The Aberystwyth Leaf Evaluation Dataset: A plant growth visible light image dataset of Arabidopsis thaliana

Jonathan Bell, Hannah Dee

November 22, 2016

1 Introduction

We are releasing the dataset acquired to support the work of the EPSRC funded project "Dynamic Modelling of Plant Growth with Computer Vision" (grant code EP/LO17253/1) in the hope that its availability will help further advance the state of the art in the use of image analysis for plant sciences. It is a visible light, top down, timelapse image dataset of *Arabidopsis thaliana* (Arabidopsis). It incorporates the original images, some with leaf-level ground truth annotations, harvested plant ground truth data and scanned images together with supporting software.

1.1 Structure

There are three principal sections to this document, as follows:-

- Images and annotations describes the images themselves and the image ground truth annotations.
- Harvested plant data describes the harvested plant data and scanned images.
- Software describes the segmentation evaluation software and distortion correction software supplied with this dataset.

The dataset is arranged in subdirectories that echo this structure. Names of files and directories, as well as commands to be entered at a terminal, are highlighted by use of a `fixed width font`.

2 Images and annotations

The dataset consists of top down, timelapse, visible spectrum images of Arabidopsis thaliana obtained using the Photon Systems Instruments (PSI) PlantScreen phenotyping platform located at the National Plant Phenomics Centre at Plas Gogerddan, Aberystwyth, UK.

2.1 Plant material

Arabidopsis thaliana: accession Columbia (Col-0) plants were grown in 180ml compost (Levington F2 + 205 grit sand) in PSI 6cm square pots. The plants were grown with gravimetric watering to 75% field capacity under a 10 hour day 20°C/15°C glasshouse regime on the PSI platform. Plants were sown 23/11/2015 and pricked out into weighed pots at 10 days old. 80 plants were split between four trays. Two plants were removed from each tray at 30, 37, 43, 49 and 53 days after sowing and final harvest was 56 days after sowing. The plants were removed so as to lessen the possibility of neighbouring plants overlapping each other, and to obtain information about the plant through destructive harvesting. However, there are a few instances where overlap between plants has happened nonetheless.

2.2 Image acquisition

Imaging started on 14/12/2016, 21 days after sowing, and continued to final harvest so there was an imaging period of 35 days. The plants were top view imaged using the visible spectrum every 15 minutes (nominally, in practice the interval was approximately 13 minutes) during the daylight period. The first image was typically taken soon after 9am and the last after 8pm, so over a longer period than the nominal 10 hour day. There are gaps in the image sequence attributable to machine malfunctions. These are listed in Table 1.

Table 1: Gaps in the image capture sequence.

Date	Gap start	Gap end
14/12/2015	12.54	14.15
14/12/2015	16.24	end of day
15/12/2015	9.03	13.33
23/12/2015	16.23	17.28
23/12/2015	19.57	21.04
1/1/2016	10.01	end of day
11/1/2016	12.29	16.42
15/1/2016	start of day	10.04

The times in the table are for the first tray photographed. Imaging continued later than usual on 23/12/2015. The imaging resulted in the acquisition of 1676 images of each tray, each image having 20, 18, 16, 14, 12 or 10 plants as plants were harvested.

Images were taken using the PSI platform's built in camera, an IDS uEye 5480SE, resolution 2560*1920 (5Mpx) fitted with a Tamron 8mm f1.4 lens. This exhibits some barrel distortion. Images were saved as .png files (so compression is lossless) with filenames that incorporate tray number (031, 032, 033 and 034) and date and time of capture. Date and time are in the format YYYY-MM-DD–hh-mm-ss so lexical ordering matches chronological order. Days after sowing can be found by adding 7 to the day for those images taken in December 2015 and adding 38 to the day for those taken in January 2016. Times are slightly different between trays as images were taken sequentially. Other than png compression, no post-processing was done. This means images exhibit the camera's barrel distortion. Code to correct this is supplied along with the dataset. See the section on Supporting software below.

