



## **CAN\_EYE V6.4.91 user manual**

Marie Weiss, Frédéric Baret

### **► To cite this version:**

| Marie Weiss, Frédéric Baret. CAN\_EYE V6.4.91 user manual. 2017, 56 p. hal-02788819

**HAL Id: hal-02788819**

**<https://hal.inrae.fr/hal-02788819>**

Submitted on 20 Apr 2021

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



## **CAN\_EYE V6.4.91 USER MANUAL**

**Updated October, 10<sup>th</sup> 2017**

**Contributing Authors: M. Weiss, F. Baret  
UMT CAPTE / UMR1114 EMMAH, INRA**

**Contact: [can\\_eye-paca@inra.fr](mailto:can_eye-paca@inra.fr)**

# CONTENT

---

<b>CAN_EYE V6.4.6 USER MANUAL</b>	<b>1</b>
1.1. CAN_EYE specific features	6
1.2. CAN_EYE basic principles	7
1.3. Hardware and software requirements	10
1.4. Copyright and limitations of responsibility	10
1.5. Upgrading the version	10
1.6. Bug report	10
2. INSTALLATION	11
3. USING CAN_EYE STEP BY STEP	15
3.1. Choosing the image type to be processed	15
DHP: Digital Hemispherical Photographs	16
DP57: Digital Photographs acquired with a camera inclined at 57.5°	17
3.2. Defining the processing parameters	19
Processing parameters for DHP	19
Processing parameters for digital images acquired at 57° (DP57)	23
Processing parameters for images acquired at nadir	24
3.3. SELECTING PERTINENT IMAGES	26
3.4. MASKING IMAGES	28
GAMMA: Gamma Correction	29
SELECT: Selection of one image to define a mask	29
APPLY ALL: Apply the last mask to all images	30
UNAPPLY ALL: Delete the last mask applied to all images	30
UNDO: Undo last mask	30
RESET: Reset all the masks	30
DONE: End the masking	30
3.5. CHOICE OF THE INTERACTIVE CLASSIFICATION METHOD	31
No mixed pixels (2 classes)	31
2 Classes + Mixed Pixels	31
3.6. Classifying the images	32
Toolbar MENU (top left)	32
Class MENU (bottom)	32
Main MENU (right)	34
4. CAN_EYE OUTPUT DESCRIPTION	35

4.1. Definitions and theoretical background.....	35
Introduction .....	35
Modeling the Gap Fraction .....	35
Modeling the leaf inclination distribution function $g(l, \theta_l, \varphi_l)$ .....	37
Estimating leaf area index and leaf inclination from gap fraction measurements .....	37
Cover fraction computation.....	40
FAPAR computation .....	40
4.2. Description of CAN_EYE output directory content.....	41
Hemispherical images .....	41
Images acquired at $57^\circ$ .....	45
Images acquired at Nadir.....	45
5. THE “SUMMARY” MODULE.....	46
Digital Hemispherical Images (DHP) .....	46
Images acquired at $57.5^\circ$ (DP57) .....	47
Images acquired at Nadir (DP0).....	47
6. DHP “CALIBRATION” MODULE .....	48
6.1. System Definition .....	49
6.2. Optical center characterization .....	50
6.3. Projection function characterization.....	52
7. REFERENCES.....	55

## List of Figures

---

Figure 1. Overview of the CAN_EYE processing of a series of images .....	9
Figure 2. CAN_EYE main menu .....	15
Figure 3. Example of the « SugarBeet » directory containing a series of 12 images (.JPG) corresponding to one ESU and to be processed concurrently .....	16
Figure 4. CAN_EYE Menu for digital hemispherical photographs (fish-eye lens) .....	17
Figure 5. CAN_EYE Menu for digital photographs acquired with a camera inclined at 57.5 .....	18
Figure 6. CAN_EYE Menu for digital photographs acquired with the camera at the vertical (nadir viewing) .....	19
Figure 7. Processing parameter window for hemispherical images .....	21
Figure 8. Defining calibration parameters in the Processing parameter window .....	22
Figure 9. Creating projection function and optical center characteristics .....	22
Figure 10. Determining the default projection function parameter of two different systems. On the left: fish-eye length with a large field of view (around 60°). On the right fish-eye length with a large field of view (around 60°). The diagonal is shown in red, <b>LD</b> is the length of the portion shown in blue. ....	23
Figure 11. Processing parameter window for images acquired at 57.5° .....	24
Figure 12. Processing parameter window for images acquired at nadir (0°) .....	26
Figure 13. Selecting pertinent images .....	27
Figure 14. Main masking window: the button name appearing in light gray indicate that the functionalities are not available at this step of the processing. ....	28
Figure 15. Gamma correction value (masking step) .....	29
Figure 16. Secondary masking window (after having selected the image to be masked). The mask appears in black bordered by red color. ....	29
Figure 17. Applying a mask to all images. The same masked areas appear in black on all images.....	30
Figure 18. Choice of the interactive classification method.....	31
Figure 19. Classification window.....	32
Figure 20. Choice of a color for a class.....	32
Figure 21. Using vegetation color indices to pre-classify the vegetation (No mixed pixel classification method: classified vegetation pixels appear in light green. The other pixels are considered to belong to the soil). ....	34
Figure 22. Example of classification results .....	42
Figure 23. Example of average gap fraction polar plot.....	43
Figure 24. Figure showing the results for the computed clumping factor (CAN-EYE version 5 in green and Version 6 in red). Model is shown in dashed lines while measured value (extracted from the binary images) is shown in symbols.....	43
Figure 25. Results of PAI and ALA estimation with CAN_EYE. RMSE value between the mono-directional gap fraction computed from the images and the closest one found in the LUT (red line). Green line shows the corresponding value. ....	44
Figure 26. Measured (green) and modelled gap fraction when considering (red) or not (dark grey) the clumping factor. ....	44
Figure 27. CAN_EYE summary menu .....	46
Figure 28. CAN_EYE DHP calibration menu .....	48
Figure 29. Example of the “Start “ sheet of the calibration excel file.....	49
Figure 30. Image coordinate system.....	50

Figure 31. Illustration of the holes drilled in the fish-eye cap. The red arrow indicates the rotation of the cap.....	51
Figure 32. A series of images taken for several positions of the fish-eye cap. In this case, three holes were considered. ....	51
Figure 33 : Example of an excel file sheet to be filled to determine the optical center of a system.....	51
Figure 34. Example of a projection function sheet for the calibration excel file. ....	52
Figure 35. Experimental design scheme. ....	53
Figure 36. Example of an image of the experimental design taken with the hemispherical camera and used for the calibration of the projection function. The horizontal dotted yellow line corresponds to the diameter of the image passing through the optical center (defined by its coordinates as measured previously). It must be aligned as much as possible with the 3 rulers since the camera should be aligned thanks to the front nail and background line. ....	54
Figure 37. Left: example of a CAN_EYE optical center calibration output, showing the fitting of the circles to the holes positions in the case of three holes. The actual optical center is shown by the red cross. Right: determination of the projection function. The different colored points correspond to the 3 rulers and the 3 lines to the 3 derived models (polynomial of order 1,2,or 3). ....	54



## 1. INTRODUCTION

CAN\_EYE is a free software developed at the EMMAH laboratory (Mediterranean environment and agro-hydro system modelling) in the French National Institute of Agricultural Research (INRA). The authors remind that this software is a didactic product made only for pedagogic uses. It is protected in France by Intellectual Property Regulations and abroad by international agreements on the copyright.

It can be downloaded at [http://www6.paca.inra.fr/can\\_eye](http://www6.paca.inra.fr/can_eye). For any information, question or bug report, please contact [can\\_eye-paca@inra.fr](mailto:can_eye-paca@inra.fr)

### 1.1. CAN\_EYE specific features

CAN\_EYE is an imaging software (Operating system: only Windows) used to extract the following canopy structure characteristics from true color images (either acquired with a fish-eye or with a classic objective)

- \* PAI : Plant Area Index
- \* ALA : Average Leaf inclination Angle
- \* FAPAR : Fraction of Absorbed Photosynthetically Active Radiation
- \* FCOVER : Vegetation cover fraction
- \* Bidirectional & mono-directional gap fraction

CAN\_EYE is designed to process several images corresponding to a given plot at once with optimal performances. The images can be either RGB images or binary. It can be used to process:

1. DHP: Digital hemispherical photographs (derivation of PAI, ALA, FAPAR, FCOVER and gap fraction) acquired with a fish-eye+ camera system
2. DP57: Digital images acquired with a classic camera inclined at 57.5° from the vertical (measurement of PAI, and Gap fraction at 57.5°)
3. DP0: Digital Images acquired at nadir (vertical camera) to estimate the vegetation cover fraction (FCOVER)

CAN\_EYE has a set of specific features that improves its efficiency, accuracy, flexibility and traceability:

- Efficiency: a series of images is typically processed within 2 to 20 minutes, depending on the complexity of the images, the experience of the user and the performances of the used computer.
- Accuracy: the image conversion into a binarized image (green vegetation/other is performed through an interactive classification process. This provides more flexibility to separate green elements from the sky or the soil and allows acquiring images from both above or below the canopy.

- Flexibility: CAN\_EYE allows calibrating the imaging system, as well as defining the area of interest. Interactive masking tools allow the user to eliminate parts of the photos contaminated by undesirable objects. Interactive zooming tools are also included.
- Portability : CAN\_EYE is very easy to install
- Traceability: CAN\_EYE was developed such as all the processing steps are tracked. For these purposes, an html report and output documents (excel format) as well as intermediate results are automatically generated by the software.

## 1.2. CAN\_EYE basic principles

Figure 1 provides an overview of the different steps required to process series of images with CAN\_EYE:

### Set up of the processing

The user first selects the menu corresponding to the type of images he has acquired (DHP, DP57, DP0), if they are in RGB colors or binary and acquired upward (looking at the sky) or downward (looking at the soil).

After selecting the directory where the images to be processed are stored, the user has to define the characteristics of the processing. This setup configuration can be saved and used to process another series of photos (Figure 1, step 1). Note that for hemispherical images, a calibration method is proposed to characterize most of the fish-eye lens +camera systems.

When selecting binary images, the following steps are not required and CAN\_EYE can be used in batch processing (by selecting the directory that contains all the series of images to be processed) to automatically generate the outputs.

### Pre-processing the images (in case of RGB images)

The images are loaded and displayed in a window. It is possible to remove interactively the images that the user does not want to process. At this stage, it is possible to mask parts of the images that are undesirable. In addition, the 'gamma' factor can be changed to brighten or darken the images and provide a better visual discrimination between the vegetation elements and the background (Figure 1, step 2).

### Segmentation (in case of color images - RGB)

The segmentation is the most critical phase. It is an interactive step since the colors associated to each class depend on the illumination conditions and on the objects themselves. Further, some shaded (downward looking) or overexposed (upward looking) areas may make the distinction between green vegetation and background more difficult. The number of classes must be first set up (Figure 1, step 3). The user can choose between:

- 2 classes: green vegetation and background: the user must classify all the pixels of one of the two classes and all the unclassified pixels are considered to belong to the other class. It is generally more efficient and fast to choose the class that corresponds to the one represented with the less number of colors.



- 3 classes: green vegetation, background and mixed pixels: for both classes, the user selects the corresponding pixels. The remaining non classified pixels are considered as mixed and are processed as such during the binarisation process. This segmentation method is more tedious and time demanding than the one with 2 classes only.

Note that at the beginning of the classification process, the user can use indices to roughly classify the images by thresholding methods (Figure 1, step 4) and can then interactively refine his segmentation (Figure 1, step 5). Once the color allocation to classes is completed, the images are transformed into binary images.

### **Generation of the outputs**

This last step is automatic and does not require interaction with the user. The images are binarised using the segmentation results (Figure 1, step 6). A series of files are produced and stored in a directory located where the initial images were stored. The outputs (Figure 1, step 7) include a report file (html format) where all the elements of the processing are described. Additional text files (EXCEL or ASCII format) are also created where gap fractions, computed PAI, ALA, FAPAR and FCOVER are saved, depending on the type of processed images. Optionally, intermediate results can also be stored, allowing CAN\_EYE to run in batch mode to perform new or additional processing.

Note that there is also a possibility to summarize the results from series of CAN\_EYE results in an excel file.

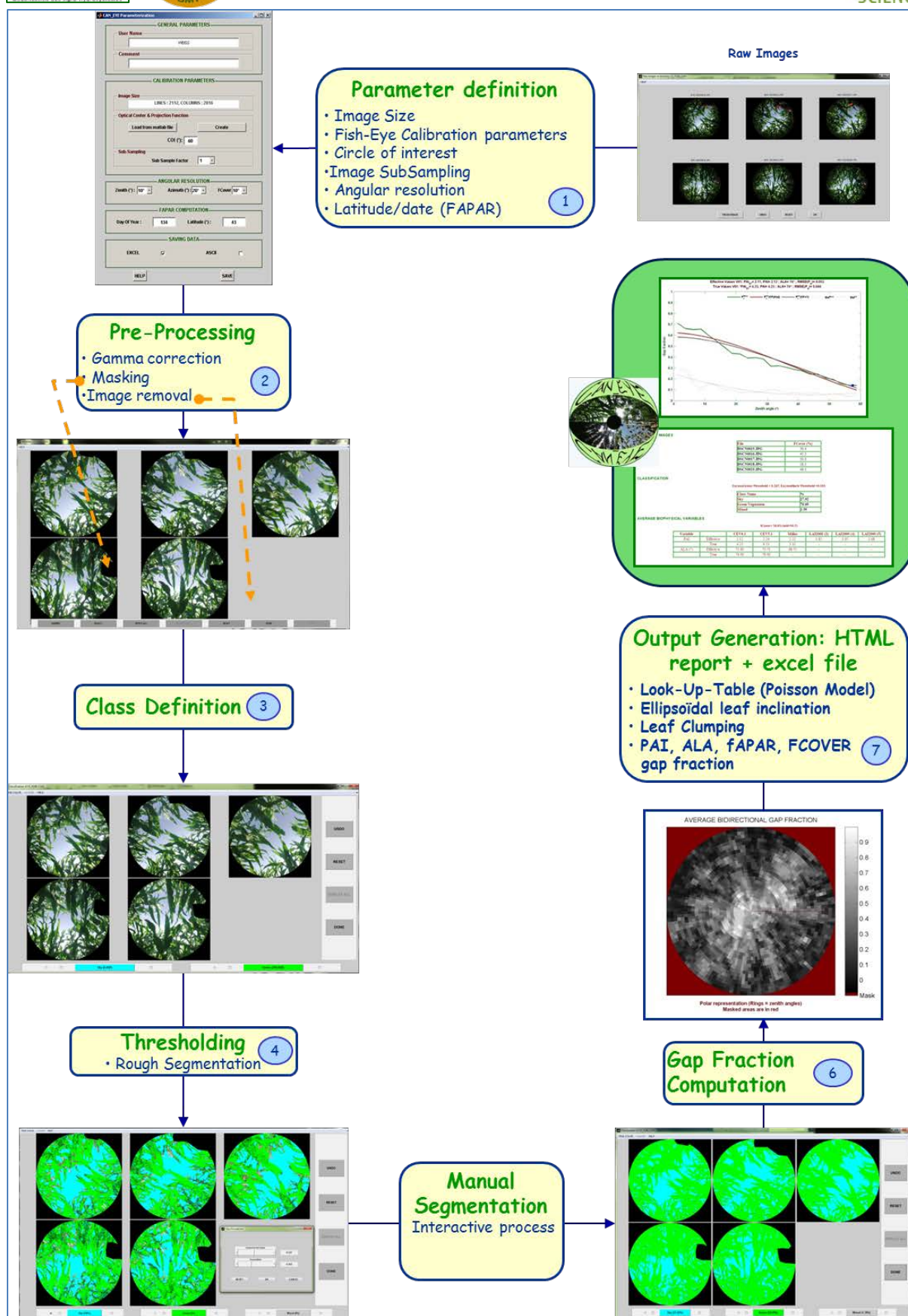


Figure 1. Overview of the CAN\_EYE processing of a series of images

### 1.3. Hardware and software requirements

CAN\_EYE is a software developed under MATLAB® and can be installed on any computer working under windows OS system. It is a compiled version that does not require to install MATLAB® on the computer. However, to use CAN\_EYE, the Matlab Component Runtime (MCR) is mandatory. The MCR is a collection of libraries required to run compiled MATLAB® codes and is downloaded for free. Note that each CAN\_EYE version is associated to a given MCR version indicated in the name of the CAN\_EYE installation file.

The required MCR version should be automatically installed when installing CAN\_EYE. However, sometimes, the automatic installation of the matlab component runtime can be very slow or does not work due to specific firewall or anti-virus settings. Therefore, the installation must be achieved manually after downloading (<http://fr.mathworks.com/products/compiler/mcr/>). Refer to section 2 or the CAN\_EYE website for more information ([http://www6.paca.inra.fr/can\\_eye](http://www6.paca.inra.fr/can_eye)).

Note that CAN\_EYE is no more supported on 32bits machine since version 6.43 (that can still be downloaded from the web site).

### 1.4. Copyright and limitations of responsibility

The CAN\_EYE software and the accompanying documentation described herein are provided as freeware. INRA reserves the right to make changes to this document and software at any time and without notice.

INRA shall not be liable for any direct, consequential, or other damages suffered by the use of the CAN\_EYE software package or its documentation.

### 1.5. Upgrading the version

A dedicated web site is available for the CAN\_EYE software at the address: [http://www6.paca.inra.fr/can\\_eye](http://www6.paca.inra.fr/can_eye). This site is regularly updated and new versions are available as well as a small FAQ. When downloading CAN\_EYE, you are asked to provide your e-mail. You will then be automatically informed of new releases. You can also subscribe to News on the web site.

### 1.6. Bug report

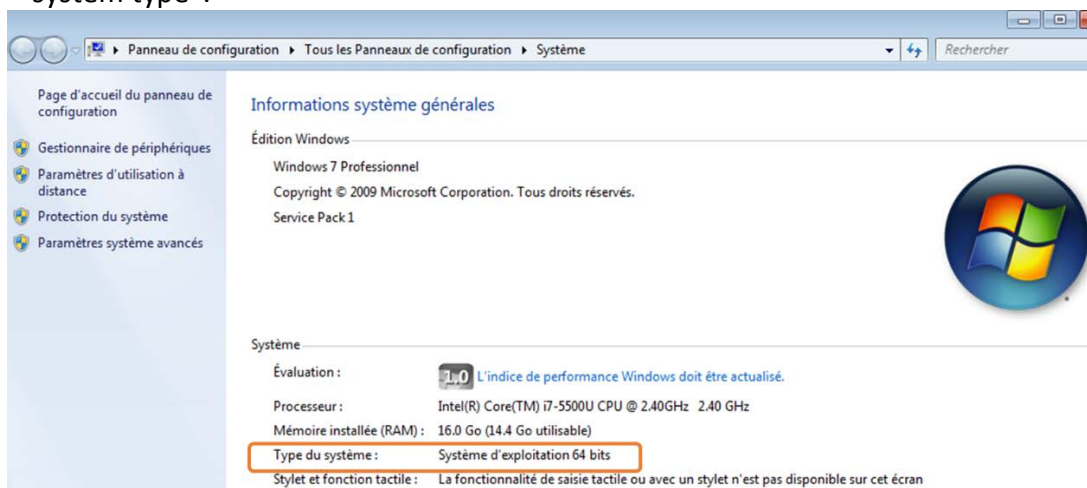
CAN\_EYE is aiming at being free from bugs. However not all the situations have been tested, and possible remaining bugs may exist. It is therefore recommended to use always the latest CAN\_EYE version available.

To report a bug, send a bug report to [can\\_eye-paca@inra.fr](mailto:can_eye-paca@inra.fr). The file "CAN\_EYE\_LogFile.txt" located in the directory C:\Users\username\AppData\Local\CAN\_EYE (where *username* corresponds to the login name of the windows session) must also be attached to your message. The user may also add an example of the processed images and the lens calibration parameters. Note that the time required to fix the bug will vary depending on its nature, and the workforce availability at INRA. We will try to do our best to answer rapidly.

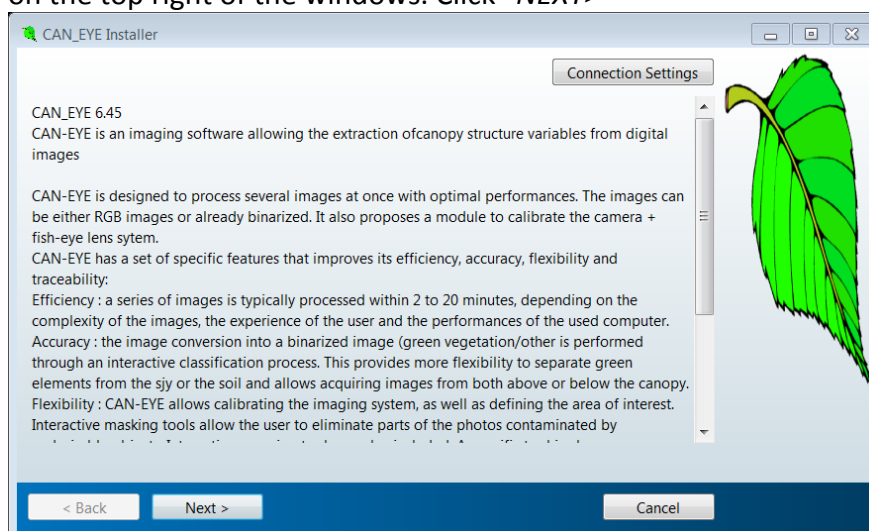
To facilitate the understanding of the bug report, the bug report may also contain additional information on the previous steps achieved before the bug happens: number of processed images, type, size, format, number of classes, etc... Send also eventually your parameter file (located in C:\Users\username\AppData\Local\CAN\_EYE\Param\_V6 where *username* corresponds to the login name of the windows session).

