assignments-canopy

Francesco Chianucci

The whole game of canopy photography

1. Field collection of hemispherical (DHP) or cover (DCP) images
    - Sampling design
    - Optimal sky conditions
    - Setting optimal exposure
2. Download and rename data
3. (for RAW) develop RAW into JPEG imagery using bRaw
4. For DHP: process images with hemispheR
5. For DCP: process images with coveR

Field collection

It is not the focus of this assignment. You can find detailed info in Chianucci 2020

The general recommendations are (Chianucci 2020):

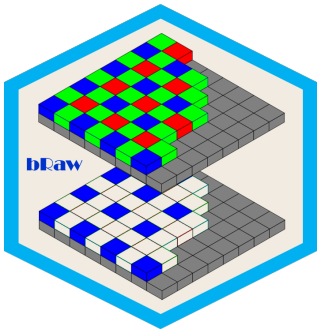
- Acquire images in diffuse sky conditions.
- Underexpose by one or two stops in A-priority mode with small aperture (large f-number).
- Shoot in raw mode (when available from the camera).
- Check the shooting results in the field using the camera histogram.

Pre-processing

Once collected, downloaded and renamed images, a first common analysis step in R is to convert RAW to JPEG images (if you have collected RAW images)

This can be accomplished with the bRaw package

Convert RAW imagery



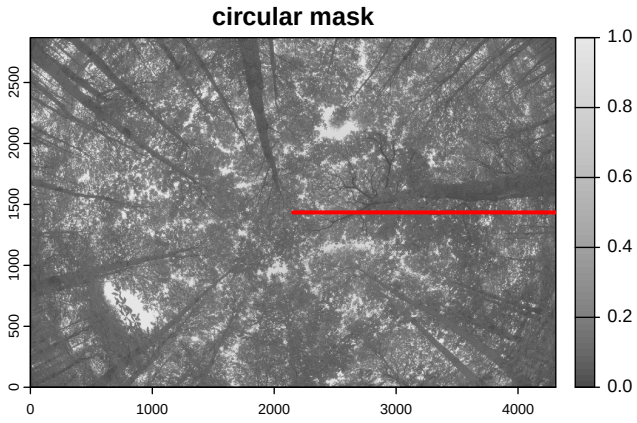
Basically, bRaw to convert RAW to a single-channel blue, with a linear contrast stretch.

You need also to copy the dcraw program (see readme)

The functions is `raw_blue()`

```
# devtools::install_gitlab('fchianucci/bRaw')
library(bRaw)

filename<-system.file('extdata/531539_AGO12_531.NEF',package='bRaw')
raw_blue(filename,
         display=T)
```



- It automatically export images in a “bRaw” subfolder
- For circular fisheye images, you can also set a circular mask parameters.

Fisheye photography

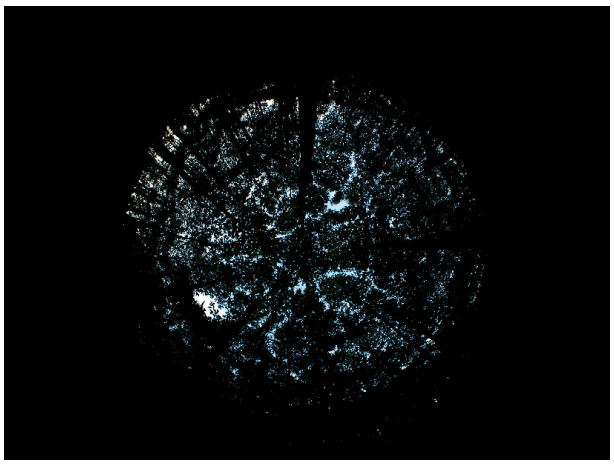
The basic steps of processing fisheye imagery is:

1. Import an image channel (or apply channel mixing) and applies a circular mask
2. Create a binary image of sky (1) and canopy (0) pixels
3. Divide the hemisphere in circular annuli (rings) and radial sectors (segments)
4. Apply theoretical formulas to convert LAI from angular gap fraction.

