Phenocopter Package Vignette

Amy Chan <Amy.Chan@csiro.au>

April 4, 2012

Contents

1 Installing

1. Linux/Mac: install the `exiftool` binary (if you don’t have it already).
   On Ubuntu, `apt-get install libimage-exiftool-perl`.
   Alternatively, `yum install perl-Image-Exiftool` on Fedora.
   Ensure it is in your `PATH`.
   (Windows - don’t worry, it comes with the `imageUtilities` package already).

2. Download the `utilitiesR` package from
   `https://bitbucket.org/mathematicalcoffee/utilitiesr/downloads`.

3. Download the `imageUtilities` package from
   `https://bitbucket.org/mathematicalcoffee/imageutilitiesr/downloads`.

4. Download the `lensDistortion` package from
   `https://bitbucket.org/mathematicalcoffee/lensDistortion-package/downloads`.

5. Download the `phenocopter` package from
   `https://bitbucket.org/mathematicalcoffee/phenocopter-package/downloads`.

6. Start R and install the `grid` and `jpeg` packages:

```
install.packages(c('grid','jpeg'))
# If you want to read TIFFs:
# install.packages(c('pixmap','gridExtra'))
# If you want to read PNGs:
# install.packages('png')
```

7. Install the downloaded packages from their zip files, either using 'Packages > Install from local zip file(s)...' or by script:

```
# put appropriate file names here.
install.packages(c('utilitiesR_1.x.x.zip',
                   'imageUtilities_1.x.x.zip',
                   'lensDistortion_1.x.x.zip',
                   'phenocopter_1.x.x.zip',
                 ))
```

   Note that the order of installation is utilitiesR, then imageUtilities, then lensDistortion, then phenocopter.

8. (Optional) Install ImageMagick. This can greatly speed up the lens distortion step in the algorithm. Get it from `http://www.imagemagick.org/script/index.php` (Windows: pick 'Binary Releases' and 'Windows' from the main page).

2 Introduction

The `phenocopter` package is the main one - the others are supporting packages for it.

The `phenocopter` package currently covers the process of extracting individual plots from raw images. The user selects a region of interest from the raw image and identifies the corner plots of that region. The code then corrects the image for lens and perspective distortion, and identifies each plot in the region of interest based on the information given.

The output of the process is an Rdata file (`.rda`) per image of all this information so that individual plot regions can be extracted later on demand.

This "pre-processing" is split into two halves:

1. **identification** - select region of interest and identify corner plots
2. **preprocessing** - perform lens and perspective distortion, plot identification, and storage of data. No user input required.

In future, the package will also include functions that use the previously-produced `.rda` files and perform analysis on individual plots (ground cover, etc).

3 Basic Usage

The most basic usage of the phenocopter package is to load the package, perform the interactive part (`SETUP_PHENOCOPTER_EXPERIMENT`), and then perform the non-interactive part (`RUN_PHENOCOPTER_EXPERIMENT`). This is **not** the recommended usage though!

```
# 1. load library.
library(phenocopter)

# 2. interactive part:
obj <- SETUP_PHENOCOPTER_EXPERIMENT()

# 3. non-interactive part:
results <- RUN_PHENOCOPTER_EXPERIMENT( obj$details, obj$options )
```

The interactive part (`SETUP_PHENOCOPTER_EXPERIMENT`) will guide you through choosing a place to save the output, and choosing the image files you wish to process. It will then guide you through selecting the region of interest for each image, providing the size of a standard plot, and identifying the corner plots. It produces an R object `obj` that is then fed in to `RUN_PHENOCOPTER_EXPERIMENT`.

`RUN_PHENOCOPTER_EXPERIMENT` is the “preprocessing" part of the algorithm and does not require use interaction. It performs the lens/perspective correction on the images and identifies plots and splits them up.

However, the **recommended** way to run the phenocopter code is to write a script for it where you explicitly set the files to be processed and options to process with. This is much more reproducible and customisable than the above method.

4 Template Script

The recommended (but not the simplest) way to use the `phenocopter` package is to write a script for it.