## 2. INSTALLATION

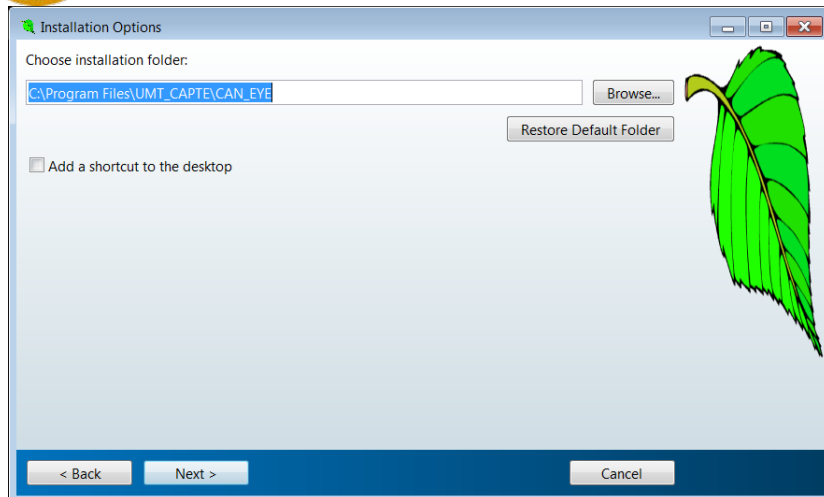
- Check the system type: 32 bits/64 bits: Select the “My Computer” icon on your desktop. Right-click the icon and select Properties from the menu. Check the item “system type”.



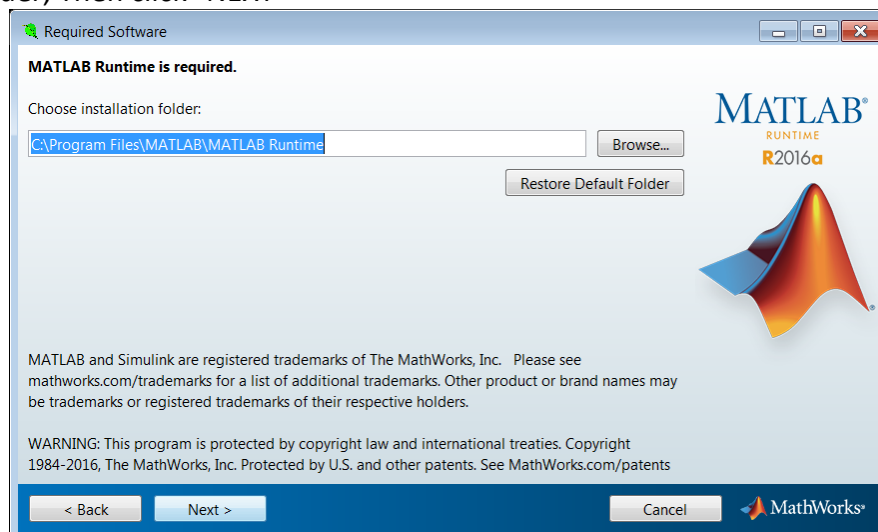
- Download the CAN\_EYE installation file compatible with your system. Click twice on the CAN\_EYE\_VX.xx\_Web\_RYYYYy\_ZZbits, (X.xx is the version number, YYYYYy is the year and version of the matlab component runtime, and ZZ is your system type).
- Once the installation is prepared, the CAN\_EYE installer automatically opens. Note that you need to be connected to internet, otherwise use the connection settings button on the top right of the windows. Click “NEXT>”



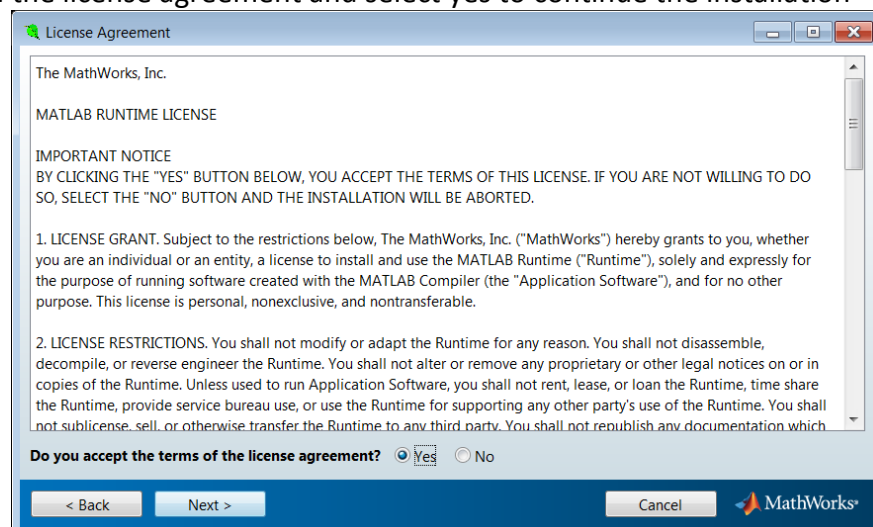
- Choose the installation directory. You can also choose to have a shortcut on your desktop (check the box). Then click “NEXT>”



- The system is checking if the compatible Matlab Component Runtime (MCR) is already installed on your computer.
  - If the MCR is not installed, the following window opens, select the installation folder, Then click “NEXT>”

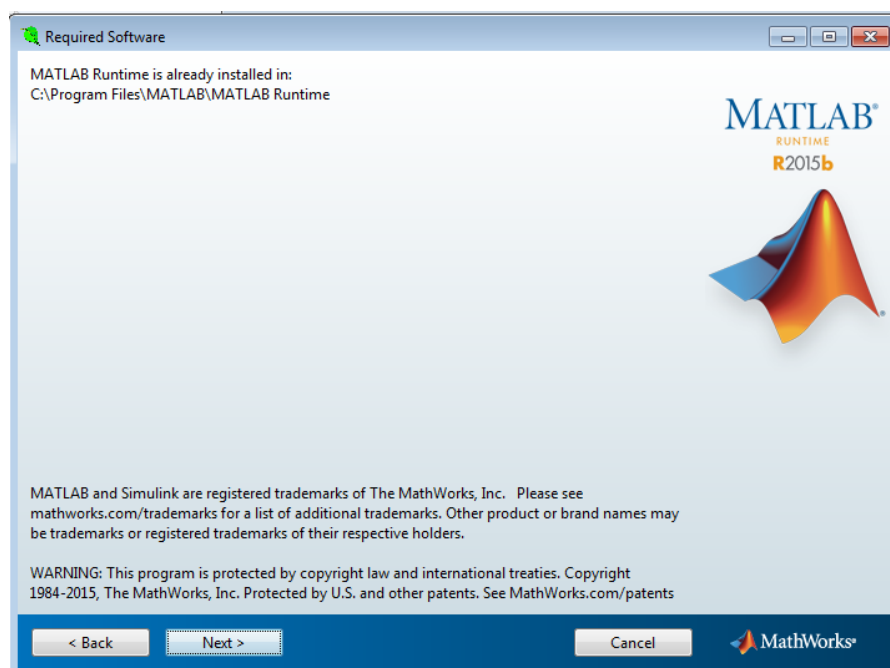


Read the license agreement and select yes to continue the installation

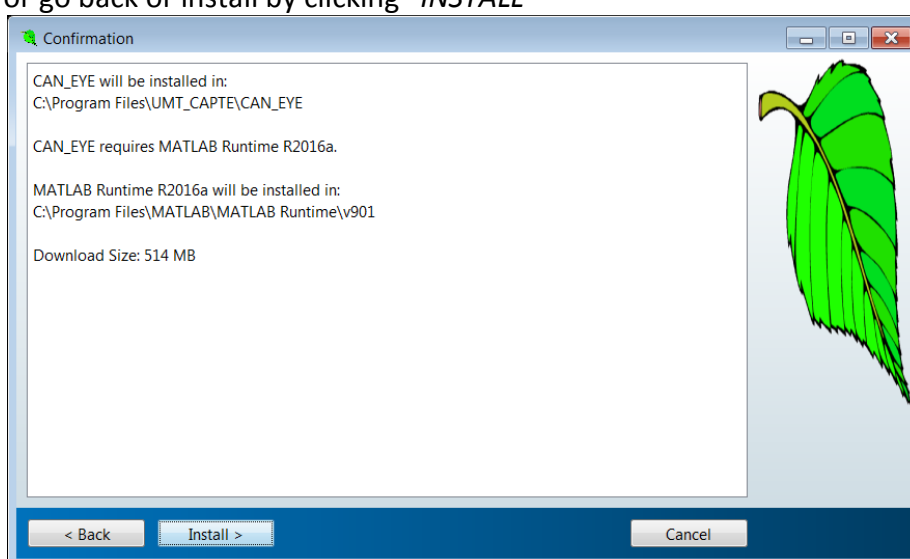




- If the MCR is already installed, the following window opens, then click “NEXT>”

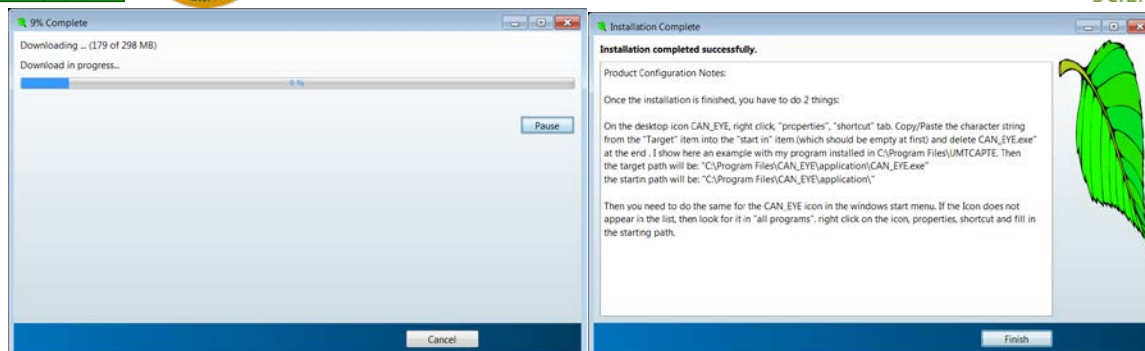


The system then confirms the installation (including the MCR installation if necessary). You can cancel or go back or install by clicking “INSTALL”



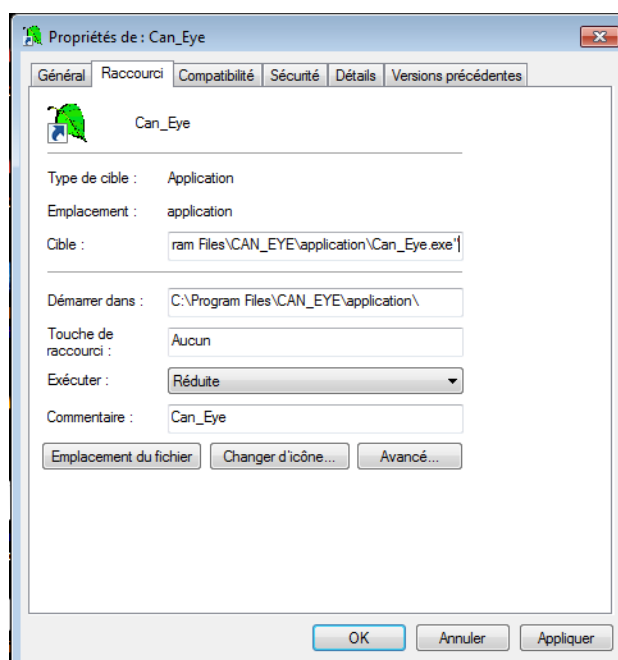
- A waitbar is then displayed, showing the progress of the installation (note that it can take several minutes, especially if the MCR is installed) and a confirmation window is displayed.





- Click on **"FINISH"**
- To be able to run CAN\_EYE from the windows startup menu or from your desktop icon, you need to perform one last manipulation. On the desktop icon CAN\_EYE, right click, "properties", "shortcut" tab. Copy/Paste the character string from the "Target" item into the "start in" item (which should be empty at first) and delete CAN\_EYE.exe" at the end. If the program is installed in C:\Program Files\, then
  - the target path will be: "C:\Program Files\CAN\_EYE\application\CAN\_EYE.exe"
  - the starting path will be: "C:\Program Files\CAN\_EYE\application\"

You need to do the same for the CAN\_EYE icon in the windows start menu. If the Icon does not appear in the list, then look for it in "all programs"



- Click twice on the CAN\_EYE icon to run the software. After a moment, the CAN-EYE main menu appears.

## 3. USING CAN\_EYE STEP BY STEP

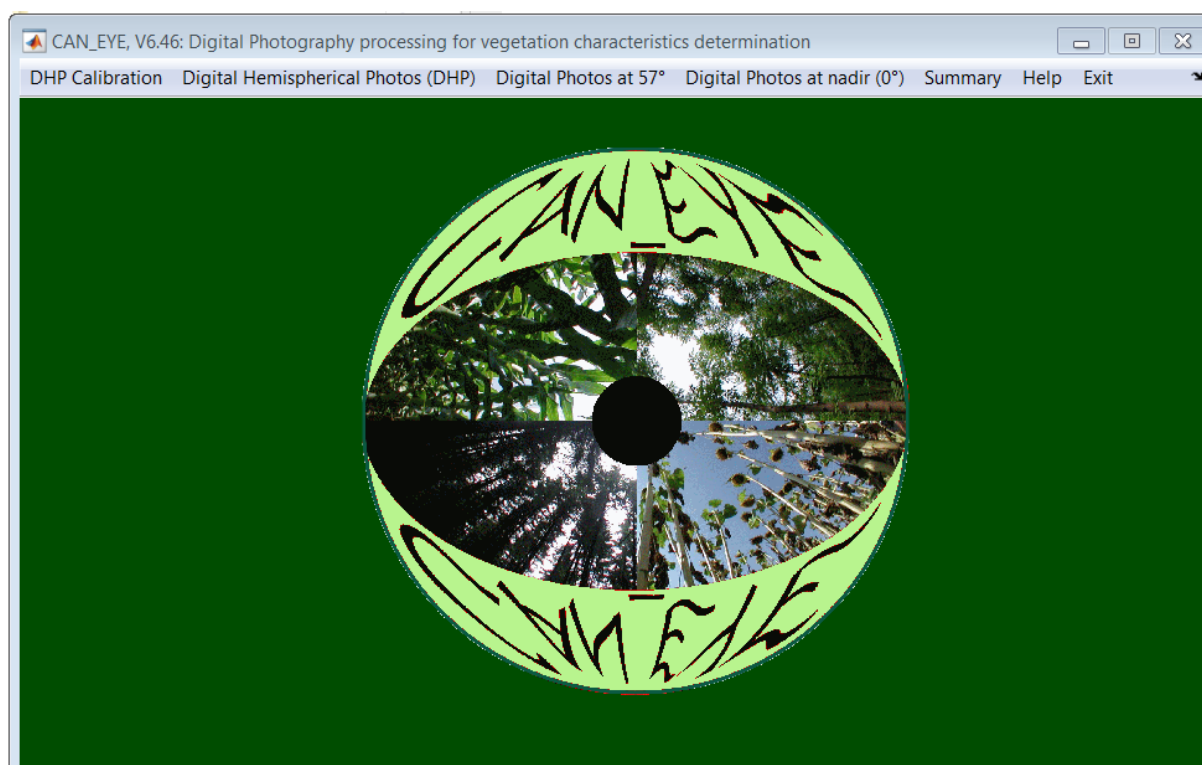


Figure 2. CAN\_EYE main menu

### 3.1. Choosing the image type to be processed

Series of images (either 'jpg' or 'TIF' format) to be processed at once must be stored in the same directory (Figure 2). These images are assumed to correspond to the same ESU (Elementary Sampling Unit) inside a field.

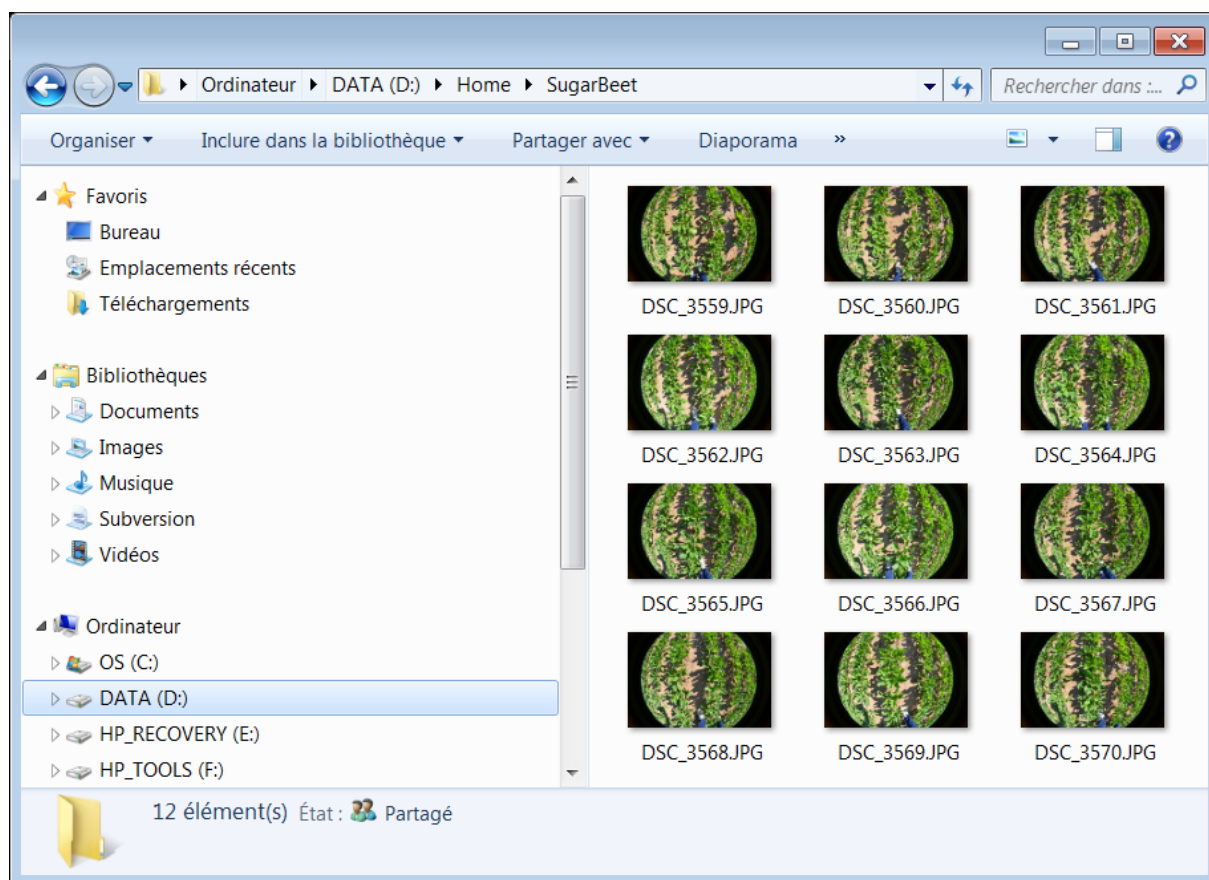
CAN\_EYE accepts only TIF and JPEG images. The images can be of any size (resolution). However, all the images to be processed concurrently and stored in a single directory should have the same format, size, camera setup (zoom, ...), as well as the same direction (upward or downward). If this is not the case, create as many directories as combinations of format, size (as well as camera set-up and direction).

The illumination conditions should be about the same within a series of images. If there are large differences in illumination conditions (such as strong direct light or strong diffuse conditions), it is recommended to split the series of images into homogeneous sub-series.

No more than 25 images can be processed by CAN\_EYE at once. Consequently, if more than 25 images are available for the same ESU, the user must split them into 2 or more directories.

Note that considering the assumptions made in CAN\_EYE (Poisson model), **it is not correct** to estimate the Plant Area index from the gap fraction using **a single image**. A minimum of 8 to 12 images is required to characterize an ESU (Elementary Sampling Unit) (Weiss et al., 2003). It is **not possible also to determine the PAI of a single tree or plant with CAN\_EYE**.

The image naming convention requires no special characters (such as ' ' '&',...). The postfix should be either .TIFF (TIF) or .JPG (JPEG). The image name is then tracked later through the processing and report files.