The images are in the `images_and_annotations` directory, arranged like this, using tray 31 as an example:-

- `PSI_Tray031`
  - `tv` added by PSI machine to separate from side views, not taken in this experiment.
    - `gt` contains ground truth annotations (Tray 31 and 32 only).
    - `masks` contains the PSI auto-generated plant level masks
    - the original images.

The masks in the `masks` subdirectories are corrected for distortion using PSI proprietary code. Therefore, they do not match the uncorrected images and cannot be guaranteed to match images corrected using the supplied software. We provide these masks for completeness but as they provide an algorithmic segmentation based upon proprietary code they may be of limited use to vision researchers.

The annotation files in the `gt` subdirectory represent hand-labelled segmentations, and these are described in the following subsection.

2.3 Annotations

In addition to the original images we provide hand-annotated "ground truth" images of a selection of the images. All images have the same filename as the corresponding original image with the suffix `_gt` preceding the filename extension (`.png`). They are located in a subdirectory, also called `gt`, of the directory containing the corresponding original images. The suffix prevents the inadvertent over-writing of the original should the two images be moved to the same directory. The annotations were done from the original images so need correcting for barrel distortion if they are to match corrected images.

We have annotations of the last image taken each day during the growth period from one of the four trays (Labelled Tray 031). In addition there are annotations of the first image taken 21, 28, 35 and 42 days after sowing and the first image taken after 2pm 28, 35, 42 and 49 days after sowing. This amounts to 43 images with between 10 and 20 plants in each image and a total of 706 annotated plant images from tray 031, as shown in Table 2.

Table 2: Annotated plant images from tray 31

Plants per image	Images	Total plants
20	12	240
18	9	162
16	8	128
14	7	98
12	4	48
10	3	30
Total	43	706

There are also annotations of the first image taken of Tray 032 and the first image taken after 2pm every third day, starting from the 22nd day after sowing. This adds another 210 plant images, as in Table 3

Table 3: Annotated plant images from tray 32

Plants per image	Days after sowing	Images	Total plants
20	21, 22, 25, 28	4	80
18	31, 34	2	36
16	37, 40	2	32
14	43, 46	2	28
12	49, 52	2	24
10	55	1	10
Total		13	210

2.3.1 Colouring of annotations

All annotations use a sequence of colours to show the order in which leaves appear. Colours were specified using RGB values such that each new leaf used a higher value than its predecessor. Each of red, green and blue are incremented though the range before the next is started. The background is black. The following sequence was used:-

- R 30 to 250, G 0, B 0 for first 12 leaves
- R 250, G 30 to 250, B 0 for the next 12
- R 250, G 250, B 30 to 150 for the next 7
- R 250, G 0, B 30 to 250 for the next 12
- R 250, G 30 to 150, B 250 for the next 7
- R 250, G 100, B 30 to 230 for the next 11
- R 250, G 30 to 230, B 100 for the next 11

This gives 72 different colours, so up to 72 leaves can be annotated. Colours are not reused. Each increment step is 20. Where the annotator was uncertain of the order, the incremented value was reduced by 5. This is the case with the first leaves as by the time image capture was started there were several leaves showing. So where a plant has six leaves visible in the first image, the first seven red values will be 25, 45, 65, 85, 105, 125, 150 as the annotator can identify the 7th leaf. This means that the ordering of leaves is uncertain, at least in some cases, but each leaf does keep its colour through the timelapse sequence so the annotation supports the tracking of leaves. The same scheme is used for all plants in an image. Examples of annotations from early and late in the growth period are shown in Figure 1.

Figure 1: Two views of the same plant taken 27 and 42 days after sowing together with ground truth annotations.

This approach has the unfortunate side effect of making colours less readily distinguishable visually - the early annotations especially are not very eye catching! All leaves have a higher red value than other components as this made the annotation itself easier. This is why a full range of colours was not used.

The supplied evaluation software scans over both images and lists the colours used so there need be no correspondence between the segmentation and annotation colours - they need not be represented as successive integer values.

