, focal_length_mm, pixel_pitch_um){
2   require(NISTunits)
3
4   # Calculate the number of pixels in the radius of the crop
5   px_crop <- full_circle_radius_px * prop_crop

```



```

6
7 # Calculate the radius of the
8 theta <- 2 * asin(((pixel_pitch_um * px_crop) / (2 * focal_length_mm * 1000)))
9
10 deg_theta <- NISTradianT0deg(theta)
11
12 print(paste("Angle of view: ", round(deg_theta, digits = 2), "A", sep = ""))
13 }

```

Code 6: R function to calculate the zenith angle from a cropped circular image.

These values can be used in the ImageJ macro `gap_frac_circle.ijm` to alter `circle_diam` to make a circular selection of the image of a certain view angle.

5.4 Calculating Leaf Area Index

This part relies mostly on code written by [Hans ter Steege's HemiPhot package](#), which ports WinPhot into the R language. **More about the history of Winphot.** 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 feel like I have more power over how the images are thresholded this way.

The first thing to do is create an image ('white_image.jpg') in Photoshop like the one below, which has a white circle and black background the same size as my hemispherical images. This is so you can find the pixel diameter of the image for the calculations. Alternatively, just use the pixel diameter used in the ImageJ circular cropping macros:

white_image.jpg)

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")
6
7 # Packages
8 library(jpeg)

```

5. Add `white_image.jpg` to the same folder where the thresholded images are found
6. 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 # List all images in the directory
2 all_images <- list.files("img/", pattern = ".tif")
3
4 # How many images
5 img_length = length(all_images)
6
7 # Create empty dataframe, 6x7 and fill it with zeroes
8 all_data = data.frame(matrix(data = 0, nrow = img_length, ncol = 7))
9 names(all_data) = c("File", "CanOpen", "LAI", "DirectAbove",
10 "DiffAbove", "DirectBelow", "DiffBelow")
11 # Fill first column with image names
12 all_data[,1] = all_images

```

7. Read in the reference image ('white_img.jpg') as a matrix of pixel values:

```
1 white_img <- readJPEG("img/white_image.jpg", native = F)
```

8. 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
```

9. 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 ## Image parameters
2
3 ### Drawing circles and identifying the image centre point
4 hemi_dim <- dim(white_img)
5 radius <- max(rowSums(white_img[,1] > 0.4) / 2)
6
7 ### determine using a single image and fill in here for batch processing
8 location.cx = (hemi_dim[2] / 2) # x coordinate of center of image
9 location.cy = (hemi_dim[1] / 2) # y coordinate of center image
10 location.cr = radius # radius of circle
11 location.threshold = 0.42
```

10. Set some atmospheric parameters. I've loosely estimated these for a location in Angola where I work, but by no means is it scientific. I would not have much confidence in the statistics generated using these parameters, namely DirectAbove, DiffAbove, DirectBelow and DiffBelow.

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

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

```
1 for(i in 1:img_length){
2   ## read file
3   image <- readJPEG(paste("test_img/", all_images[i], sep = ""), native = F)
4
5   ## conver 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  ## select blue channel
12  image <- SelectRGB(image, "B")
13
14  #threshold
15  image <- ThresholdImage(im = image, th = location.threshold, draw.image = F)
16
17  # canopy openness
18  gap.fractions <- CalcGapFractions(image)
19  all_data[i,2] = CalcOpenness(fractions = gap.fractions)
20
21  ## calculate LAI according to Licor's LAI Analyzer
22  all_data[i,3] = CalcLAI(fractions = gap.fractions)
23
24  ## Photosynthetic Photon Flux Density (PPDF, umol m-1 s-1) P
25  rad <- CalcPAR.Day(im = image,
26    lat = location.latitude, d = location.days,
```

```

27     tau = location.tau, uoc = location.uoc,
28         draw.tracks = F, full.day = F)
29     all_data[i,4] = rad[1]
30     all_data[i,5] = rad[2]
31     all_data[i,6] = rad[3]
32     all_data[i,7] = rad[4]
33 }

```

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

```

1 all_data

```

There are many other functions in `hemiphot.R` and I recommend reading through them along with the documentation to see what is right for your needs.

5.5 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. However, I'm not sure where I got this assumption from, or whether it's even true, I've just always believed it. So to set my mind at ease and back up my reasons for telling others to set the focus to infinity I trawled through a few papers. This is just a rundown of what those papers were with quotes of what they said regarding the issue.

This first paper only says that non-DSLR cameras should have the focus set to infinity, either that or it's confusingly written.

> All camera internal software filters (e.g. sharpen the picture) need to be turned off. For non-DSLR cameras the "Fish-Eye setting" means that the zoom is fixed at the widest angle and *focus is set to infinity*. > > – Melrose et al. 2012

These next few simply say that they did set it to infinity, but don't say why..

> 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 et al. 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 & 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

This one references a really dodgy looking old downloadable tutorial series on taking good digital photos, called 123di. Most of it is behind a paywall though, so I can't read the applicable bit. But at least it sort of says why they 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 a— 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

References

Melrose et al. 2012 - Manual on methods and criteria for harmonized sampling, assessment, monitoring and analysis of the effects of air pollution on forests - Part XVII Leaf Area Measurements

Englund et al. 2000 - Evaluation of digital and film hemispherical photography and spherical densiometry for measuring forest light environments

Hu et al. 2009 - Estimation of canopy gap size and gap shape using a hemispherical photograph

Frazer et al. 2001 - A comparison of digital and film fisheye photography for analysis of forest canopy structure and gap light transmission

Jonckheere et al. 2005 - Assessment of automatic gap fraction estimation of forests from digital hemispherical photography

Paletto & Tosi 2009 - Forest canopy cover and canopy closure: comparison of assessment techniques

5.6 Zenith angle for LAI calculations

This is the only reference I have found so far which specifically states *why* the angle of view should be limited from a full 180 degree hemispherical image.

> 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. 2009

6 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.

6.1 LiCOR LAI-2000

6.2 LiDAR

6.3 Periscope densitometer

6.4 Reflective dish densitometer

6.5 Leaf collection

6.6 Manual harvesting