**Figure 3. Example of the « SugarBeet » directory containing a series of 12 images (.JPG) corresponding to one ESU and to be processed concurrently**

## *DHP: Digital Hemispherical Photographs*

Select 'Digital Hemispherical Photography (DHP)'. This Menu allows the processing of DHP images acquired with a camera + fish-eye lens. This allows the derivation of PAI, ALA, FAPAR and FCOVER. It is possible to directly process:

- **Color DHP (RGB)** (jpeg or TIF) acquired with the system:
  - **upward** : camera on the ground looking towards the sky
  - **downward** : camera above the canopy looking towards the soil

Select the images to be processed. Results will be stored in a sub-directory created in the image directory and called CE\_P180\_imagedirectoryname

- **Binary DHP** *i.e.*, images that have already been classified by the user with another tool than CAN-EYE. These images must be a gray level image (Red Channel=Green Channel=Blue Channel) in TIF format, unsigned integer 8 bit (uint8). Only 3 values are possible: 0 indicates that the pixel is vegetation, 100 indicates the pixel is a gap, 255 indicates a masked pixel. All the images corresponding to the same ESU (Elementary

Sampling Unit) must be stored in a zip file. Select the directory where the zip files are stored. All the zip files will be processed, and results will be stored in subdirectories called CE\_P180\_zipfilenames.

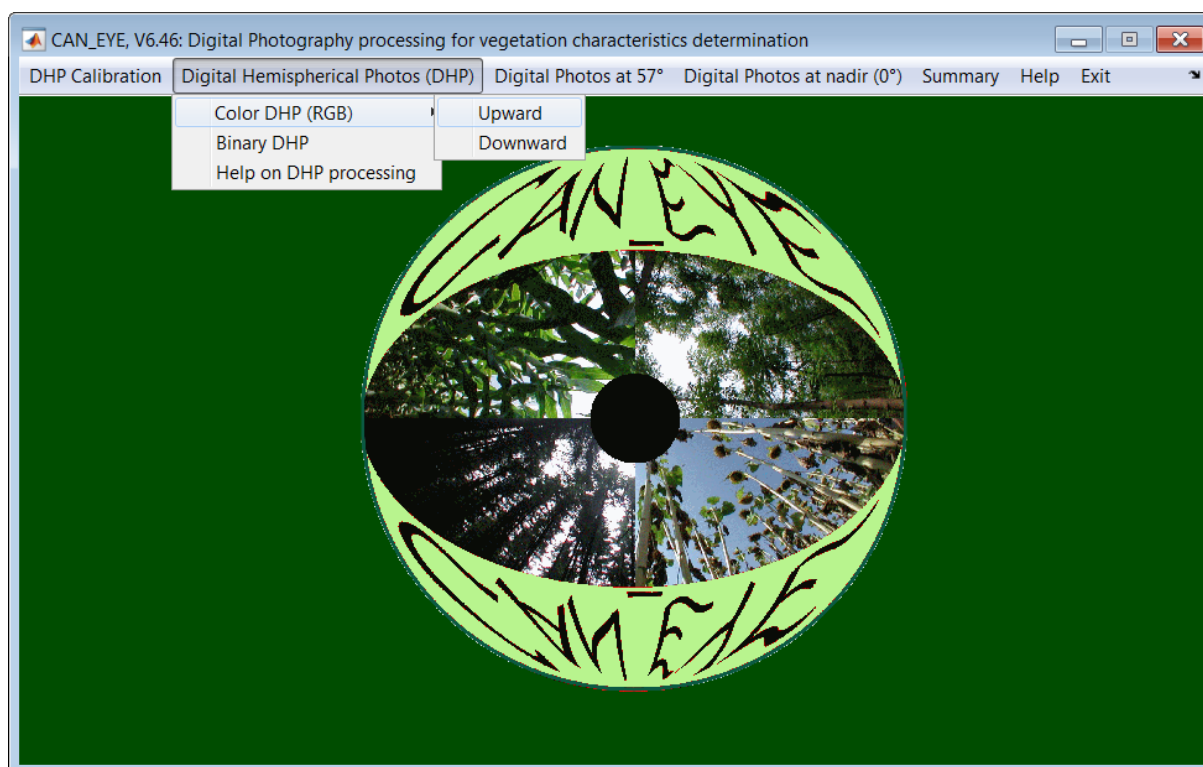


Figure 4. CAN\_EYE Menu for digital hemispherical photographs (fish-eye lens)

## *DP57: Digital Photographs acquired with a camera inclined at 57.5°*

Select 'Digital Photos at 57°'. This Menu allows the processing of images acquired with a camera inclined at 57.5° from the vertical. For this particular direction the gap fraction is independent on leaf inclination angle (Weiss et al., 2003). This allows the derivation of PAI only.

It is possible to directly process:

- **Color DP57 (RGB)** (jpeg or TIF) acquired with the system:
  - **upward** : camera on the ground looking towards the sky
  - **downward** : camera above the canopy looking towards the soil

Select the images to be processed are stored. Results will be stored in a sub-directory created in the image directory and called CE\_P157\_imagedirectoryname
- **Binary DP57** *i.e.*, images that have already been classified by the user with another tool than CAN-EYE. These images must be a gray level image (Red Channel=Green Channel=Blue Channel) in TIF format, unsigned integer 8 bit (uint8). Only 3 values are possible: 0 indicates that the pixel is vegetation, 100 indicates the pixel is a gap, 255 indicates the pixel is masked. All the images corresponding to an ESU (Elementary Sampling Unit) must be stored in a zip file. Select the directory where the zip files are

stored. All the zip files will be processed, and results will be stored in subdirectories called CE\_P57\_zipfilenames.

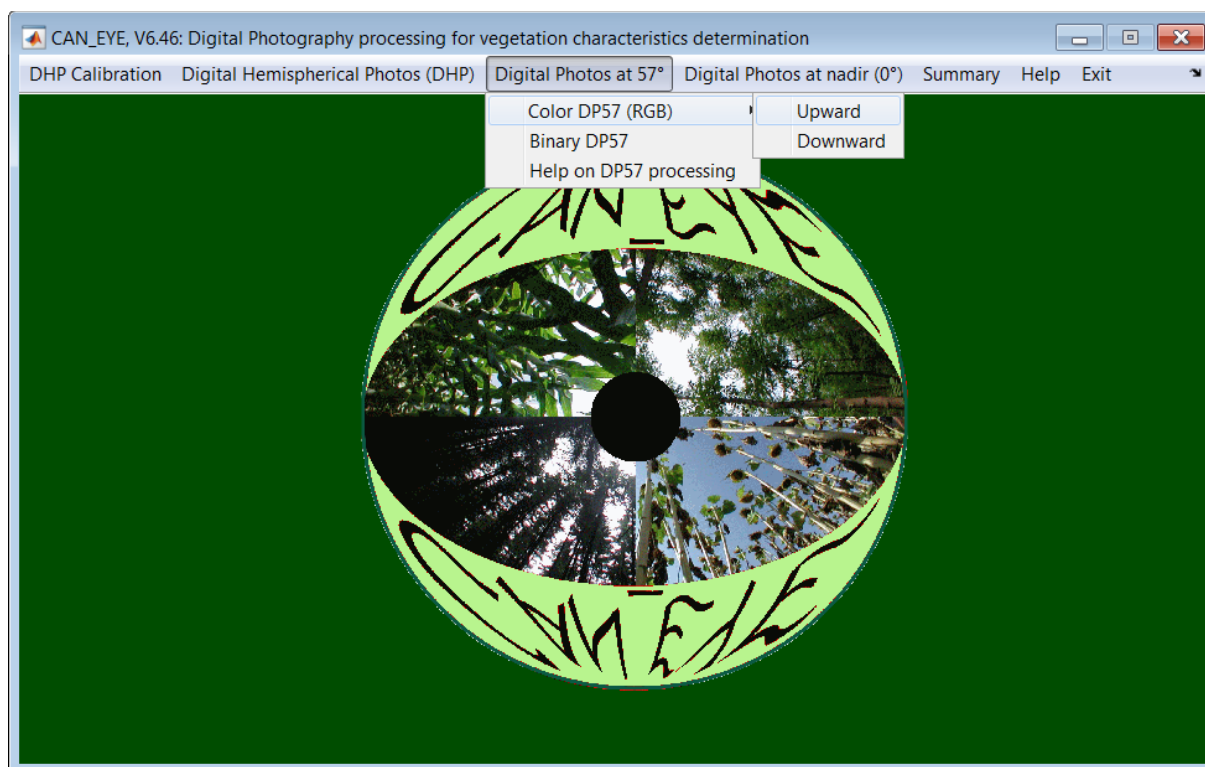


Figure 5. CAN\_EYE Menu for digital photographs acquired with a camera inclined at 57.5

## *DPO: Digital Photographs acquired with a camera at the vertical*

Select 'Digital Photos at nadir (0°)'. This Menu allows the processing of images acquired with a camera looking vertically towards the ground or the sky (no fish-eye lens). This allows the derivation of the vegetation cover fraction (FCOVER) only. It is possible to directly process:

- **Color DP0 (RGB)** (jpeg or TIF) acquired with the system:
  - **upward** : camera on the ground looking at the sky
  - **downward** : camera above the canopy looking at the soil

Once the user has chosen one of these two options, he is asked to choose a series of images (either 'jpg' or 'TIF' format) that he wants to process at the same time. Then the user is asked to provide the CAN\_EYE processing parameters. Results are stored in a subdirectory called CE\_NADIR\_Directoryname.



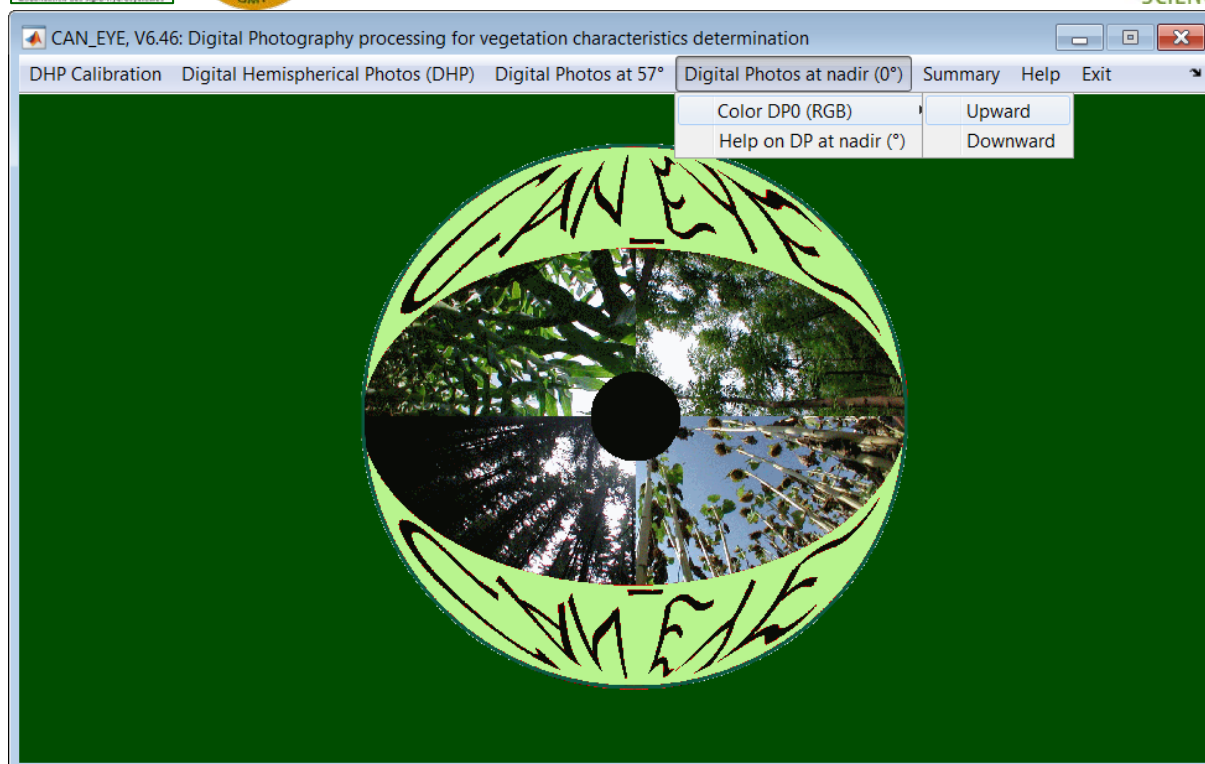


Figure 6. CAN\_EYE Menu for digital photographs acquired with the camera at the vertical (nadir viewing)

## 3.2. Defining the processing parameters

Once the directory containing the images to be processed is selected, the user must provide the processing parameters (including optics characterization and CAN\_EYE specific parameters) that differ depending on the camera/lens type and settings. Information must be manually entered in a specific window (described hereafter).

For hemispherical images, some pre-computation are performed and saved in order to save time when processing new directories with images acquired with the same system and settings. The processing parameters are then stored in the Param\_V6 subdirectory (C:\Users\username\AppData\Local\CAN\_EYE, where username corresponds to the login name of the windows session), with a default proposed name.

### *Processing parameters for DHP*

This window allows defining all the camera+fish-eye lens characteristics (optical center, projection function), as well as the characteristics required for the processing (angular resolution, FAPAR computation). This allows making some pre-computation that are saved to speed-up the processing when using the system for another set of images acquired in the same conditions.

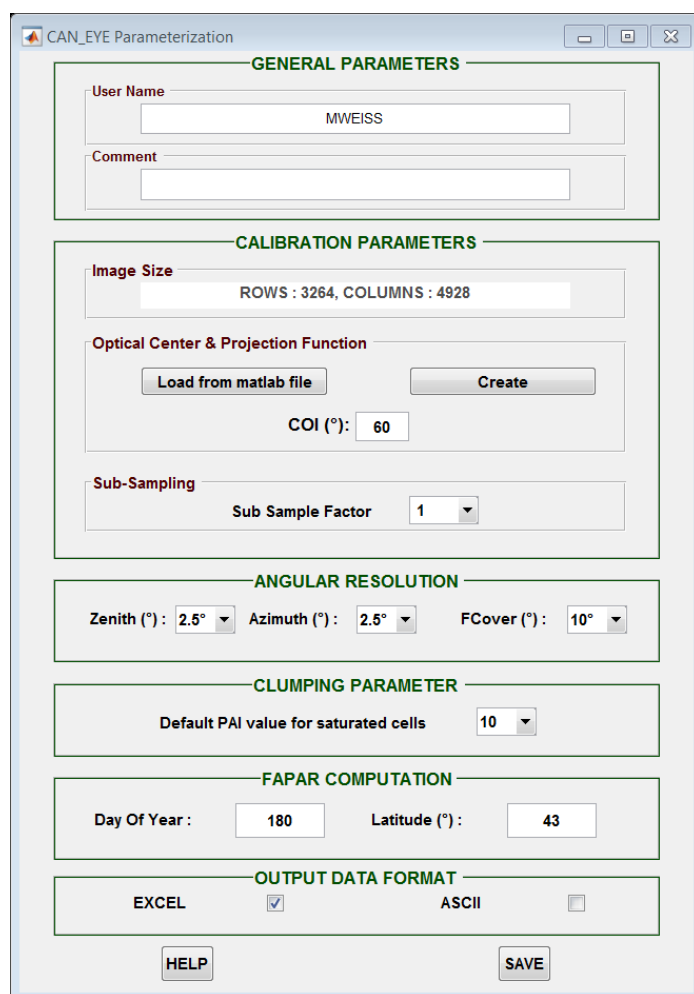
Note that CAN\_EYE supports only polar projection function or equidistant projection function that is the most common for fish-eye lenses: the angular distances (in degrees) in the object region are proportional to radial distances in pixels on the image plane (Hughes et al., 2010). To allow derivation from this polar projection, it is possible to use a polynomial of order 2 or 3. CAN\_EYE proposes a method to determine the characteristics of a given system. For more information, see section 5.



## Description of the processing parameter window for DHP

- **User Name:** by default will display the “user name” environment system variable (login name of the windows session) of the computer
- **Comment:** add any useful comment (will be written in the processing report & output files)
- **Image size:** automatically filled in by CAN\_EYE
- **Optical center & Projection function:** these characteristics can either be loaded from a matlab file generated with the CAN\_EYE calibration menu or directly provided by the user (see hereafter)
- **COI** is the limit of the image in degrees used during the processing. By default, set to 0°-60° (zenith angles higher >60° are not taken into account due to large occurrence of mixed pixels in these areas).
- **Sub Sample Factor:** If there are too many images or they are too big, the computer may not have enough memory. A possibility is to perform the processing using only one pixel over 2 (Sub Sample Factor=2) or one pixel over 3 (Sub Sample Factor=3).
- **Angular Resolution** in the zenith ( $\theta$ ) and azimuth ( $\varphi$ ) directions (°) determines the angles for which the Gap Fraction will be computed. Low values induce higher computation time, but more accuracy. By default, it is set to the lowest value (highest resolution) for both angles.
- **FCover** (in degrees) defines the size of the solid angle used to compute the cover fraction (*i.e* the gap fraction in the nadir direction). By default it is set to 10 degrees.
- **Clumping Parameter:** when computing the clumping parameter, each zenithal ring is divided into cells (see section 4.1, p39). No gap in a cell (saturated cell) theoretically corresponds to an infinite PAI. The infinite PAI is then replaced by the PAISat value. Note that the number of saturated cells found during the computation is provided with the results.
- **FAPAR** is computed as the integral of (1-gap fraction) during the sun course. The latter is determined for a given day provided in the day number of the year (default is the acquisition date of the image) at a given latitude in degrees (default is 43° but you need to provide the latitude of your experiment site)
- **SAVING DATA:** results can be written in an excel file or ascii file or both
- **SAVE:** click on this button once you have finished filling the parameters. The parameter file will be saved in the directory C:\Users\mweiss\AppData\Local\CAN\_EYE\Param\_V6. This parameter file can be used later on to process images issued from the same system (camera + fish-eye lens). Note that this allows performing some pre-computations and speed up future processing.
- The default parameter file name for the specific window shown here as an example is:  
P180\_2112\_2816\_Cent\_1063\_1390\_ProjFuncDeg1\_COI60\_Sub1\_Teta5\_Phi20\_FCOV10.mat  
Where:  
P180 : indicates that the parameter file is for a camera+fish-eye system  
2112\_2816 : is the resolution of the image (lines – rows)  
Cent\_1063\_1390 : are the coordinates of the optical center

ProjFuncDeg1 : 1 is the polynomial degree of the projection function  
 COI60 : the limit of the circle of interest is 60°  
 Sub1 : subsample factor is 1 (image not degraded)  
 Teta5 : The zenith angular resolution for gap fraction is 5°  
 Phi20 : The azimuth angular resolution for gap fraction is 20°  
 Fcov10 : The vegetation cover fraction is estimated using the ring from  
 zenith angle =0° to zenith angle =10°



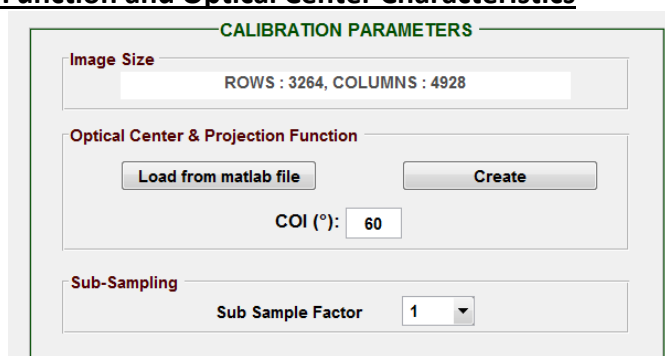
The image shows the 'CAN\_EYE Parameterization' window. It is divided into several sections:

- GENERAL PARAMETERS:** Includes 'User Name' (MWEISS) and 'Comment'.
- CALIBRATION PARAMETERS:** Includes 'Image Size' (ROWS : 3264, COLUMNS : 4928), 'Optical Center & Projection Function' (with 'Load from matlab file' and 'Create' buttons, and 'COI (°): 60'), and 'Sub-Sampling' (Sub Sample Factor: 1).
- ANGULAR RESOLUTION:** Includes 'Zenith (°): 2.5°', 'Azimuth (°): 2.5°', and 'FCover (°): 10°'.
- CLUMPING PARAMETER:** Includes 'Default PAI value for saturated cells' (10).
- FAPAR COMPUTATION:** Includes 'Day Of Year' (180) and 'Latitude (°)' (43).
- OUTPUT DATA FORMAT:** Includes 'EXCEL' (checked) and 'ASCII' (unchecked).

At the bottom, there are 'HELP' and 'SAVE' buttons.

Figure 7. Processing parameter window for hemispherical images

## Creating Projection Function and Optical Center Characteristics



This image shows a zoomed-in view of the 'CALIBRATION PARAMETERS' section of the window. It includes:

- Image Size:** ROWS : 3264, COLUMNS : 4928
- Optical Center & Projection Function:** 'Load from matlab file' and 'Create' buttons, and 'COI (°): 60'.
- Sub-Sampling:** Sub Sample Factor: 1.