2.3.2 Training and test data

The suggestion is that the images of tray 031 are used as training data and the images of tray 032 as test data. This means no plant will appear in both sets. However, it has been suggested that the daily change in leaf orientation ("hyponasty") might result in inconsistency between these annotated subsets of our data, being from different times of day. A visual inspection suggests that this is not the case, the plants being acclimitised to the greenhouse's artificial day length. However, if this is felt to be a problem, an alternative approach would be to halve each image and use all the top halves as training data and bottom halves as test data or divide so there are left hand images with three columns of four pots and right hand images with two. Dividing the images does, of course, reduce the amount of available training data, but increases the amount of available test data. Note that if the barrel distortion exhibited by the images is to be corrected, this needs be done before any cropping of the images.

3 Harvested plant data

Besides the image annotations, there is "plant based" ground truth. This was obtained from the plants harvested periodically and from all remaining plants at the end of image acquisition. The rosette area and fresh and dry weights were obtained and results included in the Excel spreadsheet `AT20 graphed GT.xlsx` in the `harvested_plant_data` directory. Rosette area was obtained using a Windias Delta T leaf area scanner. In addition the rosette and separated leaves were scanned using a flat bed scanner at 300dpi and the images saved as .png. Leaves were arranged in development order. These images are provided in subdirectories named for the date of harvest. The image filenames include the tray number and location of the plant in the tray (positions a1 to d5). Data from the harvest undertaken on 30/12/2015 is inconsistent and should be ignored. In addition, the arrangement of the separated leaves scanned from this harvest should not be relied upon as being in development order.

There is also a preliminary report, `Preliminary analysis AT20.docx`.

4 Software

Alongside the images, we provide software to support evaluation of segmentation and to correct the optical distortion apparent in the PSI images.

4.1 Evaluation of segmentation

This is supplied as a Java jar file as a plugin for ImageJ and as Java source code. We propose an alternative evaluation scheme to that used for the Leaf Segmentation Challenge (LSC) [1, 2]. The main difference is that our proposed scheme only allows a region (leaf) in either image to be mapped to at most one region in the other image while the LSC scheme does not enforce this. There is a discussion of our proposed approach to segmentation evaluation in [3].

The results include the number of regions (leaves) in both the test and ground truth images. The supplied code supports our preferred approach, returning both the Jaccard and Dice measurements for both plant and leaf level segmentation. It does also provide an implementation of the LSC evaluation metric for comparison. It also supports the use of any values for plant pixels (segmentation and ground truth - they need not be consecutive integers) but black is specified as the background. While the scheme for colouring supports tracing the ordering of appearance of leaves and tracking of leaves, the evaluation software does not support this. It simply chooses that combination of mapping across annotated and segmented leaves that gives the best score.

The evaluation does not support the inclusion of multiple plants in one image as colours might vary from plant to plant. There is code to support the cropping of a multiple plant image into a set of single plant images and subsequently running the evaluation on these. This has the limitations that plants need to be laid out in an even grid (it simply divides the image into equal size regions) and any empty pots will be included. The first of these is acceptable for most of the dataset images but not once plants get close to overlapping. It should be possible to spot those results that are from image crops of empty pots. We are unable to detect connected objects as some of our ground truth annotations are not connected, as the petioles (leaf stems) were partly buried.

4.1.1 Using the evaluation software

The evaluation software is at `software/evaluation`. It is supplied as a Fiji (Fiji Is Just ImageJ) plugin. To run from the supplied jar file, all that should be required is to copy the jar file to the `Fiji.app/plugins` subdirectory of your fiji directory and restart Fiji.

Once installed "evaluate segmentation" should appear in your Fiji plugins menu. Once selected you will be asked for the location of the segmented image you want to evaluate and then the ground truth image you want to evaluate it against. You will then be asked for a location to save the results. Finally you will be asked how many items there are arranged across and down your images (5 and 4 in case of our images). This allows you to get results from images with plants arranged in a grid, such as those supplied. It does rely on the objects being arranged in an evenly spaced grid. Results are saved in the specified location with the filename `evaluation.csv` and if more than one object was specified, the summary results as `summary.csv`. The new results will be appended to any existing files with these names.