They are accomplished with the hemispheR package

Next section illustrates the basic steps using an example image in the package:

```
library(hemispheR)
c.im<-system.file('extdata/circular_coolpix4500+FC-E8_chestnut.jpg',package='hemispheR')
terra::plot(terra::rast(c.im))
```



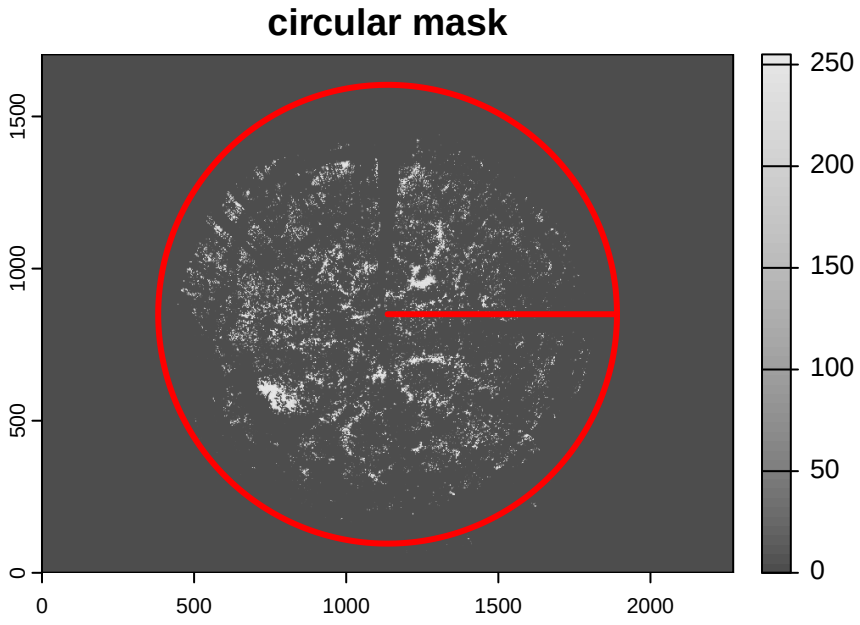
Fisheye photography: import

The basic steps of processing fisheye imagery is:

1. **Import an image channel (or apply channel mixing) and applies a circular mask**
2. Create a binary image of sky (1) and canopy (0) pixels
3. Divide the hemisphere in circular annuli (rings) and radial sectors (segments)
4. Apply theoretical formulas to convert LAI from angular gap fraction.

```
#install.packages('hemispheR')
library(hemispheR)

c.im |>
  import_fisheye(circ.mask=list(xc=1136,yc=850,rc=754),
                 channel='B',
                 gamma=2.2,
                 display=TRUE)
```



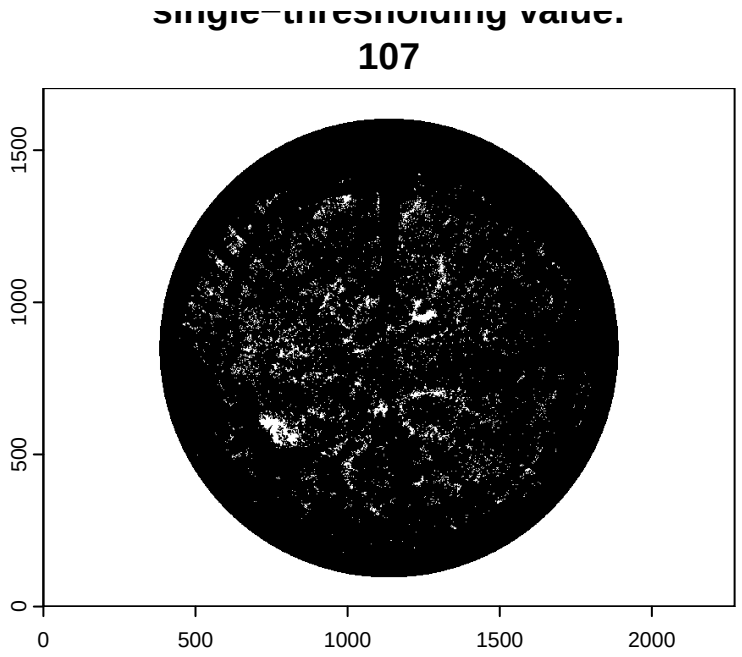
- Look at the `circ.mask` argument
- What is `channel` and `gamma` ?