Figure 8. Defining calibration parameters in the Processing parameter window

When pressing the **create button** in the CALIBRATION PARAMETERS subsection, the user must enter the characteristics of the lens used to acquire the images:

- Optical Center: location of the optical center along the lines (Y) and rows (X) knowing that the upper left corner of the image has coordinate (1,1) and the lower right corner is located at the (NbLines, NbRows)
- Projection function: the radius of the image (in degree) is considered as a polynomial function (maximum order = 3) of the distance between a pixel of the image to the optical center (if the degree of the polynomial is 1, then the projection is assumed to be polar or equidistant). Be very careful when entering the coefficient to sort them in the right order (descending power). Note that after entering the polynomial coefficient, the polynomial function is plotted so that the user is able to check if it is correct.

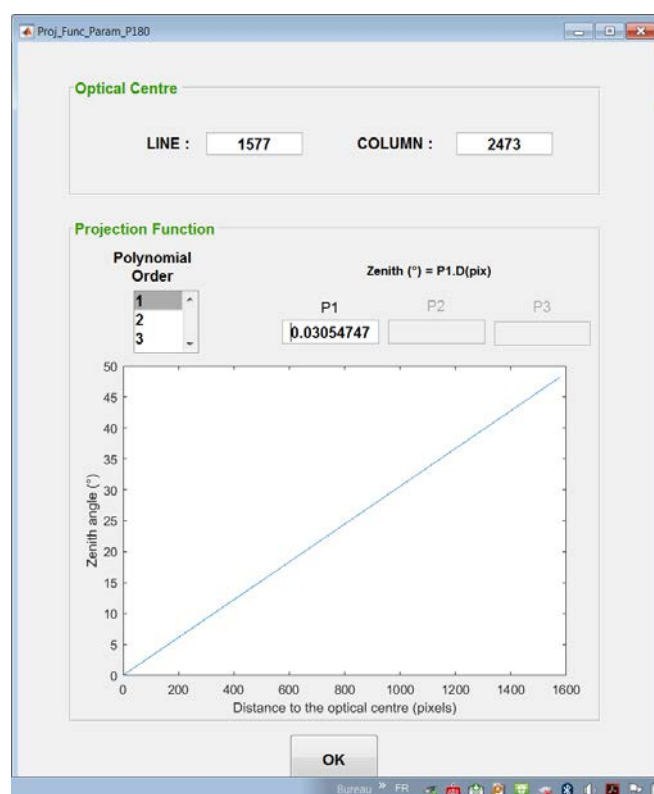


Figure 9. Creating projection function and optical center characteristics

If the camera can not calibrated, it can be assumed as perfect. In this case, the lens characteristics can be computed as the following:

- the optical sensor is at the center of the image: its coordinates are therefore half the number of lines and the number of rows
- the projection function is equidistant (polynomial order 1). The P1 coefficient is then computed as the ratio of the maximum field of view of the fish-eye lens ( $FOV_{max}$ ) divided by the length of the portion of the diagonal ( $L_D$ ) of the image that does not include background fill color. The maximum field of view of the fish-eye lens is provided in the fish-eye technical specifications and corresponds to the field of view

in the diagonal of the image.  $P_1 = \frac{FOV_{max}}{L_D}$ . When the image is not circular (Figure 10, right part),  $L_D = \sqrt{Length^2 + Width^2}$

We recommend not to use these default parameters and to calibrate the system in order to get the most accurate results. A calibration method is proposed in section 6 for equidistant projection. Note that implementing the calibration experiment should not take more than half a day.

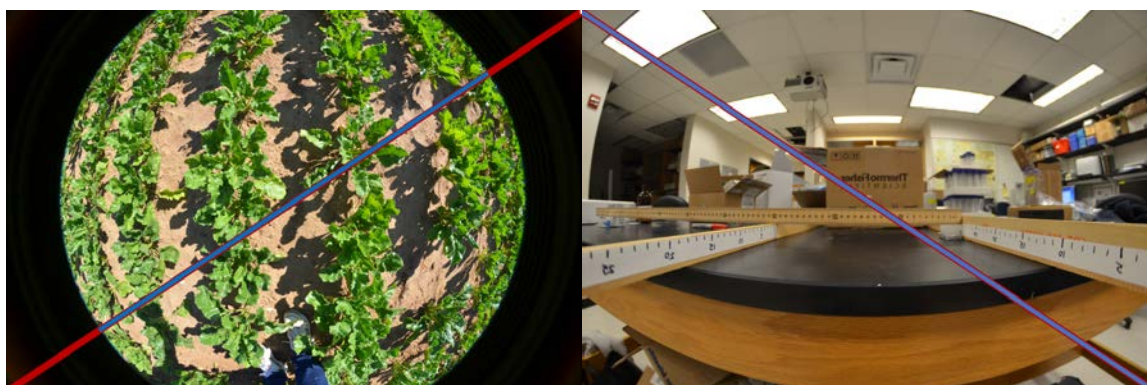


Figure 10. Determining the default projection function parameter of two different systems. On the left: fish-eye length with a large field of view (around 90°). On the right fish-eye length with a narrower field of view (around 60°). The diagonal is shown in red,  $L_D$  is the length of the portion shown in blue.

When pressing the '**Load from matlab file**' button in the CALIBRATION PARAMETERS subsection, the user must provide the file (extension '.mat') created with the CAN\_EYE calibration menu (see section 6).

## *Processing parameters for digital images acquired at 57° (DP57)*

This window allows defining all the camera characteristics (field of view), as well as the characteristics required for the processing of images acquired with the camera inclined at 57.5° (no fish-eye lens):

- **User Name:** by default will display the "user name" environment system variable of the computer
- **Comment:** add any useful comment (will be written in the processing report & output files)
- **Image size:** automatically filled by CAN\_EYE from the properties of the images
- **Sub Sample Factor:** in case you are processing very high resolution images, and/or a low field of view camera, and/or a high number of images, your computer may be "out of memory". Use the subsample factor (SSF) to process only one pixel over SSF of the images.
- **Focal Length:** Automatically filled by CAN\_EYE from the properties of the images
- **CCD Type:** Provide the camera CCD type. This is indicated in the camera technical specifications. It is expressed as a fraction of an inch, with a one in the numerator, and a decimal number in the denominator. This "inch" system brings a result approximately 1.5 times the length of the diagonal of the sensor. Due to inch-based sensor formats being not standardized, their exact dimensions may vary, this is the

reason why the camera CCD width and height must also be indicated. This allows computing the actual sensor field of view. For a given CCD Type, CAN\_EYE provides typical CCD width and height. However, the actual values must be checked in the camera technical specifications.

- **CCD Width (mm):** Provide the camera CCD width. For a given CCD Type, CAN\_EYE provides typical CCD width. However, the actual values must be checked in the camera technical specifications.
- **CCD Height (mm):** Provide the camera CCD height. For a given CCD Type, CAN\_EYE provides typical CCD width. However, the actual values must be checked in the camera technical specifications.
- **Field of View (FOV °):** this allows extracting the useful part of the images, *i.e.*,  $0^\circ \pm \text{FOV}^\circ$  to compute the cover fraction at  $57^\circ$ .
- **Cell Size:** the true leaf area index is computed using the Lang & Xiang average logarithm method. The Cell Size in pixels corresponds to the size of the cell where local PAI is computed and the medium is assumed turbid. Default Value is 30 pixels.
- **Output Format:** results can be written either in an excel file or ascii file or both

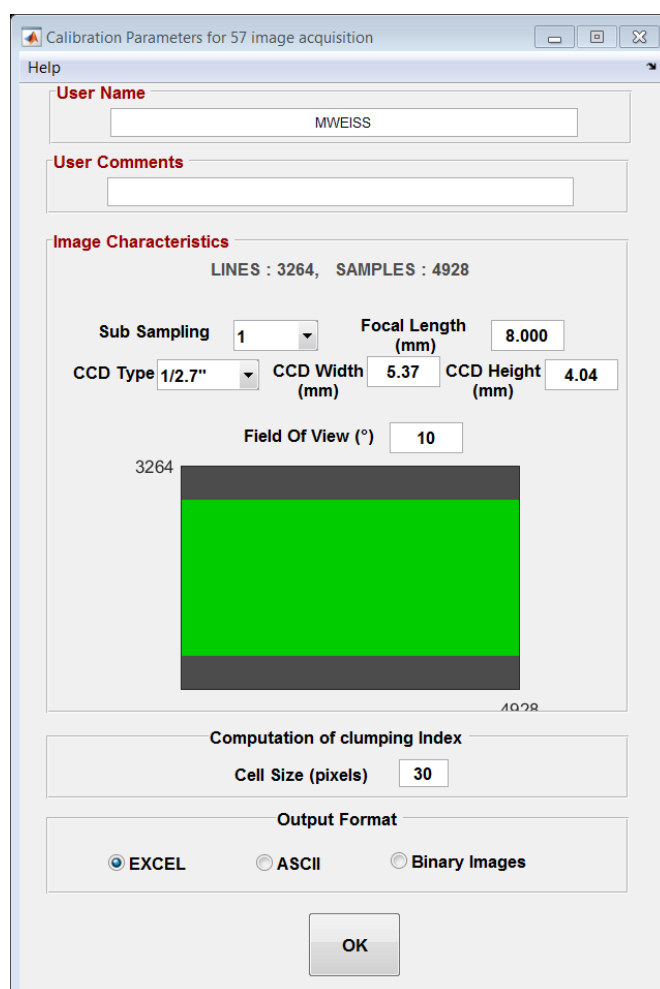


Figure 11. Processing parameter window for images acquired at  $57.5^\circ$

*Processing parameters for images acquired at nadir*

- **User Name:** by default will display the “user name” environment system variable of the computer
- **Comment:** add any useful comment (will be written in the processing report & output files)
- **Image size:** automatically filled by CAN\_EYE from the properties of the images
- **Sub Sample Factor:** in case you are processing very high resolution images, and/or a low field of view camera, and/or a high number of images, your computer may be “out of memory”. Use the subsample factor (SSF) to process only one pixel over SSF of the images.
- **Focal Length:** Automatically filled by CAN\_EYE from the properties of the images
- **CCD Type:** Provide the camera CCD type. This is indicated in the camera technical specifications. It is expressed as a fraction of an inch, with a one in the numerator, and a decimal number in the denominator. This "inch" system brings a result approximately 1.5 times the length of the diagonal of the sensor. Due to inch-based sensor formats being not standardized, their exact dimensions may vary, this is the reason why the camera CCD width and height must also be indicated. This allows computing the actual sensor field of view. For a given CCD Type, CAN\_EYE provides typical CCD width and height. However, the actual values must be checked in the camera technical specifications.
- **CCD Width (mm):** Provide the camera CCD width. For a given CCD Type, CAN\_EYE provides typical CCD width. However, the actual values must be checked in the camera technical specifications.
- **CCD Height (mm):** Provide the camera CCD height. For a given CCD Type, CAN\_EYE provides typical CCD width. However, the actual values must be checked in the camera technical specifications.
- **Field of View (FOV °):** this allows extracting the useful part of the images, *i.e.*,  $0^\circ \pm \text{FOV}^\circ$  to compute the cover fraction at  $0^\circ$  (nadir).
- **Output Format:** results can be written either in an excel file or ascii file or both



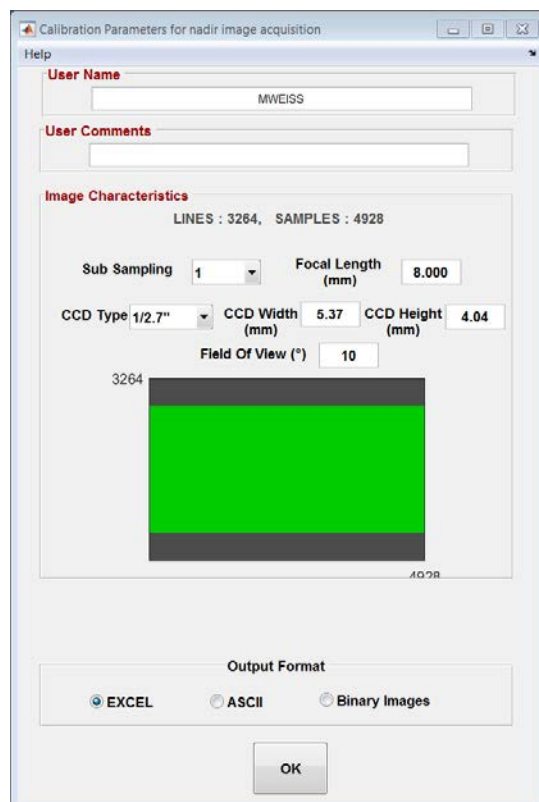
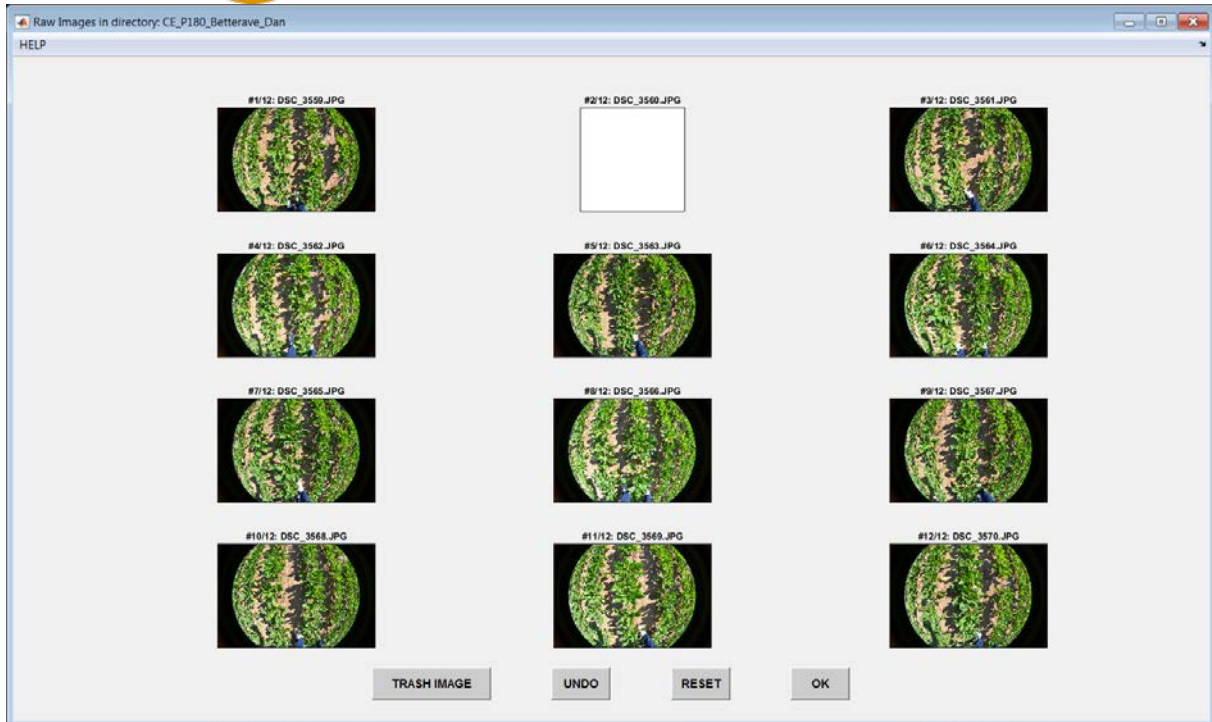


Figure 12. Processing parameter window for images acquired at nadir (0°)

### 3.3. SELECTING PERTINENT IMAGES

Once the processing parameters are defined, a window is opened, showing all the images contained in the processing directory. Some of the images may not be pertinent for the processing (fuzzy or overexposed images for example): they must be eliminated.



**Figure 13. Selecting pertinent images**

**TRASH IMAGE:** click on this button and select the image(s) that will not be used in the processing with the left mouse button. End your selection by clicking the right mouse button. The selected images will become white, indicating that they won't be used for the processing.

**UNDO:** click on this button to cancel your last selection.

**RESET:** click on this button to reset the selection (in that case all the images in the directory are selected).

**DONE:** click on this button when you have finished your selection.

## 3.4. MASKING IMAGES

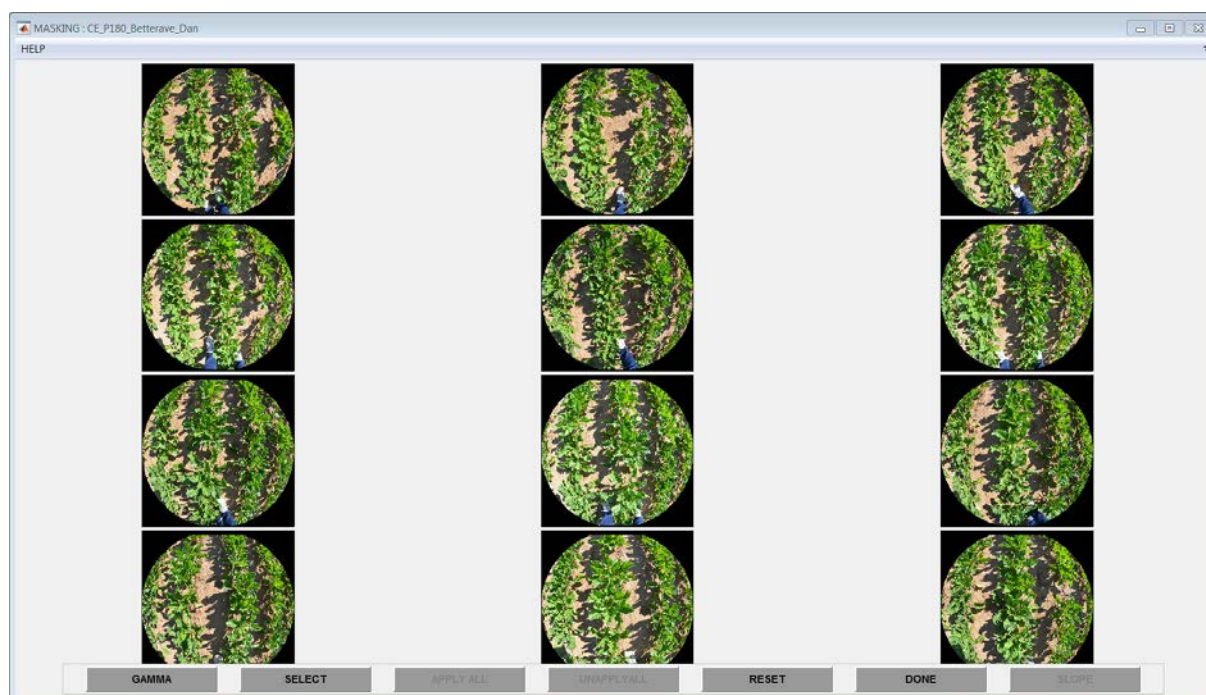
Some parts of the images should not be processed by the user because:

- They do not correspond to vegetation or background (Operator or undesired object, are present on the images)
- They will be hardly classified (Presence of some over-exposed areas in the images : both leaves or soil or sky appear completely white)

This part of the processing allows masking these areas so that they will not be taken into account in the image processing.

There is also the possibility to apply a gamma correction factor to the image. This correction is just a visual correction to help the user to better discriminate between vegetation and background. It will not impact the model inversion results.

Once the pertinent images are selected, the user gets access to the following windows. Note that only the useful part of the images that were not trashed are displayed. The useful part is determined thanks to the CAN\_EYE processing parameters (for DHP: COI, optical center and projection function, for standard images: CCD sensor type, width, and length, as well as FOV).



**Figure 14. Main masking window: the button name appearing in light gray indicate that the functionalities are not available at this step of the processing.**

## **GAMMA:** *Gamma Correction*

Select this button to get access either to a frame where you can put your gamma value (left side of the window), either to the slider changing the gamma value. The gamma can be reset to its initial value (reset button). Click OK when the image appears easily classifiable for you.

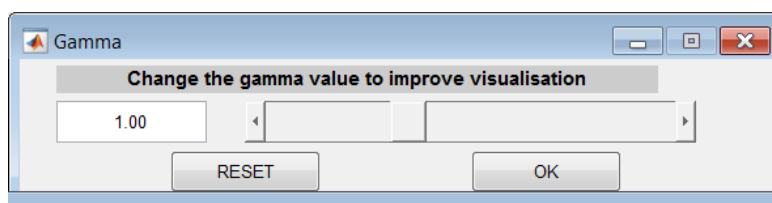


Figure 15. Gamma correction value (masking step)

## **SELECT:** *Selection of one image to define a mask*

After clicking on this button, the user has to select the image to mask (mouse left click on the desired image). The image then appears solely on the window, with 3 buttons available

### **MASK:**

After clicking on this button, the user has to define a polygon (corresponding to the masked area on your image) by clicking the left mouse button to indicate the polygon vertices. The user can then visualize the drawn polygon (see figure below). To end the masking, the user just clicks on the right mouse button. The user can add as many masks as he wants (by selecting again the mask button). The mask appears in black, as the background of the image. Tip: when defining the polygon on the circular border, extend the polygon to the black background to make sure that all the border is masked.