A typical script is displayed below. It looks long but it isn't, don't be daunted! It really isn't all that different to the basic usage above.

```
###############
# Step 1.
# Load library
###############
library(phenocopter)

###############
# Step 2.
# Create a list of files you want to process.
# A bit of shorthand defining the paths because
#  I'm too lazy to type them in the full.
###############
baseDir  <- 'E:/Phenocopter/Dalbeg_Season/28-11-2011/flight_1'
imageDir <- file.path(baseDir,'vis')
thermalDir <- file.path(baseDir,'thermal')
# images I want to process. Two visuals and a thermal.
fileList <- c(
              file.path(imageDir,'r0045918.jpg'),
              file.path(imageDir,'r0045919.jpg'),
              file.path(thermalDir,'frame20111028_132024.bin')
              )
```

```
###############
# Step 3.
# customise options. see ?defaultOptionsPhenocopter.
###############
opts <- defaultOptionsPhenocopter()

# Set the root directory you in which want results to appear
# Will be created if it doesn't exist.
opts$BASE_SAVE_DIR        <- "./results"

# if you want all sorts of debugging output images, uncomment the following.
# Not really necessary though.
#opts$SAVE_BARREL_CORRECTED <- T     # barrel-corrected image,
#opts$SAVE_RECTANGLIFIED    <- T     # rectanglified and barrel-corrected image,
#opts$SAVE_OVERLAY          <- T     # the grid superimposed on the rectanglified image
#opts$SAVE_PLOTS            <- T     # the individual plot pictures.
#opts$SAVE_TRIMMED_PLOTS    <- T     # the trimmed-sides individual plot pictures.
# save the output here.
# see ?defaultOptionsPhenocopter for more things to tweak.

# ** Load parameters to correct thermal distortion. Leave this be!
data(cameraCalibrationParameters) # -> openCVParameters
camout <- subset(openCVParameters, model=='Miricle 307 KS' & zoom=='out')
camin <- subset(openCVParameters, model=='Miricle 307 KS' & zoom=='in')
opts[c('OPENCV_K1','OPENCV_K2','OPENCV_K3','OPENCV_P1','OPENCV_P2')] <-
    camout[,c('k1','k2','k3','p1','p2')]
opts[c('OPENCV_FX','OPENCV_FY','OPENCV_CX','OPENCV_CY')] <-
    camout[,c('fx','fy','cx','cy')]
opts[c('OPENCV_FXSCALE','OPENCV_FYSCALE',
       'OPENCV_CXSCALE','OPENCV_CYSCALE')] <-
           camin[,c('fx','fy','cx','cy')]

###############
# Step 4.
# Do the user-interactive part of the experiment.
###############
obj <- SETUP_PHENOCOPTER_EXPERIMENT( options = opts, fileList=fileList )
# or you could load data from a previous run, housed in opts$BASE_SAVE_DIR
# obj <- loadOptions( fileList, opts$BASE_SAVE_DIR )

###############
# Step 5.
# Do the non-user-interactive part of the experiment.
###############
results <- RUN_PHENOCOPTER_EXPERIMENT( obj$details, obj$options )
```

Take this script and modify it as necessary; then run it in R by using `source('my_script_name')`.

The script can be found in the `phenocopter` directory of your library; it's called `template_script.r`. Alternatively, start R and perform:

```r
library(phenocopter)
# loc has the location of  emplate_script.r.
loc <- system.file('template_script.r',package='phenocopter')
# You can copy it & paste somewhere else to modify, or just do
#  (replace 'to' with where you want to copy it to):
file.copy(from=loc, to='~/template_script.r')
```

What happens in this script? Well, just like before, the library is first loaded into R.

Then (Step 2), a character vector `fileList` is initialised. This is a vector of files we want to process.

In Step 3, we specify some options to configure the inputs, outputs, etc. of the code. This is done via the function `defaultOptionsPhenocopter` (see `?defaultOptionsPhenocopter` for more details).

```r
opts <- defaultOptionsPhenocopter()
# set options like so:
opts$OPTION_NAME <- OPTION_VALUE
# for example, to set the output directory:
opts$BASE_SAVE_DIR <- './results'
```

There are a number of other options the user may wish to tweak. These are documented in `?defaultOptionsPhenocopter` (or the pdf help for that function).

The default is not to perform any lens distortion correction on any of the images. The code starting from `data(cameraCalibrationParameters)` to just before Step 4 populates the `opts` structure with the necessary parameters to do the correction for the current thermal camera (Miricle 307 KS).

Note that if the camera is changed, the distortion parameters in `opts` will need to be changed also. At the moment (March 2012), no lens distortion correction on the visual or NIR cameras is performed (the parameters are unknown).