Results are saved to a tab separated value file for opening in a spreadsheet. If there is one object in the images, this lists the following:-

- test image name
- ground truth image name
- test image region (leaf) count
- truth image region (leaf) count
- segmentation difference - the amount by which the number of subsidiary regions in the two images differ.
- object Jaccard index - for plant from background segmentation.
- sub region Jaccard - for leaf level segmentation. The best match between individual leaves, not allowing a region in either image to map to more than one region in the other. This is our preferred metric.
- object Dice - for plant from background segmentation.
- sub region Dice - for leaf level segmentation. The best match between individual leaves, not allowing a region in either image to map to more than one region in the other.
- symmetric best Dice - this does allow a region in one image to map to more than one region in the other. It gives the same result as the Leaf Segmentation Challenge evaluation code.

If there is more than one object in an image, the results file gives the results for each object and mean results for the image as a whole. Results are similar but beside the segmentation (leaf count) and segmentation difference results, whether each object is under- over- or correctly segmented is shown. The summary row gives the mean amount of under and over segmentations from the objects (plants) that are under and over segmented. There is also a summary file that just gives the mean results for the image as a whole but with no leaf count. The results are added to any existing results file on subsequent runs.

4.1.2 Source code

The source code and supporting resource files for building the plugin using Maven in Eclipse can be found in `software/evaluation/source`.

4.2 Optical distortion

The files in `software/camera_correction` can be used to build a small command line application to correct the barrel distortion exhibited by the camera used to acquire these images.

The PSI camera was calibrated using the OpenCV calibration tutorial code, and the chequerboard photographs for this are included alongside the code. This generates an xml file with values that can be provided to the Undistort application, also supplied. This is supplied as C++ source code together with CMakleLists.txt to enable the application to be built using cmake.

The Undistort executable is run from the command line

```
# Undistort <image filename> psi_camera_data.xml
```

and it produces a new image and saves it alongside the original using the original filename with `_corrected` appended. It can be run without the calibration xml file, in which case default values for the PSI camera are used.

4.2.1 Dependencies

The Undistort code depends on

- OpenCV - http://opencv.org/
- CMake - https://cmake.org/

4.2.2 Building the program

1. Copy the source code file, `Undistort.cpp`, `CMakeLists.txt`, and (optionally) the calibration xml file, `psi_camera_data.xml` to your chosen location.
2. If you do not have them, you will need to install OpenCV and CMake.
3. Once you have the dependencies, check that the `link_directories` entry in `CMakeLists.txt` points to where your OpenCV is located, changing as required.
4. In a terminal
   ```
   # cd
   ```
   to the location you copied the files to and run
   ```
   # cmake .
   ```
   and then
   ```
   # make
   ```
   and the application should build.

This has only been tested on Linux. The OpenCV website have tutorials on building OpenCV application in Windows using Microsoft Visual Studio.

The calibration file `psi_camera_data.xml` was generated using the "Camera Calibration with OpenCV" tutorial at http://docs.opencv.org. The photographs that were used are in the `calib` subdirectory of `software/camera_correction`.

References

[1] Massimo Minervini, Andreas Fischbach, Hanno Scharr, and Sotirios A. Tsaftaris. Finely-grained annotated datasets for image-based plant phenotyping. *Pattern Recognition Letters*, 2015.

[2] Hanno Scharr, Massimo Minervini, Andreas Fischbach, and Sotirios A. Tsaftaris. Annotated image datasets of rosette plants. Technical Report -, Institute of Bio- and Geosciences: Plant Sciences (IBG-2), Forschungszentrum Jülich GmbH, Jülich, Germany, 2014.

[3] Jonathan Bell and Hannah M. Dee. The subset-matched Jaccard index for evaluation of segmentation for plant images. Technical report, Department of Computer Science, Aberystwyth University, 2016. available at: https://arxiv.org/abs/1611.06880.