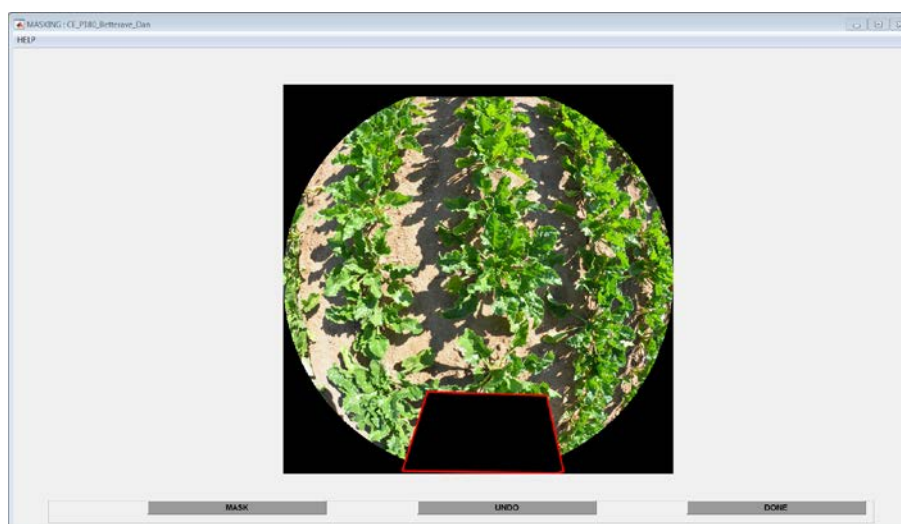


Figure 16. Secondary masking window (after having selected the image to be masked). The mask appears in black bordered by red color.

### **UNDO:**

Clicking on his button cancels the last mask that was designed.

## DONE:

Click on this button when you have finished the masking process on the image

## APPLY ALL: *Apply the last mask to all images*

When clicking on this button, the last mask that has been drawn is applied to all the images (see figure below). This can be very useful if the object to be masked is always at the same place in the images (such as the operator for example).



Figure 17. Applying a mask to all images. The same masked areas appear in black on all images.

## UNAPPLY ALL: *Delete the last mask applied to all images*

Clicking on this button deletes the last mask applied to all the images.

## UNDO: *Undo last mask*

Clicking on this button deletes the last mask that was defined. It is possible to click x times on this button to delete the last x masks that were defined.

## RESET: *Reset all the masks*

Clicking on this button resets all the masks for all the images.

## DONE: *End the masking*

Clicking on this button ends the masking process and leads to the next step, e.g. image classification



## 3.5. CHOICE OF THE INTERACTIVE CLASSIFICATION METHOD

Once the masking step is achieved, CAN\_EYE runs the image indexation to speed the processing time. The CLASS DEFINITION window allows indicating the way you intend to achieve the interactive classification with CAN\_EYE. Two choices are possible depending if you consider mixed pixels or not.

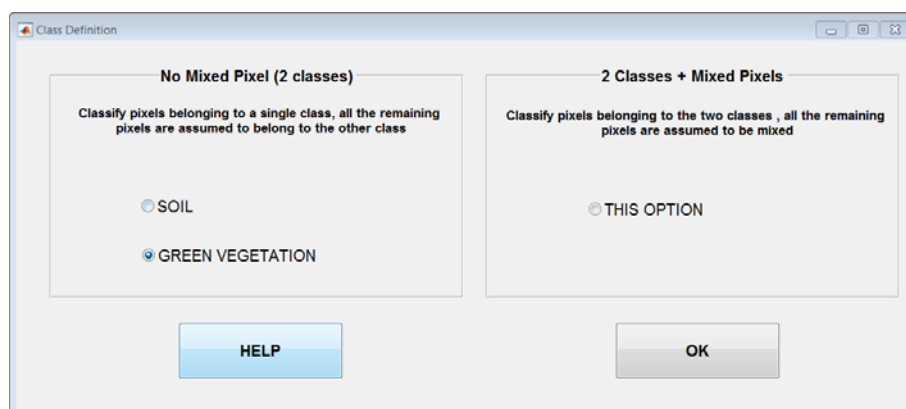


Figure 18. Choice of the interactive classification method

### No mixed pixels (2 classes)

This option allows you to classify pixels belonging to one class **only** (background or vegetation) and all the pixels that remain unclassified are assumed to belong to the other class (vegetation or background). In this example, the user chose to select only pixels corresponding to the vegetation.

Note that if the user is processing images acquired downward, the background is named 'soil', while if he is processing upward images, it is named 'sky'.

### 2 Classes + Mixed Pixels

In this case, the unclassified pixels will be considered as mixed. The user will have to classify

- all the pixels that, for sure, belong to the background
  - all the pixels that, for sure, belong to the green vegetation
  - all the remaining pixels are considered as mixed and will be processed later as such
- The gap fraction will be computed as the weighted average between background (0) and green vegetation (1). Weights are computed as the distance in terms of color between the pixels and the two classes.



## 3.6. Classifying the images

Once the classes are defined, the classification window appears on the screen. Note that if no radio button is selected in the bottom menu, the user can zoom on a particular image simply by clicking on it. Then, he can zoom on this particular image by clicking the left mouse button, or by dragging a rectangle. To zoom out, click the right mouse button and use the contextual menu.

To go back and display all the images, select the **DISPLAY ALL** button.

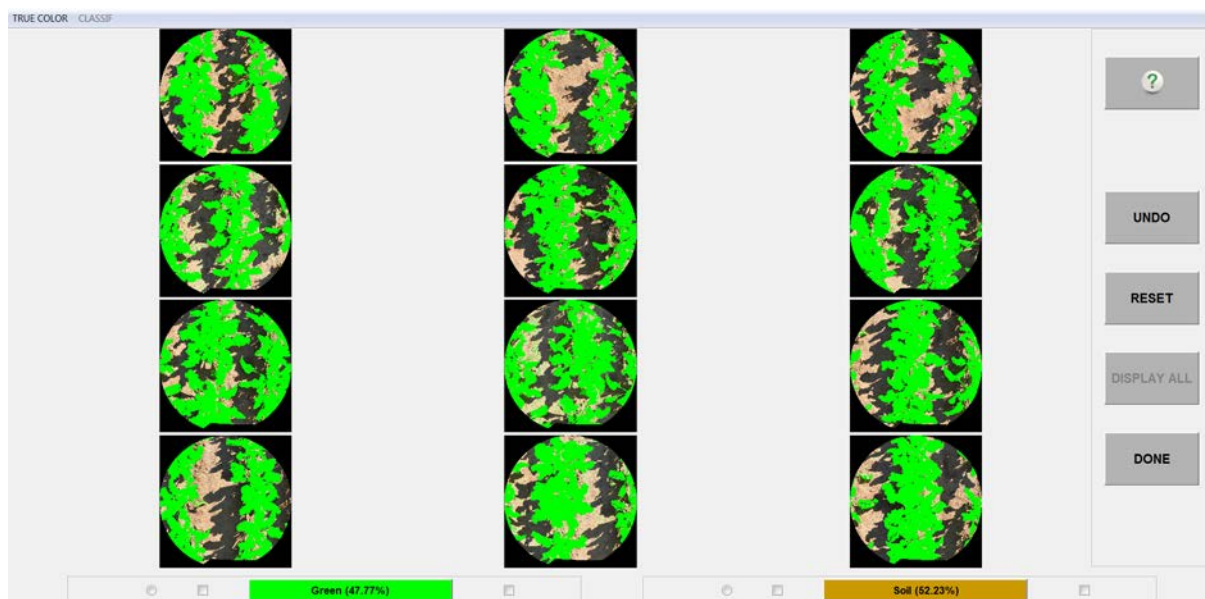


Figure 19. Classification window

### Toolbar MENU (top left)

- **TRUE COLOR:** the original images are displayed.
- **CLASSIF:** the original images are displayed with the classification colors: pixels already classified are shown in their class color, except if they are mixed (2 classes + mixed pixel classification method) or if they belong to the second class (classification method with no mixed pixels).

### Class MENU (bottom)

This menu is composed of two parts, one dedicated to the vegetation (left), and one to the background (right).

- **Class name:** Each class name is provided on a button with the percentage of pixels classified in this class for all the images together displayed. The background color of the button indicates the color in which the pixels belonging to this class will be shown (when the CLASSIF menu is active on the toolbar). Clicking on this button allows changing the color attributed to the class to ease the visualization for the classification. Select the color of your choice with the left mouse button.



Figure 20. Choice of a color for a class

- **Round radio button on the left of the class name:** select this button, and then pick up one or several pixels that belongs to this class with the left mouse button. To end the selection, use the right button. **All** the pixels of **all** the images that have the same color as the ones selected by the user are then also classified in the chosen class. The images are updated and classified pixels appear in the color of the selected class (when the CLASSIF menu is active on the toolbar). The percentage of classified pixels is also updated.
- **Square radio button on the left of the class name:** select this button, and then draw a polygon to delimit an area that belongs to this class using the left mouse button. End the selection with the right mouse button. Then all the pixels included in the polygon will belong to this class, as well as all the pixels in the other images that have the same color as one of these pixels. The image is updated and classified pixels appear in the color of the selected class (when the CLASSIF menu is active on the toolbar). If the interactive classification method is no mixed pixels, note that one class will be displayed in a given color while the other one will not be displayed (Figure 21).
- **Square radio button on the right of the class name:** select this button, and then draw a polygon to delimit an area that belongs to this class using the left mouse button. End the selection with the right mouse button. Then all the pixels included in the polygon will belong to this class, but, conversely to the square radio button on the left, this will have no impact outside this polygon and on the other images. The image is updated and classified pixels appear in the color of the selected class (when the CLASSIF menu is active on the toolbar). This may be useful for example when some over exposed parts of leaves are classified as sky (since they appear very bright as the sky in other part of the image) while the user knows that these pixels belong to leaves. This option should be used sparingly since it increases the window refreshing time during the classification process.

### Class pre-selection (first click on any radio button)

To speed up the classification process, when clicking the first time (or after resetting the classification process) on one of the radio buttons located around the class names, an automatic pre-classification based on color indices is proposed. CAN\_EYE uses a thresholding technique based 3 common color indices (ExcessGreen, ExcessBlue for sky background and ExcessRed for soil background) to pre-compute the classification (Meyer and Neto, 2008). The user can refine this pre-classification by using the sliders for each index in the Class Preselection window (Figure 21). The result is automatically displayed on the image. The percentage of pixels belonging to each class are updated once the OK button is selected.

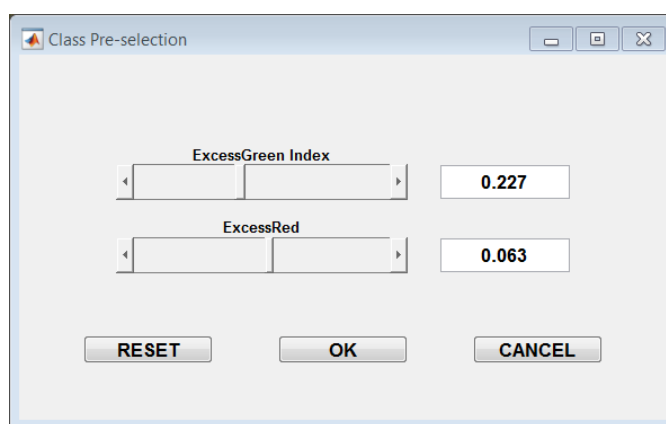
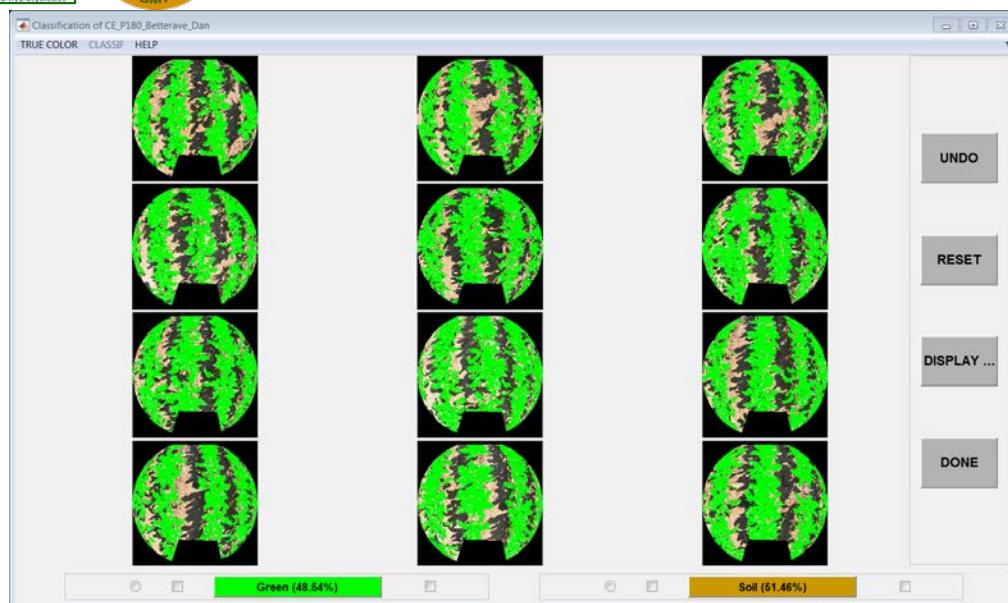


Figure 21. Using vegetation color indices to pre-classify the vegetation (No mixed pixel classification method: classified vegetation pixels appear in light green. The other pixels are considered to belong to the soil).

## Main MENU (right)

- ? : display the help page on this step.
- UNDO : undo last pixel selection
- RESET: reset all the classification process
- DISPLAY ALL: this button is enable when only one image (among all the images that are processed at the same time) is displayed in the center of the windows. It allows displaying all the images at the same time
- DONE: click on this button when the classification process is achieved. CAN\_ETE then checks if the user really want to end the classification process. Selecting yes implies that CAN\_EYE will binarise the images and compute all the outputs. Some figure showing the results are shortly displayed on the screen and are saved in the output directory.

## 4. CAN\_EYE OUTPUT DESCRIPTION

The following sections describe CAN\_EYE outputs as well as the theoretical background allowing the estimations. Table 1 presents the variables that CAN\_EYE derives from the set of digital images.

Variable	Acronym	DHP	P57	NADIR
<a href="#">Effective Plant Area Index estimated from <math>P_0(57^\circ)</math></a>	PAI57	x	x	
<a href="#">Effective Plant area index</a>	PAIeff	x		
<a href="#">Effective average plant inclination angle</a>	ALAeff	x		
<a href="#">True plant area index</a>	PAItrue	x	x	
<a href="#">True average leaf inclination angle</a>	ALA true	x		
<a href="#">Clumping Factor</a>	CF	x		
<a href="#">Cover Fraction</a>	FCOVER	x		x
<a href="#">Instantaneous 'black sky' FAPAR</a>	FAPAR <sup>BS</sup>	x		
<a href="#">White sky FAPAR</a>	FAPAR <sup>WS</sup>	x		
<a href="#">Daily black Sky FAPAR</a>	fAPAR <sup>BS</sup> <sub>Day</sub>	x		

**Table 1. CAN\_EYE output variable acronyms that can be derived from different acquisitions: hemispherical images, images at 57° and images acquired at nadir. Click on the link to directly access to the paragraph describing the variable computation.**

### 4.1. Definitions and theoretical background

#### *Introduction*

Leaf area index indirect measurement techniques are all based on contact frequency (Warren-Wilson, 1959) or gap fraction (Ross, 1981) measurements. Contact frequency is the probability that a beam (or a probe) penetrating inside the canopy will come into contact with a vegetative element. Conversely, gap frequency is the probability that this beam will have no contact with the vegetation elements until it reaches a reference level (generally the ground). The term “gap fraction” is also often used and refers to the integrated value of the gap frequency over a given domain and thus, to the quantity that can be measured, especially using hemispherical images. Therefore, measuring gap fraction is equivalent to measuring transmittance at ground level, in spectral domains where vegetative elements could be assumed black. It is then possible to consider the mono-directional gap fraction which is the fraction of ground observed in a given viewing direction (or in a given incident direction).

The objective of this section is to provide the theoretical background used in the CAN\_EYE software to derive canopy biophysical variables from the bi-directional gap fraction measured from the hemispherical images.

#### *Modeling the Gap Fraction*

##### LAI definition

The leaf area density,  $l(h)$  at level  $h$  in the canopy is defined as the leaf area per unit volume of canopy. The leaf area index (LAI) corresponds to the integral of  $l(h)$  over canopy height. It is therefore defined as the one sided leaf area per unit horizontal ground surface area

(Watson, 1947). Although this definition is clear for flat broad leaves, it may cause problems for needles and non-flat leaves. Based on radiative transfer considerations, Lang (1991) and Chen and Black (1992) and Stenberg (2006) proposed to define  $LAI$  as half the total developed area of leaves per unit ground horizontal surface area. This definition is therefore valid regardless vegetation element shape.

As defined above, leaf area index,  $LAI$ , defined as at a level  $H$  in the canopy is related to the leaf area density through:

$$\text{Eq. 1} \quad LAI = \int_0^H l(h) dh$$

### From LAI to Gap Fraction

Following Warren-Wilson (1959), the mean number of contacts  $N(H, \theta_v, \varphi_v)$  between a light beam and a vegetation element at a given canopy level  $H$  in the direction  $(\theta_v, \varphi_v)$  is:

$$\text{Eq. 2} \quad N(H, \theta_v, \varphi_v) = \int_0^H G(h, \theta_v, \varphi_v) l(h) / \cos \theta_v dh$$

where  $G(h, \theta_v, \varphi_v)$  is the projection function, i.e. the mean projection of a unit foliage area at level  $h$  in direction  $(\theta_v, \varphi_v)$ . When the leaf area density and the projection function are considered independent of the level  $h$  in the canopy, Eq. 2 simplifies in Eq. 3:

$$\text{Eq. 3} \quad N(L, \theta_v, \varphi_v) = G(\theta_v, \varphi_v) \cdot LAI / \cos \theta_v$$

The projection function is defined as follows:

$$\text{Eq. 4} \quad \begin{cases} G(\theta_v, \varphi_v) = \frac{1}{2\pi} \int_0^{2\pi} \int_0^{\pi/2} |\cos \psi| g(\theta_l, \varphi_l) \sin \theta_l d\theta_l d\varphi_l & (a) \\ \cos \psi = \cos \theta_v \cos \theta_l + \sin \theta_v \sin \theta_l \cos(\varphi_v - \varphi_l) & (b) \end{cases}$$

where  $g(\theta_l, \varphi_l)$  is the probability density function that describes leaf orientation distribution function. This induces the two normalization conditions given in Eq. 5a and Eq. 5b.

$$\text{Eq. 5} \quad \begin{cases} \frac{1}{2\pi} \int_0^{2\pi} \int_0^{\pi/2} g(\theta_l, \varphi_l) \sin \theta_l d\theta_l d\varphi_l = 1 & (a) \\ \frac{1}{2\pi} \int_0^{2\pi} \int_0^{\pi/2} G(\theta_v, \varphi_v) \sin \theta_v d\theta_v d\varphi_v = \frac{1}{2} & (b) \end{cases}$$

The contact frequency is a very appealing quantity to indirectly estimate LAI because no assumptions on leaf spatial distribution, shape, and size are required. Unfortunately, the contact frequency is very difficult to measure in a representative way within canopies. This is the reason why the gap fraction is generally preferred. In the case of a random spatial distribution of infinitely small leaves, the gap fraction  $P_0(\theta_v, \varphi_v)$  in direction  $(\theta_v, \varphi_v)$  is related to the contact frequency by:

$$\text{Eq. 6} \quad P_0(\theta_v, \varphi_v) = e^{-N(\theta_v, \varphi_v)} = e^{-G(\theta_v, \varphi_v) \cdot LAI / \cos(\theta_v)}$$

This is known as the Poisson model. Conversely to the contact frequency that is linearly related to LAI, the gap fraction is highly non linearly related to LAI. Nilson (1971) demonstrated both from theoretical and empirical evidences that the gap fraction can generally be expressed as an exponential function of the leaf area index even when the



random turbid medium assumptions associated to the Poisson model are not satisfied. In case of clumped canopies, a modified expression of the Poisson model can be written:

$$\text{Eq. 7} \quad P_0(\theta_v, \varphi_v) = e^{-\lambda_0 \cdot G(\theta_v, \varphi_v) \cdot LAI / \cos(\theta_v)}$$

where  $\lambda_0$  is the clumping parameter ( $\lambda_0 < 1$ ).

## **Modeling the leaf inclination distribution function** $g(l, \theta_l, \varphi_l)$

As shown previously, the gap fraction is both related to the leaf area index and the leaf inclination distribution function (LIDF). It is thus necessary to model the leaf inclination distribution function. The azimuthal variation of the LIDF is often assumed uniform and this is the case in the CAN\_EYE software, i.e. the probability density function  $g(\theta_l, \varphi_l)$  depends only on the leaf normal zenith angle. This assumption is verified in many canopies but may be problematic for heliotropic plants like sunflowers (Andrieu and Sinoquet, 1993).