Fisheye photography: binarize

The basic steps of processing fisheye imagery is:

1. Import an image channel (or apply channel mixing) and applies a circular mask
2. **Create a binary image of sky (1) and canopy (0) pixels**
3. Divide the hemisphere in circular annuli (rings) and radial sectors (segments)
4. Apply theoretical formulas to convert LAI from angular gap fraction.

```
c.im |>
  import_fisheye(circ.mask=list(xc=1136,yc=850,rc=754),
                 channel='B',
                 gamma=2.2,
                 message=F) |>
  binarize_fisheye(export=FALSE,
                   display=TRUE)
```



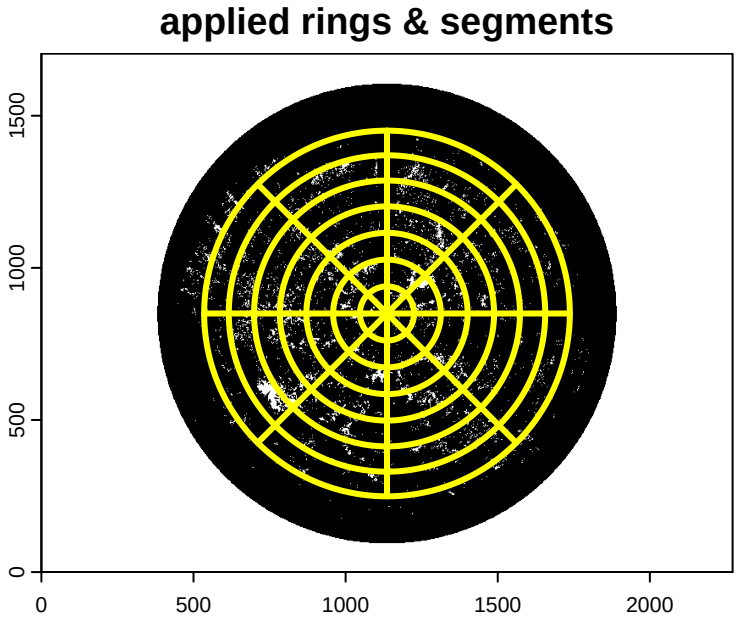
- export=T store the single channel binary image in a 'results' sub.folder

Fisheye photography: get gap fraction

The basic steps of processing fisheye imagery is:

1. Import an image channel (or apply channel mixing) and applies a circular mask
2. Create a binary image of sky (1) and canopy (0) pixels
3. **Divide the hemisphere in circular annuli (rings) and radial sectors (segments)**
4. Apply theoretical formulas to convert LAI from angular gap fraction.

```
gapfrac <- c.im |>
  import_fisheye(circ.mask=list(xc=1136,yc=850,rc=754)) |>
  binarize_fisheye() |>
  gapfrac_fisheye(maxVZA = 90,
                  lens = "FC-E8",
                  startVZA = 0,
                  endVZA = 70,
                  nrings = 7,
                  nseg = 8,
                  message = F,
                  display = T)
```



```
                                      id ring      GF0_45    GF45_90
1 circular_coolpix4500+FC-E8_chestnut.jpg    5 0.120705957 0.04372990
2 circular_coolpix4500+FC-E8_chestnut.jpg   15 0.143875567 0.19467102
3 circular_coolpix4500+FC-E8_chestnut.jpg   25 0.080959520 0.08835657
4 circular_coolpix4500+FC-E8_chestnut.jpg   35 0.109686879 0.05920023
5 circular_coolpix4500+FC-E8_chestnut.jpg   45 0.054930485 0.03136272
6 circular_coolpix4500+FC-E8_chestnut.jpg   55 0.059158535 0.01932940
7 circular_coolpix4500+FC-E8_chestnut.jpg   65 0.005935217 0.01884359
    GF90_135
1 0.07185629
2 0.04266580
3 0.07124699
4 0.04167061
5 0.03292041
6 0.01497360
7 0.02514068
```

- Important: in this step you define the lens projection distortion (`lens`)

Fisheye photography: get canopy attributes

The basic steps of processing fisheye imagery is:

1. Import an image channel (or apply channel mixing) and applies a circular mask
2. Create a binary image of sky (1) and canopy (0) pixels
3. Divide the hemisphere in circular annuli (rings) and radial sectors (segments)
4. **Apply theoretical formulas to convert LAI from angular gap fraction.**

```
results <- c.im |>
  import_fisheye(circ.mask=list(xc=1136,yc=850,rc=754)) |>
  binarize_fisheye() |>
  gapfrac_fisheye(lens = "FC-E8") |>
  canopy_fisheye()

results
```

```
# A tibble: 1 x 20
  id         Le     L    LX  LXG1  LXG2  DIFN MTA.ell     x VZA   rings azimuths
  <chr>   <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>   <dbl> <dbl> <chr> <int>    <int>
1 circul~  3.82  4.02  0.95  0.84  0.76  6.42      39  1.99 5_15~     7        8
# i 8 more variables: mask <chr>, lens <chr>, channel <chr>, stretch <chr>,
#   gamma <chr>, zonal <chr>, method <chr>, thd <chr>
```