Finally, we have the same two lines as before for `SETUP_PHENOCOPTER_EXPERIMENT` to perform the interactive part and `RUN_PHENOCOPTER_EXPERIMENT` for the non-interactive part.

Note that the interactive part of the code, `SETUP_PHENOCOPTER_EXPERIMENT`, stores the collected user input into the R list `obj`. The non-interactive part, `RUN_PHENOCOPTER_EXPERIMENT`, uses these details.

If one had already done the interactive part of the algorithm and wanted to re-run the non-interactive part, they could do so without having to do the interactive part all over again - use the `loadOptions` function. This reconstructs the `obj` object from the saved `.rda` files from previous runs.

`RUN_PHENOCOPTER_EXPERIMENT` corrects lens distortion (see `?lensDistortion` for an overview) and perspective distortion (see `?projectiveTransform`). It then computes how to split the region of interest into individual plots. By default, a “check" image will be generated that shows the user how the plots were divided and labelled (for verification purposes). The necessary information to reproduce the individual plots is stored in a `.rda` file.

5 Getting the Code to Run Smoothly

The phenocopter package expects various files to be in particular directories in order to run smoothly.

In particular, the image files, field plan (csv), and log file (ricoh.log) should be organised as follows:

```
base_dir/ ........................................ e.g. \\pi-tesla-sl\tesla-e\Phenocopter
└─ Site_Season/ ................................................... e.g. Dalbeg_Season
   ├─ field_plan.csv
   └─ date/ ........................................................... yyyy-mm-dd
      └─ flight_flightnum/ ................................................ e.g. flight_1
         ├─ ricoh.log
         ├─ vis/
         │  └─ lots of images (r00xxxx.jpg)
         ├─ NIR/
         │  └─ lots of images (r00yyyy.jpg)
         └─ thermal/
            └─ lots of images (frameyyyymmdd_hhmmss.bin)
```

A current (April 2012) example of this format can be found in
`\\pi-tesla-sl.nexus.csiro.au\tesla-e\Phenocopter\Dalbeg_Season\28-11-2011\flight_1`.
Note that if the csv/log files are not found at the expected places, you may be confronted with a dialogue box asking you to choose them.

6 FAQ

6.1 I keep getting confronted with "choose file" dialogues and I'm not sure what they're for

This is usually because the csv or log files have not been located at the expected places (see the "Getting the Code to Run Smoothly" section).
In particular, the csv field plan should be in `base_dir/Site_Season/`, and the ricoh.log is looked for in `base_dir/Site_Season/flight/`.

6.2 The "performing barrel correction..." step takes *forever*!

Install ImageMagick and make sure it's in your `PATH`.
On Linux, this means that if you type:

```
which convert
```

The path that it returns should be in your `$PATH`.
On Windows, using the ImageMagick installer (http://www.imagemagick.org/script/binary-releases.php#windows) should automatically set everything up for you. If you type:

```
convert
```

into a command prompt, you should automatically have its help file displayed.

6.3 It keeps complaining about not being able to find `exiftool`

Make sure exiftool is in your `PATH` environment variable.

On Linux, this means that if you type:

```
    which exiftool
```

The path that it returns should be in your $PATH.

On Windows, this means that exiftool.exe should be in a folder that is in the PATH environment variable. If you're not comfortable editting your PATH, then just place exiftool.exe into C:\Windows; this is probably in your PATH already.

NOTE: The Windows version of exiftool.exe is called exiftool(-k).exe by default; just rename it to exiftool.exe.

To test whether it's working, type

```
    exiftool --help
```

into a command prompt and you should see a help file appear.

6.4 I can't seem to use my field plan with SETUP_PHENOCOPTER_EXPERIMENT

Your field plan **must** be saved as a CSV file in order to work with SETUP_PHENOCOPTER_EXPERIMENT. XLS and XLSX files will not work (although we plan to support them in the future).

6.5 Other questions

Email me: <Amy.Chan@csiro.au>. The package is still in development phase so the kinks are not yet ironed out.

7 See Also

- The documentation for the phenocopter package.
- ExifTool: http://www.sno.phy.queensu.ca/~phil/exiftool/
- Packages: https://bitbucket.org/mathematicalcoffee in the 'Repositories' list - click on the desired package and then go to the 'Downloads' tab to obtain the zip file.
- ImageMagick: http://www.imagemagick.org/script/index.php