Among existing models, the ellipsoidal distribution is very convenient and widely used (Campbell, 1986; Campbell, 1990; Wang and Jarvis, 1988): leaf inclination distribution is described by the ratio of the horizontal to the vertical axes of the ellipse that is related to the average leaf inclination angle (ALA variable in CAN\_EYE) knowing that  $\overline{\theta_l} = \frac{2}{\pi} \int_0^{\pi/2} g(\theta_l) \theta_l d\theta_l$  and that  $g(\theta_l)$  is the probability density function that verifies the normalization condition (Eq. 5).

## ***Estimating leaf area index and leaf inclination from gap fraction measurements***

### **Use of a single direction: LAI57**

Considering the inclined point quadrat method, Warren-Wilson (1960) has proposed a formulation of the variation of the contact frequency as a function of the view zenith and foliage inclination angles. Using this formulation, Warren-Wilson (1963) showed that for a view angle of  $57.5^\circ$  the G-function (Eq 4) can be considered as almost independent on leaf inclination ( $G = 0.5$ ). Using contact frequency at this particular  $57.5^\circ$  angle, Warren-Wilson (1963) derived leaf area index independently from the leaf inclination distribution function within an accuracy of about 7%. Bonhomme et al., (1974) applied this technique using the gap fraction measurements and found a very good agreement between the actual and estimated LAI values for young crops.

Therefore, for this particular viewing direction, LAI can be easily deduced from gap fraction:

$$\text{Eq 8} \quad Po(57.5^\circ) = \exp(-0.5LAI/\cos(57.5^\circ)) \Leftrightarrow LAI = \frac{-\ln(Po(57.5^\circ))}{0.93}$$

The CAN\_EYE software proposes an estimate of the LAI derived from this equation, called LAI57.

### **Use of multiple directions: LAIeff, ALAeff**

Among the several methods described in Weiss et al (2004), the LAI estimation in the CAN\_EYE software is performed by model inversion since, conversely to the use of the Miller's formula, it can take into account only a part of the zenith angle range sampled by hemispherical images. This is very useful since there is a possibility to reduce the image field

of view to less than 90° zenith. This feature is very important due to the high probability of mixed pixels in the part of the image corresponding to large zenith view angles. LAI and ALA are directly retrieved by inverting in CAN\_EYE using Eq 6 and assuming an ellipsoidal distribution of the leaf inclination using look-up-table techniques (Knyazikhin et al., 1998; Weiss et al., 2000). A large range of random combinations of *LAI* (between 0 and 10, step of 0.01) and *ALA* (10° and 80°, step of 2°) values is used to build a database made of the corresponding gap fraction values (Eq 6) in the zenithal directions defined by the CAN\_EYE user (parameter window definition during the CAN\_EYE processing). The process consists then in selecting the LUT element in the database that is the closest to the measured  $P_o$ . The distance (cost function  $C_k$ ) of the  $k^{\text{th}}$  element of the LUT to the measured gap fraction is computed as the sum of two terms:

CAN\_EYE V5.1 : Eq. 7

$$J_k = \sqrt{\underbrace{\frac{\sum_{i=1}^{Nb\_Zenith\_Dir} w_i (P_o^{LUT(k)}(\theta_i) - P_o^{MES}(\theta_i))^2}{\sigma_{MOD}(P_o^{MES}(\theta_i))}}_{\text{First Term}}} + \underbrace{\left( \frac{ALA^{LUT(k)} - 60}{30} \right)^2}_{\text{Second Term}}$$

CAN\_EYE V6.1: Eq. 8

$$J_k = \sqrt{\underbrace{\frac{\sum_{i=1}^{Nb\_Zenith\_Dir} w_i (P_o^{LUT(k)}(\theta_i) - P_o^{MES}(\theta_i))^2}{\sigma_{MOD}(P_o^{MES}(\theta_i))}}_{\text{First Term}}} + \underbrace{\left( \frac{PAI^{LUT(k)} - PAI^{57}}{\sigma_{PAI57}} \right)^2}_{\text{Second Term}}$$

The first term computes a weighted relative root mean square error between the measured gap fraction and the LUT one. The weights  $w_i$  take into account the fact that some zenithal directions may contain a lot of masked pixel and therefore, the corresponding gap fraction may not be very representative of the image:

$$\text{Eq. 9} \quad w_i = \frac{NPix_i - Nmask_i}{NPix_i}, \quad \sum_{i=1}^{Nb\_Zenith\_Dir} w_i = 1$$

The relative root mean square error is divided by a “modelled” standard deviation of the measured gap fraction derived from the empirical values  $\sigma(P_o^{MES}(\theta_i))$  computed from the images corresponding to the same plot for each zenithal direction  $I$ , when estimating the measured gap fraction after the CAN\_EYE classification step. In order to smooth  $\sigma$  zenithal variations, a second order polynomial is fitted on  $\sigma(P_o^{MES}(\theta_i))$  to provide  $\sigma_{MOD}(P_o^{MES}(\theta_i))$ .

The second term of Eq. 7 and Eq. 8 is the regularization term (Combal et al, 2002), that imposes constraints to improve the PAI estimates. Two equations are proposed:

- Constraint used in CAN\_EYE V5.1 on the retrieved ALA values that assumes an average leaf angle close to 60° ±30°.
- Constraint used in CAN\_EYE V6.1 on the retrieved PAI value that must be close from the one retrieved from the zenithal ring at 57°. This constraint is more efficient and not does not suppose any assumption than in Eq 7 but it can be computed only when the 57° ring is available (COI ≥ 60°)

The LUT gap fraction that provides the minimum value of  $J_k$  is then considered as the solution. The corresponding LAI and ALA provide the estimate of the measured CAN\_EYE leaf area index and average leaf inclination angle. As there is no assumption about clumping in the expression of the gap fraction used to simulate the LUT (Eq. 6), the foliage is assumed randomly distributed, which is generally not the case in actual canopies. Therefore, retrieval of LAI based on the Poisson model and using gap fraction measurements will provide estimates of an effective LAI,  $LAI^{eff}$ , and corresponding average inclination angle  $ALA_{eff}$  that allows the description of the observed gap fraction assuming a random spatial distribution.

Note that CAN\_EYE also proposed other ways of computing  $PAI_{eff}$  and  $ALA_{eff}$ , using Miller's formula (Miller, 1967), which assumed that gap fraction only depends from view zenith angle :

$$\text{Eq. 10} \quad PAI = 2 \int_0^{\pi/2} -\ln(P_o(\theta_v)) \cos \theta_v \sin \theta_v d\theta_v$$

Welles and Norman (1991) proposed a practical method to compute the integral of Eq.10 from gap fraction measurements in several directions for the LAI2000 instrument. CAN\_EYE proposed the effective PAI estimates using both Miller and LAI2000 measurements. For LAI2000, the ring angular response is taken into account and the computation is made of 3, 4 and 5 rings (Weiss et al., 2004).

## From effective leaf area index to true LAI

The “true LAI”, that can be measured only using a planimeter with however possible allometric relationships to reduce the sampling (Frazer et al., 1997), is related to the effective leaf area index through:

$$\text{Eq. 11} \quad LAI^{eff} = \lambda_o LAI$$

where  $\lambda_o$  is the aggregation or dispersion parameter (Nilson 1971; Lemeur and Blad, 1974) or clumping index (Chen and Black, 1992). It depends both on plant structure, *i.e.* the way foliage is located along stems for plants and trunks branches or shoots for trees, and canopy structure, *i.e.* the relative position of the plants in the canopy. The shape and size of leaves might also play an important role on the clumping.

In CAN\_EYE, the clumping index is computed using the Lang and Yueqin (1986) logarithm gap fraction averaging method. The principle is based on the assumption that vegetation elements are locally assumed randomly distributed. Each zenithal ring is divided into groups (called cells) of individual pixels. The size of the individual cells must compromise between two criterions: it should be large enough so that the statistics of the gap fraction are meaningful and small enough so that the assumption of randomness of leaf distribution within the cell is valid. For each cell,  $P_o$  is computed as well as its logarithm. If there is no gap in the cell (only vegetation, *i.e.*  $P_o=0$ ),  $P_o$  is assumed to be equal to a  $P_o^{sat}$  value derived from simple Poisson law, using a prescribed  $LAI^{sat}$  value.  $P_o^{cell}(\theta)$ , as well as  $\ln(P_o^{cell}(\theta))$  are then averaged over the azimuth and over the images for each zenithal ring. The averaging still takes into account the masked areas using  $w_i$ . The ratio of these two quantities provides the clumping parameter CF or  $\lambda_o$  for each zenithal ring:

$$CF = \lambda_o(\theta, ALA^{eff}) = \frac{\text{mean}[\log(P_o^{Cell}(\theta))]}{\log[\text{mean}(P_o^{Cell}(\theta))]}$$

Note that since  $P_o^{sat}$  is simulated using the Poisson model, it depends on the value chosen for both  $LAI^{sat}$  and the average leaf inclination angle, the clumping parameter is computed for the whole range of variation of ALA and a  $LAI^{sat}$  varying between 8 and 12 (Note that all the results in the CAN\_EYE html report are provided for  $LAI^{sat} = 10$ ). Then the same algorithm, as described previously for effective LAI (§0), is applied by building a LUT using the modified Poisson model (eq 7) to provide  $LAI^{true}$  and  $ALA^{true}$  as well as the corresponding clumping parameter.

## LAI or PAI?

Claiming that devices and associated methods based on gap fraction measurements provide an estimate of the leaf area index is not right since indirect measurements only allow assessing plant area index. Further, it is not possible to know if some leaves are present behind the stems, branches or trunk. Therefore, masking some parts of the plants (which is possible using CAN\_EYE) to keep only the visible leaves is not correct and could lead to large under-estimation of the actual LAI value, depending on the way leaves are grouped with the other parts of the plant. Therefore, all CAN\_EYE outputs correspond to plant area index and not leaf area index.

Note also that the term GAI (Green Area Index: total area of green elements within the canopy) can also be used if only the green elements are selected during the classification process.

## *Cover fraction computation*

Cover fraction (fCover) is defined as the fraction of the soil covered by the vegetation viewed in the nadir direction:

Eq 12.  $fCover = 1 - P_o(0)$

Using hemispherical images, it is not possible to get a value in the exact nadir direction, and the cover fraction must be integrated over a range of zenith angles. In CAN\_EYE, the default value for this range is set to 0-10°. The user can change this value when defining the CAN\_EYE parameters (which also concerns the description of the hemispherical lens properties) at the beginning of the processing.

## *FAPAR computation*

FAPAR is the fraction of absorbed photosynthetically active radiation (400-700nm) by the vegetation. It varies with sun position. As there is little scattering by leaves in that particular spectral domain due to the strong absorbing features of the photosynthetic pigments (Andrieu and Baret, 1993), fAPAR is often assumed to be equal to fIPAR (fraction of Intercepted photosynthetically active radiation), and therefore to the gap fraction. The actual fAPAR is the sum of two terms, weighted by the diffuse fraction in the PAR domain: the 'black sky' fAPAR that corresponds to the direct component (collimated beam irradiance in the sun direction only) and the 'white sky' or the diffuse component. The closest approximation to white sky fAPAR occurs under a deep cloud cover that may generate an almost isotropic

diffuse downward. Following Martonchik et al {, 2000 #578), the adjectives black and white are not related to the color of the sky, but rather to the angular distribution of light intensity. Providing the latitude and the date of the image acquisition, the CAN\_EYE software proposes three outputs for FAPAR:

1. The instantaneous 'black sky' FAPAR ( $FAPAR^{BS}$ ): it is the black sky FAPAR at a given solar position (date, hour and latitude). Depending on latitude, CAN\_EYE computes the solar zenith angle every solar hour during half the day (there is symmetry at 12:00). The instantaneous FAPAR is then approximated at each solar hour as the gap fraction in the corresponding solar zenith angle:

$$fAPAR^{BS}(\theta_s) = 1 - P_o(\theta_s)$$

2. The daily integrated black sky (or direct) FAPAR is computed as the following:

$$fAPAR_{Day}^{BS} = \frac{\int_{sunrise}^{sunset} \cos(\theta)(1 - P_o(\theta))d\theta}{\int_{sunrise}^{sunset} \cos(\theta)d\theta}$$

3. The white sky (or diffuse) FAPAR is computed as the following:

$$fAPAR^{WS} = \frac{1}{\pi} \int_0^{2\pi} \int_0^{\pi/2} (1 - P_o(\theta)) \cos \theta \sin \theta d\theta d\varphi = 2 \int_0^{\pi/2} (1 - P_o(\theta)) \cos \theta \sin \theta d\theta$$

## 4.2. Description of CAN\_EYE output directory content

During the processing, a CAN\_EYE output subdirectory is automatically generated in the directory containing the images to be processed. It contains:

- a html report of the processing: screen copies of the different steps (Masking, Classification, Resulting graphs), a table with the processing characteristics and a table containing the CAN\_EYE output variable estimates.
- An excel or ascii file (depending on the CAN\_EYE characterization parameters, see section 3.2) providing all the output variable estimates.

### *Hemispherical images*

DHP results are contained in a subdirectory called 'CE\_P180\_xxx', where xxx is the name of the directory of the images that have been processed.

**The html report** is named 'CE\_P180\_Report\_xxx.html'. It contains 10 sections:

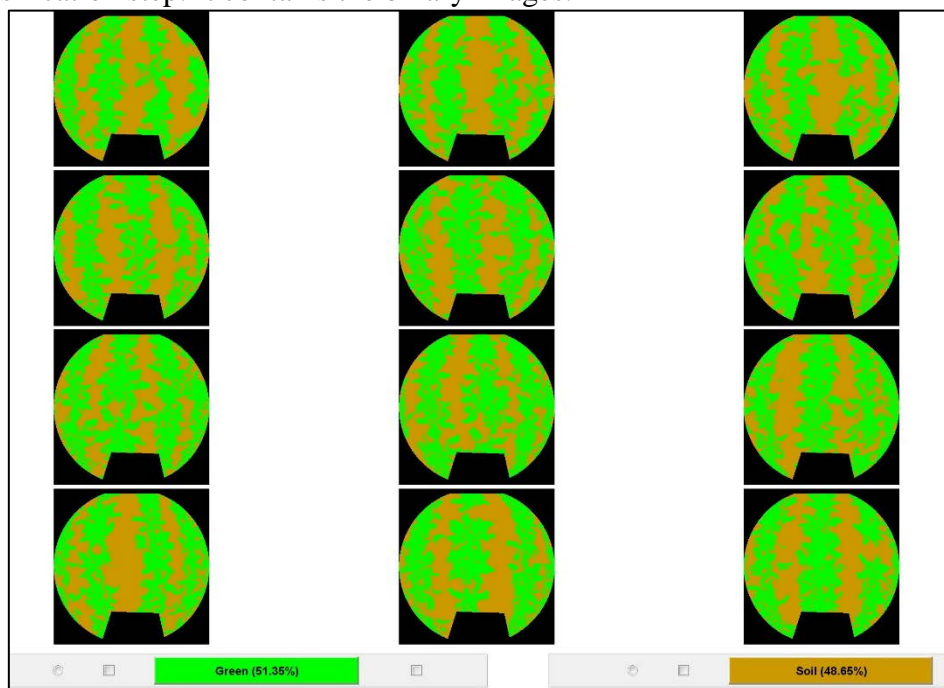
- GENERAL INFORMATION (CAN\_EYE version, User, processing Date, Processing Duration)
- GENERAL PARAMETERS: Angular resolution, Circle of Interest, FCOVER integration domain, sub sampling factor, see §3.2
- CALIBRATION PARAMETERS: Optical Center, projection Function, Coefficients, see §3.2



- **SELECTED IMAGES:** table containing the list of the image file names used for the processing, as well as the corresponding FCOVER estimates
- **NUMBER OF CLASSES:** number of classes that was selected (2 or 2+ mixed, see section 3.5)
- **AVERAGE BIOPHYSICAL VARIABLES:** variables estimated from the series of images: PAI, ALA, FCOVER and FAPAR. Different PAI and ALA estimates (effective, true) from different methods (CAN\_EYEV5.1, CAN\_EYE V6.1, Miller, LAI2000, 3,4,5 rings). Details of the computation and meaning of the variables is provided in section 4.1. Note that for the computation of PAI<sub>true</sub>, the value of PAI<sub>sat</sub> is set to the value given in the parameter window (see [LAI<sub>true</sub>](#)). The percentage of saturated cells (considered as corresponding to a PAI of PAI<sub>sat</sub>) used for the computation of the clumping index and true PAI is also given as an indication of the sensitivity to PAI<sub>sat</sub>.

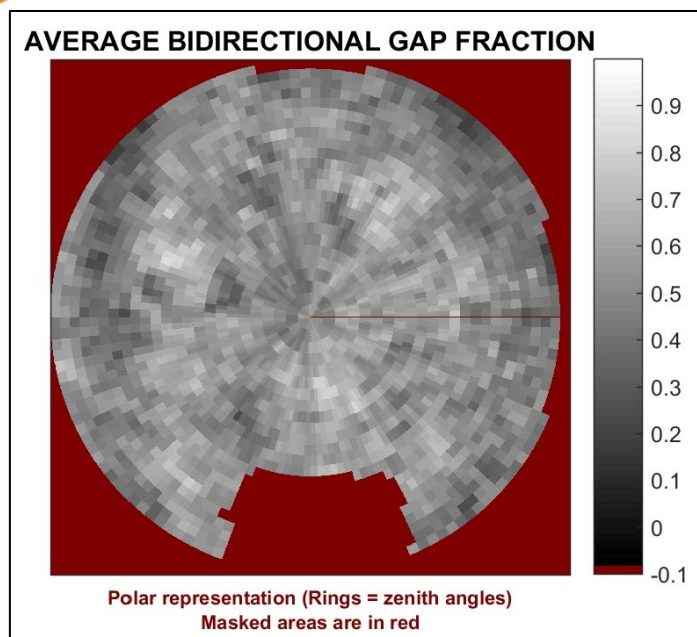
Note that if the FCOVER can be estimated for each image, this is not the case for the other variables that are derived from the mono-directional gap fraction averaged over all the processed images.

- **CLASSIFICATION RESULTS:** screen copy of the working window after the classification step. It contains the binary images.



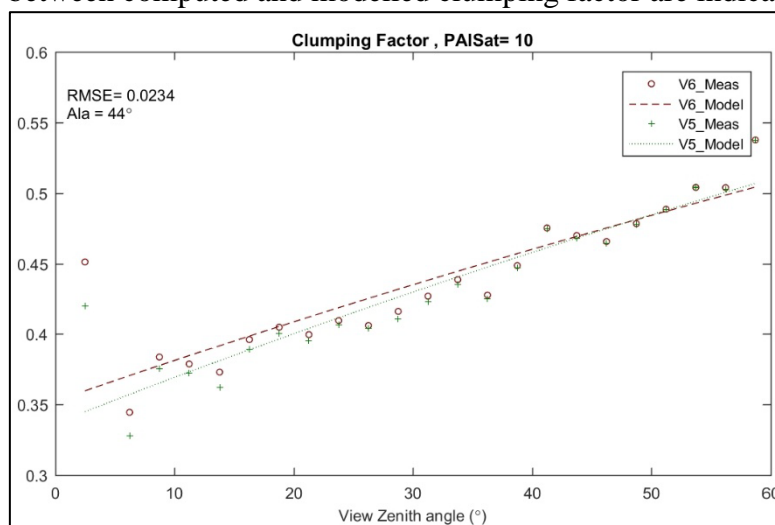
**Figure 22. Example of classification results**

- **AVERAGE GAP FRACTION:** polar plot of the average bi-directional gap fraction: rings correspond to zenithal directions. Masked areas are shown in red.



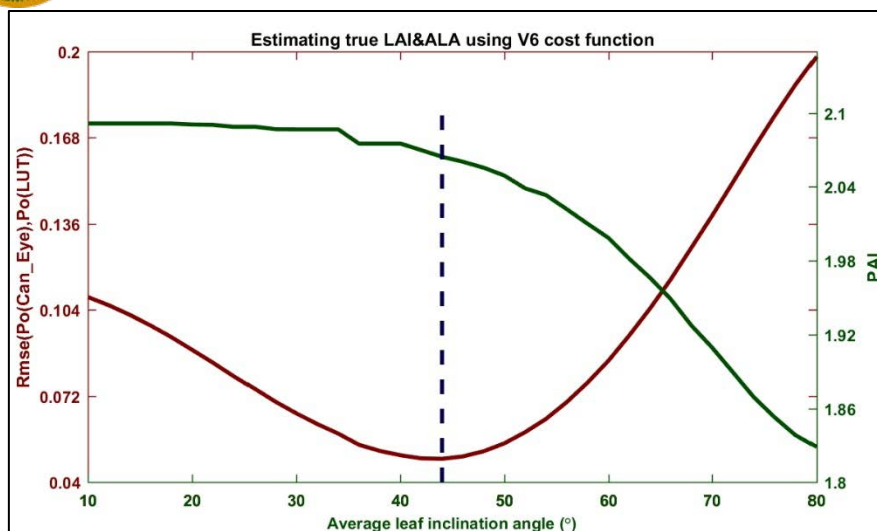
**Figure 23. Example of average gap fraction polar plot.**

- **CLUMPING FACTOR:** graph showing the comparison between the computed clumping factor (green points) and the modeled one (red line) as a function of view zenith angle for the computation of true PAI. Average leaf inclination value as well as the RMSE between computed and modelled clumping factor are indicated (see [here](#)).