Cover photography

The basic steps of processing cover imagery using coveR are:

1. Import an image channel
2. Create a binary image of sky (1) and canopy (0) pixels
3. Classify gaps based on their size
4. Apply theoretical formulas to convert LAI from angular gap fraction.

- The first two (1-2) steps are comparable with that of fisheye images (except for circular mask)
- There are two version of coveR packages. Look at the next section!

coveR: installation

- The `coveR` package is available only as development version in Gitlab:

```
# install.packages("devtools")
devtools::install_gitlab("fchianucci/coveR")
```

- The `coveR` package has a native EXIF reading functionality, which is useful for continuous camera.
- If no EXIF functionality is need, you can alternatively install the `coveR2`, which is available in CRAN:

```
install.packages('coveR2')
```

coveR: get ready!

Next section illustrates the basic steps using an example image in the package:

coveR: workflow

The basic steps of processing cover imagery using coveR are:

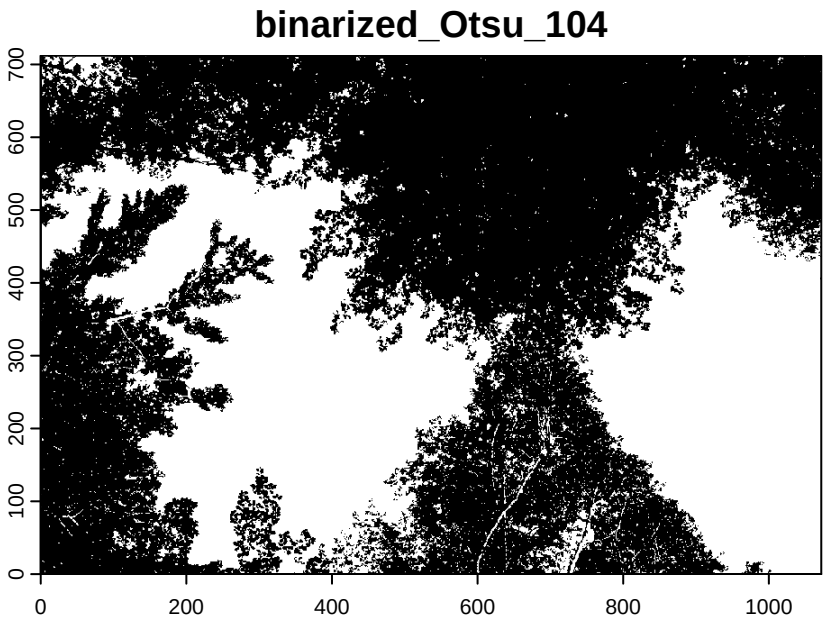
1. Import an image channel
2. Create a binary image of sky (1) and canopy (0) pixels
3. Classify gaps based on their size
4. Apply theoretical formulas to convert LAI from angular gap fraction

Both the `coveR` and `coveR2` packages feature a single function, respectively `coveR::coveR()` or `coveR2::coveR2()` which performs sequentially these steps, so it does the job in a single function!

coveR

```
library(coveR2)
results <- coveR2(
      filename=image,
      channel=3,
      thdmethod='Otsu',
      gapmethod='macfarlane',
      thd=1.3/100,
```

```
        k=0.85,
        display=T,
        export.image=F)
```




```
results
```

```
# A tibble: 1 x 12
  id          FC    CC     CP    Le     L    CI     k imgchannel gapmethod
  <chr>    <dbl> <dbl>  <dbl> <dbl> <dbl> <dbl> <dbl>      <dbl> <chr>
1 IMG1.JPG 0.565 0.604 0.0646 0.980  1.95 0.503  0.85          3 macfarlane
# i 2 more variables: imgmethod <chr>, thd <dbl>
```

- what are `thdmethod`, `gapmethod` and `k` parameters?
- you can decide to export image, and display figures.

Your assignments

As you have seen the most of canopy photography, you can try on your own with your data!