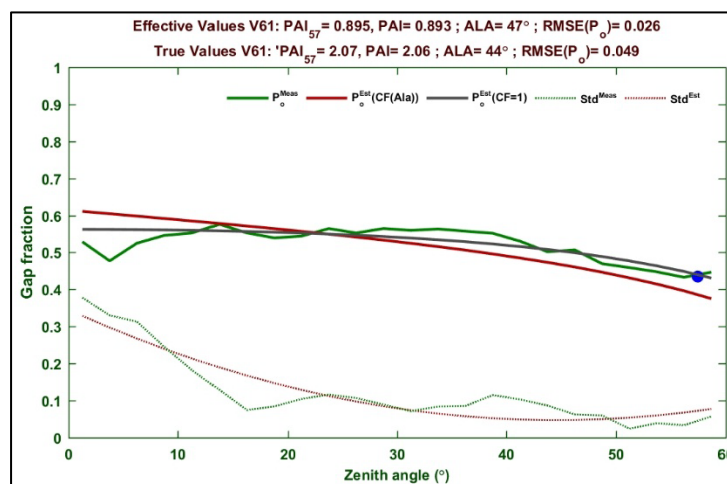
**Figure 24. Figure showing the results for the computed clumping factor (CAN-EYE version 5 in green and Version 6 in red). Model is shown in dashed lines while measured value (extracted from the binary images) is shown in symbols.**

- **MEASURED GAP FRACTION VS LUT GAP FRACTION & AVERAGE PAI, ALA, FCOVER:**
  - top graph shows the RMSE value between the mono-directional gap fraction computed from the images and the closest one found in the LUT (red line), as a function of average leaf inclination angle value. The green line shows the corresponding PAI value (that provides the lowest RMSE) as a function of average leaf inclination angle.



**Figure 25. Results of PAI and ALA estimation with CAN\_EYE. RMSE value between the mono-directional gap fraction computed from the images and the closest one found in the LUT (red line). Green line shows the corresponding value.**

- The bottom graph shows the mono-directional gap fraction measured from the images (determined by the classification step) as a function of view zenith angle (green line). The dark gray line indicates the mono-directional gap fraction of the LUT element that is the closest from the measurements (assuming no clumping effect, i.e, when estimating  $PAI_{eff}$  and  $ALA_{eff}$ ). In red, the same is shown for the mono-directional gap fraction when considering the clumping factor ( $PAI_{true}$ ,  $ALA_{true}$ ).



**Figure 26. Measured (green) and modelled gap fraction when considering (red) or not (dark grey) the clumping factor.**

**The excel file named** ‘CE\_P180\_Report\_xxx.xls’. Contains several sheets:

- CAN\_EYE P180 parameters: description of the parameters used for the processing (GENERAL PARAMETERS and CALIBRATION PARAMETERS in the html report)
- PAI, ALA: different PAI and ALA estimates (effective, true) from different methods (CAN\_EYEV5.1, CAN\_EYE V6.1, Miller, LAI2000, 3,4,5 rings). Details of the

computation and meaning of the variables is provided in section 4.1. Note that for the computation of PAI<sub>true</sub>, the value of PAI<sub>sat</sub> is set to 10 (see [LAI<sub>true</sub>](#)).

- FAPAR: daily integrated value of direct and diffuse FAPAR (measured and modelled), as well as instantaneous values
- Mono Po: averaged (over all the images) as well as individual mono-directional gap fraction as a function of view zenith angle
- P57 results: gap fraction at 57° and derived PAI for individual images and averaged over all the images. Note that PAI<sub>57°</sub> is an indicative value and should not be used since it does not fulfil the Poisson assumptions.
- Average Bidir Po: bi-directional gap fraction averaged over all the images as a function of view zenith (columns) and azimuth (rows) angles.
- Next sheets correspond to individual bio-directional gap fraction as a function of view zenith (columns) and azimuth (rows) angles. The sheet names corresponds to the image name.

### *Images acquired at 57°*

Images at 57° results are contained in a subdirectory called 'CE\_P57\_xxx', where xxx is the name of the directory that contains the images to be processed.

**The html report** is named 'CE\_P57\_Report\_xxx.html'. It contains 10 sections:

- GENERAL INFORMATION (CAN\_EYE version, User, processing Date, Processing Duration)
- CALIBRATION PARAMETERS: image size, resolution for clumping, camera model, CCD size (height and width), sub sampling factor, resolution for clumping
- SELECTED IMAGES: table containing the list of the image file names used for the processing, as well as the corresponding gap fraction at 57° estimates and corresponding PAI<sub>s</sub>
- NUMBER OF CLASSES: number of classes that were selected (2 or 2+ mixed, see section 3.5)
- PROCESSED IMAGES + MASK: screen copy of the working window after the masking step
- CLASSIFICATION RESULTS: screen copy of the working window after the classification step
- GAP FRACTION: Plot of P57 (black), effective (green) and true PAI (red) values as a function of individual image number. Note that for the computation of the true PAI, different value of PAI<sub>sat</sub> are used (see [LAI<sub>true</sub>](#)).

**The excel file named** 'CE\_P57\_Report\_xxx.xls'. Contains several sheets:

- CAN\_EYE P57 parameters: description of the parameters used for the processing (GENERAL PARAMETERS and CALIBRATION PARAMETERS in the html report)
- CAN\_EYE P57 Results: table with P57, effective and true PAI values

### *Images acquired at Nadir*

Images acquired at nadir results are contained in a subdirectory called 'CE\_NADIR\_xxx', where xxx is the name of the directory that contains the images to be processed.

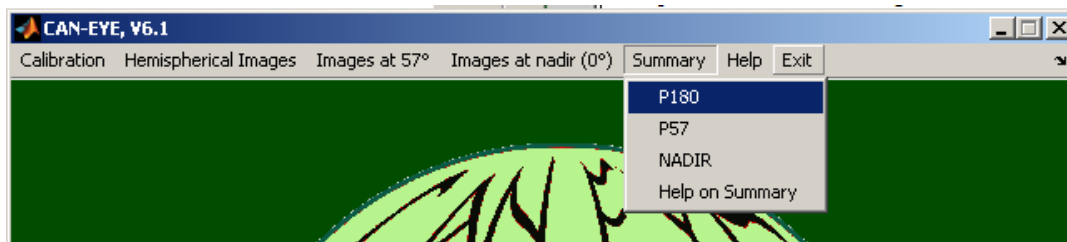
**The html report** is named 'CE\_NADIR\_Report\_xxx.html'. It contains 10 sections:

- GENERAL INFORMATION (CAN\_EYE version, User, processing Date, Processing Duration)
- CALIBRATION PARAMETERS: image size, camera model, sub sampling factor
- SELECTED IMAGES: table containing the list of the image file names used for the processing and the corresponding gap fraction at nadir estimates.
- NUMBER OF CLASSES: number of classes that were selected (2 or 2+ mixed, see section 3.5)
- PROCESSED IMAGES + MASK: screen copy of the working window after the masking step
- CLASSIFICATION RESULTS: screen copy of the working window after the classification step
- GAP FRACTION: bar plot of nadir gap fraction as a function of individual image number. Average cover fraction is shown in red.

**The excel file named** 'CE\_Nadir\_Report\_xxx.xls'. Contains several sheets:

- CAN\_EYE nadir Parameters: description of the parameters used for the processing (GENERAL PARAMETERS and CALIBRATION PARAMETERS in the html report)
- CAN\_EYE Nadir Results: table with average and individual FCOVER value

## 5. THE "SUMMARY" MODULE



**Figure 27. CAN\_EYE summary menu**

The Summary Menu allows gathering CAN\_EYE result processing Excel file in a single one. After selecting the processing type (DHP=P180, images acquired at 57°=P57 and images acquired at NADIR), the user is asked to choose a directory to proceed. The chosen directory should contain CAN\_EYE processing results sub-directories (all called CE\_P180\*\*\* or CE\_P57\*\*\* depending on the processing type). The user is then asked to provide a name for the summary excel file in which all the results will be stored.

### *Digital Hemispherical Images (DHP)*

Six sheets are generated

- Effective PAI: columns contents are, in order: directory name, CE\_P180 file name, CEV6.1 PAI estimate, CEV5.1 PAI estimate, Miller PAI estimate, LAI2000 3, 4 and five rings estimates, processing date.



- Effective ALA: columns contents are, in order: directory name, CE\_P180 file name, CEV6.1 ALA estimate, CEV5.1 ALA estimate, Miller ALA estimate, LAI2000 3, 4 and five rings estimates, processing date.
- True PAI: columns contents are, in order: directory name, CE\_P180 file name, CEV6.1 PAI estimate, CEV5.1 PAI estimate, Miller PAI estimate, LAI2000 3, 4 and five rings estimates, processing date.
- True ALA: columns contents are, in order: directory name, CE\_P180 file name, CEV6.1 ALA estimate, CEV5.1 ALA estimate, Miller ALA estimate, LAI2000 3, 4 and five rings estimates, processing date.
- FCOVER: columns contents are, in order: directory name, CE\_P180 file name, CEV6.1 FCOVER estimate, processing date
- Daily FAPAR: columns contents are, in order: directory name, CE\_P180 file name, measured direct FAPAR estimate, measured modelled FAPAR estimate, estimated direct FAPAR estimate, estimated direct FAPAR estimate, processing date

### *Images acquired at 57.5° (DP57)*

Two sheets are generated

- Mean: columns contents are, in order: directory name, CE\_P57 file name, average gap fraction value at 57°, effective PAI, True PAI for different PAISAT values (see [LAtrue](#)), Processing date
- columns contents are, in order: directory name, CEP57 file name, image name, individual gap fraction value at 57° (for the image name), effective PAI, True PAI for different PAISAT value (see [LAtrue](#)), Processing date

### *Images acquired at Nadir (DPO)*

Two sheets are generated

- Mean: columns contents are, in order: directory name, CE\_NADIR file name, average FCOVER value
- columns contents are, in order: directory name, CE\_NADIR file name, image name, individual FCOVER value for the corresponding image, Processing date

## 6. DHP "CALIBRATION" MODULE

Optical systems are not perfect and at least two main characteristics are required to perform an accurate processing of hemispherical images:

- The coordinates of the optical center
- The projection function. In CAN\_EYE, the projection function is assumed to be a polar projection: (angular distances (in degrees) in the object region are proportional to radial distances in pixel on the image plane).

Because in some situations the focal length may be manipulated by acting on the zoom system, the projection function must be also known for each focal length used.

A simple method is proposed by the CAN\_EYE calibration menu to characterize these features (Figure 28). It requires to fill in an excel file located in the "Calibration\_Data" directory. This file contains 4 sheets named "Start", "Optical Center", "Projection Function" and "Results". Inside each sheet, grey cells must remain unchanged while yellow cells must be filled the data corresponding to a given calibration. Note that, for a given camera+fish-eye system, it is required to make a calibration experiment for each image resolution that will be used with the fish-eye. In order to reduce the problem of mixed pixels, it is highly recommended to use the highest resolution providing jpg or TIF files.

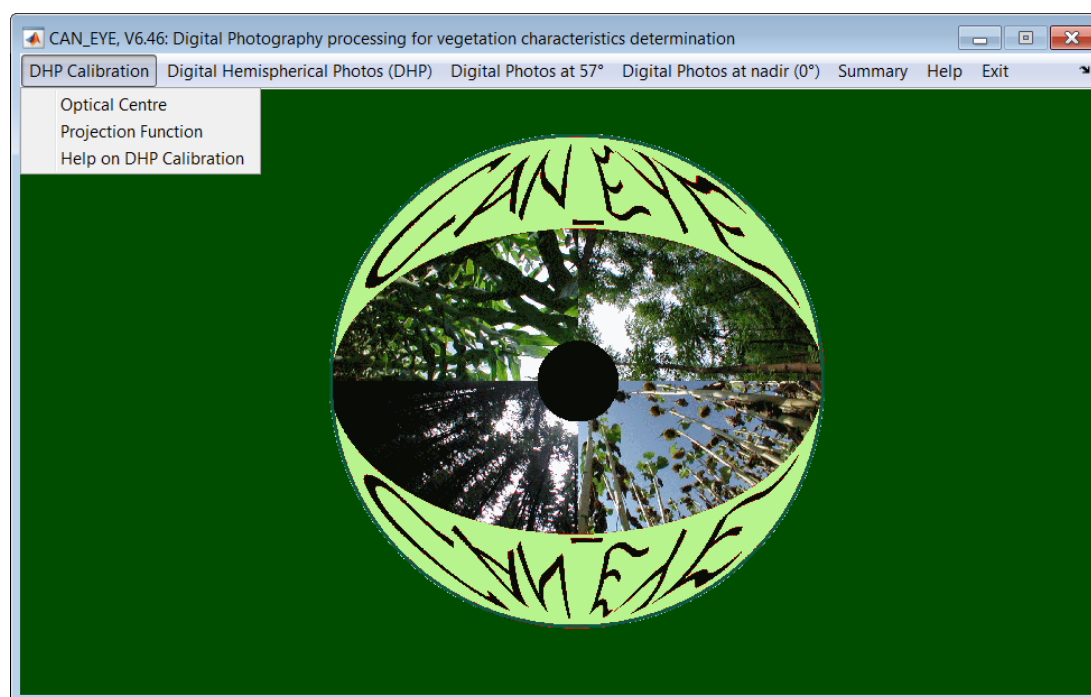


Figure 28. CAN\_EYE DHP calibration menu

The "Start" sheet describes the calibration experiment for traceability (Figure 29): Name of the user, Name of the CAMERA, Reference, and the image resolution in columns (X direction) and lines (Y resolution). This can be achieved by looking at an image file properties in the window explorer or with any image processing software.

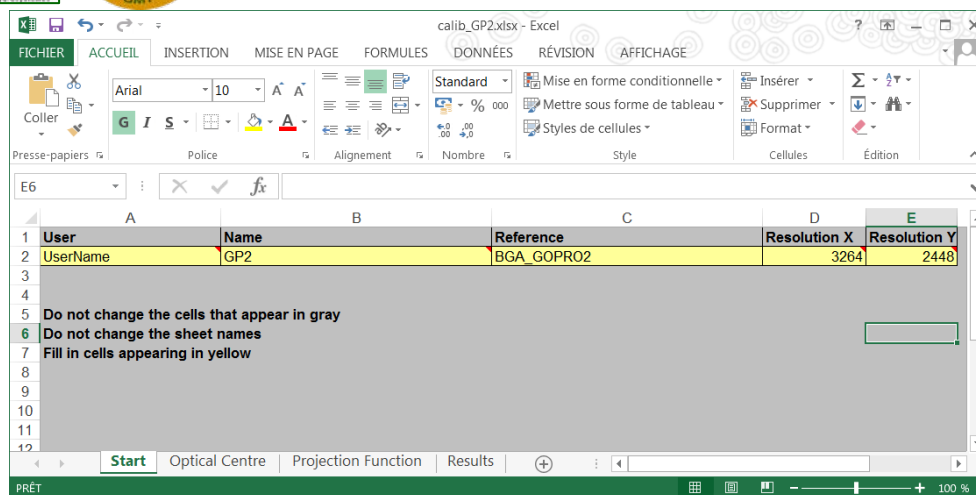


Figure 29. Example of the “Start “ sheet of the calibration excel file

The “Results” sheet contains the calibration parameter of your camera+fish-eye system derived from the calibration process: the optical center position as well as the maximum field of view (FOV in degrees) and the corresponding radius (in pixels) of the image. These values can be directly entered in the CAN\_EYE parameter window during the processing of hemispherical images (see [Filling CAN\\_EYE window parameter](#) ).

The contents of the two other sheets, as well as the principles of the measurement, are described in the following.

## 6.1. System Definition

An image can be defined by (Figure 30)

- The number of pixels in the horizontal direction ( $X_{size}$ )
- The number of pixels in the vertical direction ( $Y_{size}$ )
- The coordinates of the optical center [ $X_o$ ,  $Y_o$ ]
- The projection function, **P**, i.e. the function  $\theta = \mathbf{P}(\mathbf{R})$  that relates the view angle  $\theta$  (relative to optical axis) to the distance to the optical center **R**.

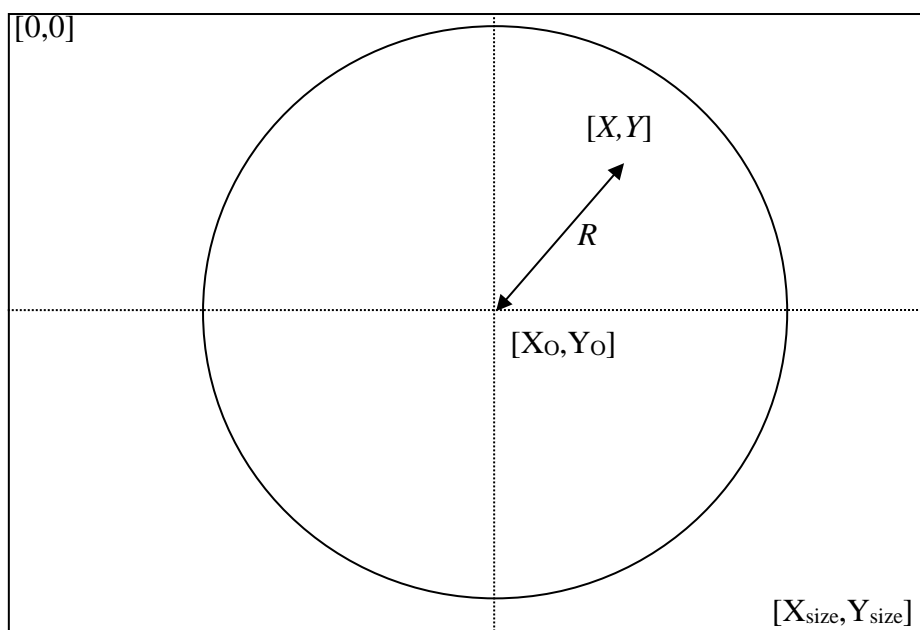


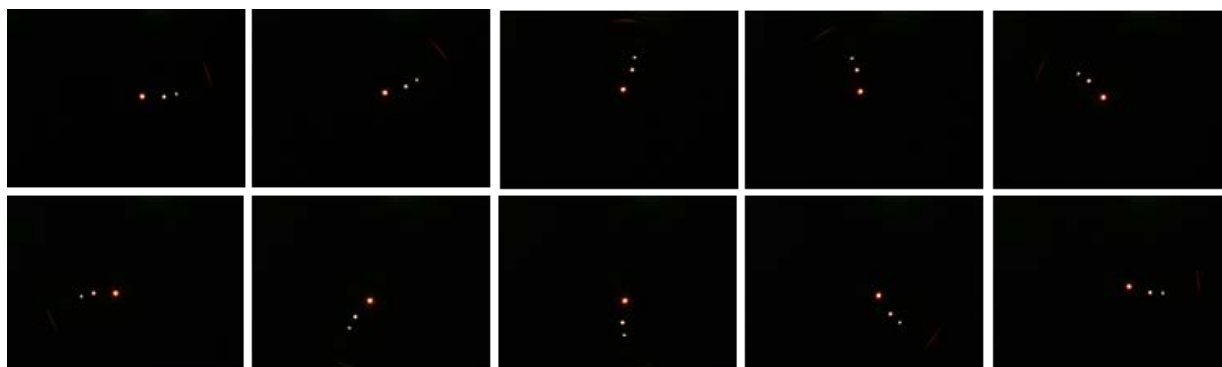
Figure 30. Image coordinate system.

## 6.2. Optical center characterization

The optical center is defined by the projection of the optical axis onto the CCD matrix where the image is recorded. This point should therefore be invariant by rotation of the system along this optical axis. A simple method to get the optical center consists in observing the coordinates of a point when it rotates along this axis. This could be achieved by drilling a small hole in the cap of the fish-eye, and acquiring photographs for a series of positions. This is illustrated by Figure 31. It is possible to use several holes to check the consistency of the estimation of the optical center Figure 32.



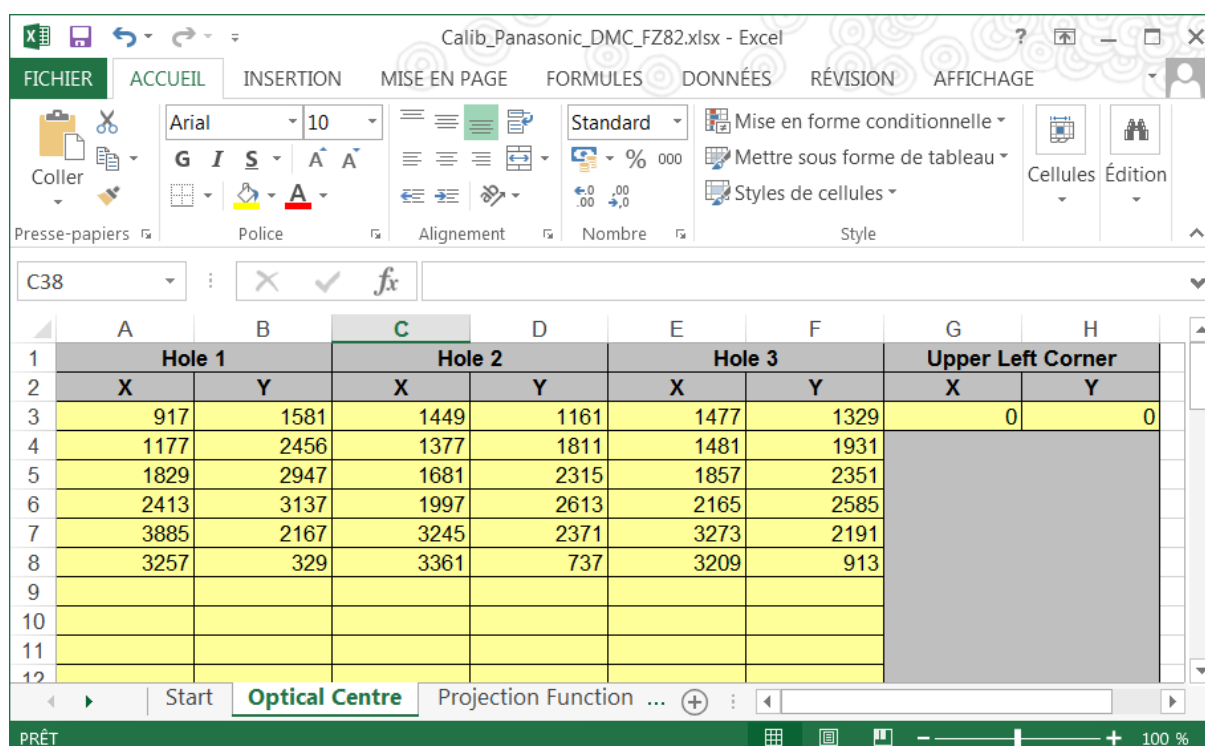
**Figure 31. Illustration of the holes drilled in the fish-eye cap. The red arrow indicates the rotation of the cap.**



**Figure 32. A series of images taken for several positions of the fish-eye cap. In this case, three holes were considered.**

The rotation positions must be if possible as symmetric as possible in order to minimize possible biases in the estimation of the optical center. Once the photos are acquired (10 to 20 positions are enough), the different holes (as well as the image upper left corner) coordinates must be extracted through an image processing software (e.g. Paintshop, Microsoft Photo Editor).

Then, the user has to fill the “Optical Center” sheet of the excel file (Figure 33). It consists in typing the 3 hole coordinates (in the X and Y directions) for the different positions of the fish-eye cap.



	Hole 1		Hole 2		Hole 3		Upper Left Corner	
	X	Y	X	Y	X	Y	X	Y
1	917	1581	1449	1161	1477	1329	0	0
2	1177	2456	1377	1811	1481	1931		
3	1829	2947	1681	2315	1857	2351		
4	2413	3137	1997	2613	2165	2585		
5	3885	2167	3245	2371	3273	2191		
6	3257	329	3361	737	3209	913		
7								
8								
9								
10								
11								
12								

**Figure 33 : Example of an excel file sheet to be filled to determine the optical center of a system.**

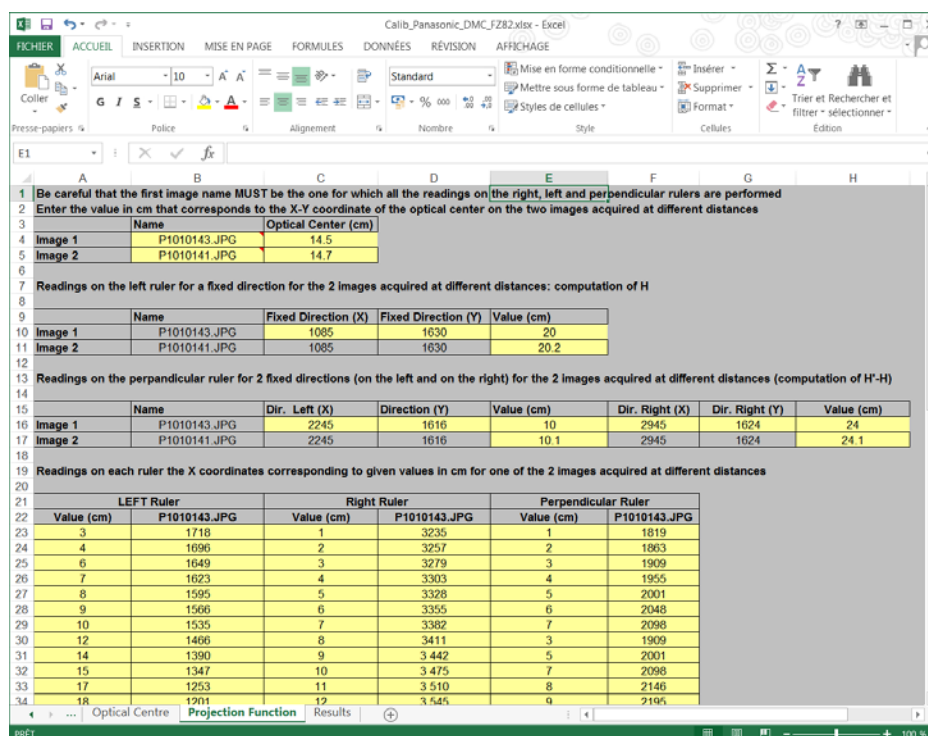


Once the “Optical Center” sheet is filled, the user must run CAN\_EYE, then go in the Calibration Menu and then select “Optical Center”. CAN\_EYE asks then to select the excel file, computes automatically the optical center position and fill the “Results” sheet.

Figure 36 shows an example of optical center fitting in the case where three holes were considered. Results show very consistent estimates of the coordinates of the optical center that is known with accuracy better than one pixel.

## 6.3. Projection function characterization

This section describes how to perform the measurements and fill the “Projection Function” sheet of the calibration excel file. The experimental design is described in Figure 35. It consists in a frame of 50\*50cm 1.5cm thick from which a 30\*30cm square is excavated from the middle of one of the side. The three sides of this gap are equipped with 30cm long rulers. **The camera is set horizontally as well as the experimental design. The camera is aligned along the main axis of the design using the front nail and background line.** Hemispherical photographs are taken at two distances ( $H$  and  $H'=H+\Delta$ ) from the center of the design and along the optical axis. The calibration excel file sheet is shown in Figure 35. Note that it is required to have run the “optical center” menu before being able to compute the projection function.



**1** Be careful that the first image name MUST be the one for which all the readings on the right, left and perpendicular rulers are performed

**2** Enter the value in cm that corresponds to the X-Y coordinate of the optical center on the two images acquired at different distances

	Name	Optical Center (cm)
Image 1	P1010143.JPG	14.5
Image 2	P1010141.JPG	14.7

**7** Readings on the left ruler for a fixed direction for the 2 images acquired at different distances: computation of H

	Name	Fixed Direction (X)	Fixed Direction (Y)	Value (cm)
Image 1	P1010143.JPG	1085	1630	20
Image 2	P1010141.JPG	1085	1630	20.2

**13** Readings on the perpendicular ruler for 2 fixed directions (on the left and on the right) for the 2 images acquired at different distances (computation of H'-H)

	Name	Dir. Left (X)	Direction (Y)	Value (cm)	Dir. Right (X)	Dir. Right (Y)	Value (cm)
Image 1	P1010143.JPG	2245	1616	10	2945	1624	24
Image 2	P1010141.JPG	2245	1616	10.1	2945	1624	24.1

**19** Readings on each ruler the X coordinates corresponding to given values in cm for one of the 2 images acquired at different distances

LEFT Ruler		Right Ruler		Perpendicular Ruler	
Value (cm)	P1010143.JPG	Value (cm)	P1010143.JPG	Value (cm)	P1010143.JPG
3	1718	1	3235	1	1819
4	1696	2	3257	2	1863
6	1649	3	3279	3	1909
7	1623	4	3303	4	1955
8	1595	5	3328	5	2001
9	1566	6	3355	6	2048
10	1535	7	3382	7	2098
12	1466	8	3411	8	2146
14	1390	9	3442	9	2196
15	1347	10	3475	10	2246
17	1253	11	3510	11	2296
18	1201	12	3545	12	2346

Figure 34. Example of a projection function sheet for the calibration excel file.

Let us assume that the two images are named Im1( $H$ ) and Im2 ( $H'$ ). Im1 must be the image for which ruler ticks are the most readable. Then, you have to look at the image using an image processing software (e.g. Paintshop, Microsoft Photo Editor) to read pixels coordinates in the image.

➤ Read the optical center position in cm on the rulers of Im1 and Im2 and fill cells B4, C4 and B5, C5 of the excel file (“Projection Function” sheet)

➤ The quantity  $\Delta$  can be easily measured by looking at one direction on the lateral ruler ( $X_{p1}, Y_{p1}$ , cells C10, D10) on Im1 reading the corresponding value  $h$  in cm (cell E10) for distance  $H$ . Then for the distance  $H+\Delta$ , the same point on Im2 corresponds to a value  $h'$  (cell E11) on the lateral ruler. It comes simply that  $\Delta = h' - h$ .

➤ On the perpendicular ruler, select two fixed directions ( $X_{p2}, Y_{p2}$ , cells C16, D16) and ( $X_{p3}, Y_{p3}$ , cells F16, G16) on Im1 read the corresponding values  $x$  in cm (cell E16 and H16) on the perpendicular ruler. Do the same for Im2 (cells E17, H17). It is then possible to compute the actual distance  $H$  if  $\Delta$  is known.

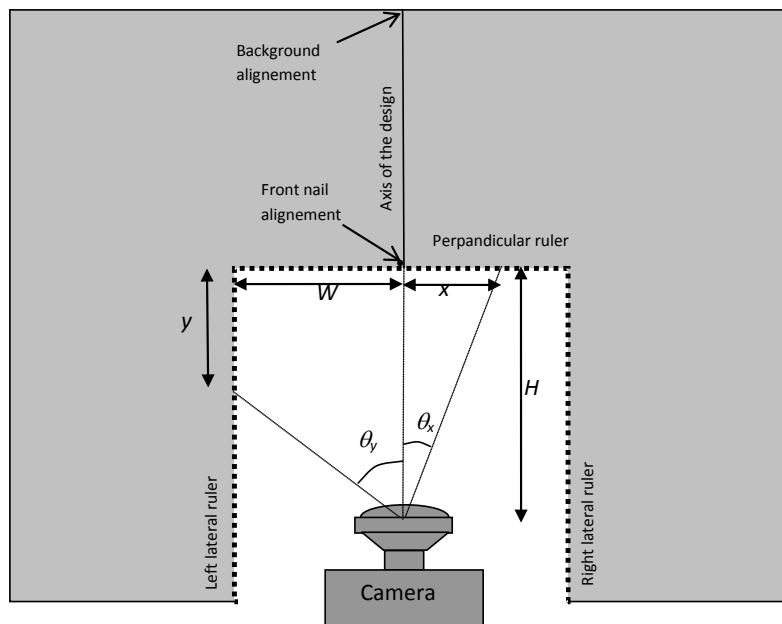
$$\begin{cases} \tan(\theta_x) = x / H \\ \tan(\theta_x) = x' / (H + \Delta) \end{cases} \Rightarrow H = \Delta \cdot x / (x - x')$$

➤ Once the distance  $H$  is known, the calibration of the projection function can be achieved if the coordinates on one of the 2 images (select the one that is the most readable) are associated to the actual distance read on the rulers. The coordinates have to be read on the line passing through the optical axis, for the three rulers. This can be achieved for each cm tick. The following equations are used to derive the angle from the values read on each ruler:

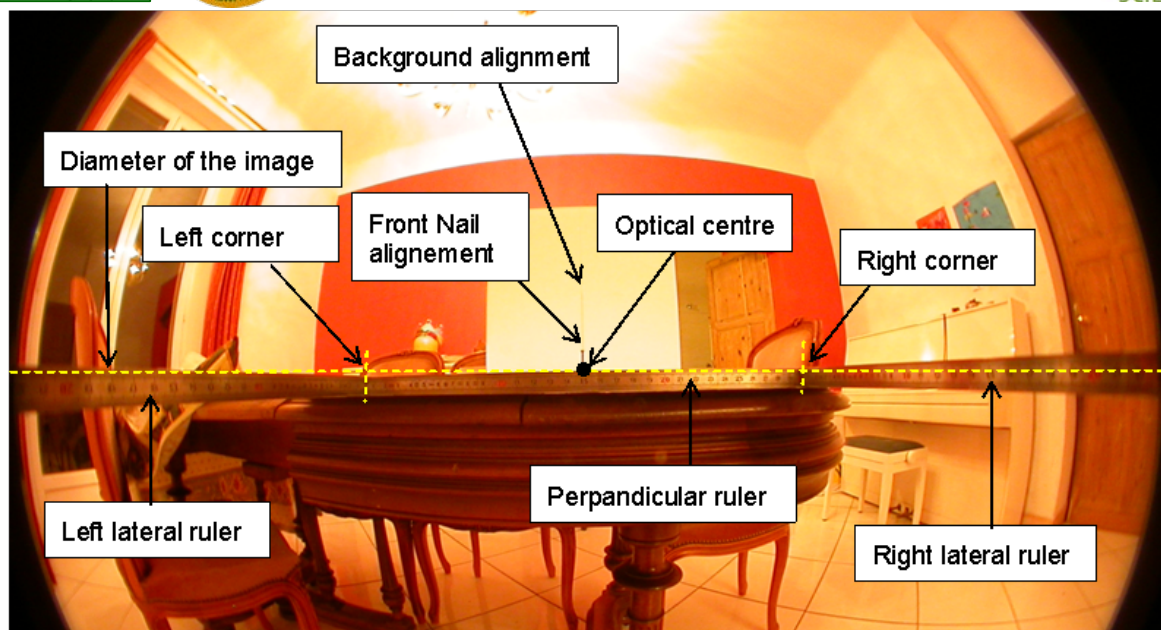
○ For the perpendicular ruler:  $\theta_x = \arctan(x / H)$

○ For the lateral rulers :  $\theta_y = \arctan(W / (H - y))$

Therefore, for the different reading in cm on the left lateral rulers (cells A23 to AXX), report the column number of the pixel in the image (cells B23 to BXX). Perform the same for the perpendicular and lateral rulers.

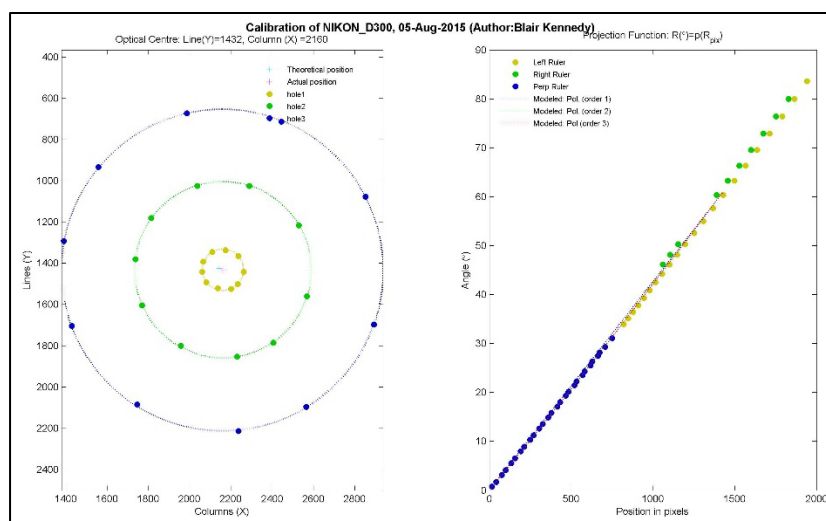


**Figure 35. Experimental design scheme.**



**Figure 36.** Example of an image of the experimental design taken with the hemispherical camera and used for the calibration of the projection function. The horizontal dotted yellow line corresponds to the diameter of the image passing through the optical center (defined by its coordinates as measured previously). **It must be aligned as much as possible with the 3 rulers since the camera should be aligned thanks to the front nail and background line.**

This process allows computing the coefficient  $a$  that relates the radius (in pixel) in the image to a corresponding viewing direction (in degrees). Then, the maximum field of view of the camera+fish-eye lens system can be computed by fitting a circle to the image (Figure 37). Once the “Projection Function” sheet is filled, the user must run CAN\_EYE, go in the Calibration Menu and then select “Projection Function”. CAN\_EYE asks then to select the excel file, computes automatically the projection function parameters and fill the “Results” sheet.



**Figure 37.** Left: example of a CAN\_EYE optical center calibration output, showing the fitting of the circles to the holes positions in the case of three holes. The actual optical center is shown by the red cross. Right: determination of the projection function. The different colored points correspond to the 3 rulers and the 3 lines to the 3 derived models (polynomial of order 1,2,or 3).

## 7. REFERENCES

- Andrieu, B. and Baret, F., 1993. Indirect methods of estimating crop structure from optical measurements. In: R.B. C. Varlet-Grancher, H. Sinoquet (Editor), In Crop structure and light microclimate - Characterization and Applications-. INRA, Paris, France, pp. 285-322.
- Andrieu, B. and Sinoquet, H., 1993. Evaluation of structure description requirements for predicting gap fraction of vegetation canopies. *Agric. For. Meteorol.*, 65: 207-227.
- Bonhomme, R., Varlet-Grancher, C. and Chartier, P., 1974. The use of hemispherical photographs for determining the leaf area index of young crops. *Photosynthetica*, 8(3): 299-301.
- Campbell, G.S., 1986. Extinction coefficients for radiation in plant canopies calculated using an ellipsoidal inclination angle distribution. *Agric. For. Meteorol.*, 36: 317-321.
- Campbell, G.S., 1990. Derivation of an angle density function for canopies with ellipsoidal leaf angle distributions. *Agric. For. Meteorol.*, 49: 173-176.
- Chen, J.M. and Black, T.A., 1992. Defining leaf area index for non-flat leaves. *Plant Cell Environ.*, 15: 421-429.
- Frazer, G.W., Trofymov, J.A. and Lertzman, K.P., 1997. A method for estimating canopy openness, effective leaf area index, and photosynthetically active photon flux density using hemispherical photography and computerized image analysis technique. *BC-X-373, Can. For. Serv. Pac. For. Cent. Inf.*
- Hughes, C., Denny, P., Jones, E. and Glavin, M., 2010. Accuracy of fish-eye lens models. *Applied Optics*, 49(17): 3338-3347.
- Knyazikhin, Y., Martonchik, J.V., Myneni, R.B., Diner, D.J. and Running, S.W., 1998. Synergistic algorithm for estimating vegetation canopy leaf area index and fraction of absorbed photosynthetically active radiation from MODIS and MISR data. *J. Geophys. Res.*, 103(D24): 32257-32275.
- Lang, A.R., 1991. Application of some of Cauchy's theorems to estimation of surface areas of leaves, needles, and branches of plants, and light transmittance. *Agric. For. Meteorol.*, 55: 191-212.
- Meyer, G.E. and Neto, J.C., 2008. Verification of color vegetation indices for automated crop imaging applications. *Computers and Electronics in Agriculture*, 63(2): 282-293.
- Miller, J.B., 1967. A formula for average foliage density. *Aust. J. Bot.*, 15: 141-144.
- Nilson, T., 1971. A theoretical analysis of the frequency of gaps in plant stands. *Agric. Meteorol.*, 8: 25-38.
- Ross, J., 1981. The radiation regime and architecture of plant stands, The Hague, 391 pp.
- Wang, Y.P. and Jarvis, P.G., 1988. Mean leaf angles for the ellipsoidal inclination angle distribution. *Agric. For. Meteorol.*, 43: 319-321.
- Warren-Wilson, J., 1959. Analysis of the spatial distribution of foliage by two-dimensional point quadrats. *New Phytol.*, 58: 92-101.
- Warren-Wilson, J., 1960. Inclined point quadrats. *New Phytol.*, 59: 1-8.
- Warren-Wilson, J., 1963. Estimation of foliage denseness and foliage angle by inclined point quadrats. *Aust. J. Bot.*, 11: 95-105.
- Watson, D.J., 1947. Comparative physiological studies in growth of field crops. I: Variation in net assimilation rate and leaf area between species and varieties, and within and between years. *Ann. Bot.*, 11: 41-76.



- Weiss, M., Baret, F., Myneni, R.B., Pragnère, A. and Knyazikhin, Y., 2000. Investigation of a model inversion technique to estimate canopy biophysical variables from spectral and directional reflectance data. *Agronomie*, 20: 3-22.
- Weiss, M., Baret, F., Smith, G.J. and Jonckheere, I., 2003. Methods for in situ leaf area index measurement, part II: from gap fraction to leaf area index: retrieval methods and sampling strategies. *Agric. For. Meteorol.*, submitted (August 2002).
- Weiss, M., Baret, F., Smith, G.J. and Jonckheere, I., 2004. Methods for in situ leaf area index measurement, part II: from gap fraction to leaf area index: retrieval methods and sampling strategies. *Agric. For. Meteorol.*, 121: 17-53.
- Welles, J.M. and Norman, J.M., 1991. Instrument for indirect measurement of canopy architecture. *Agronomy J.*, 83(5): 818-825.