Usage Instructions for Micro-CT Plant Images

Nathan Hughes (nah31@aber.ac.uk)

August 21, 2018

1 Releases

This software works as a iteration-based release, adding new features and optimising and as such please refer to releases for the latest stable version.

2 Usage

Usage of this software is straightforward. Inputting a directory, a voxel size and a minimum size of expected grain objects will output and write grain statistics and image to file.

2.1 Setup variables

A brief setup of environment variables are required, this is an example:

```
voxelSize = 68.8; % or whatever micro-meter to voxel ratio was used in scanning
minimumGrainSize = 10000; % a minimum grain size of interest
structEleSize = 5; % a size of structuring element to use for morphological operations

% Every folder in CT-Scans folder and every ISQ file in them
directory = '/home/files/CT-Scans/*.ISQ';

% More optional parameters which change type of scanning
startFrom = 1;
endAt = 0;
watershed = true;
```

2.2 Optional Parameters

- `startFrom` indicates file to start at from the directory indicated, almost always use '1'. It's useful for if processing fails partway through
- `endAt` is where the process will stop, useful for if you'd only like to use a few scans for testing. Default is 0 and this will process everything
- `watershed` will be a boolean value which will toggle if the more robust and strict watersheding is triggered - this will make processing take a lot longer and at times may over-segment. Generally only use if scans are failing to separate properly, or are particularly small (i.e. wild types / barley)

2.2.1 N.B.

Scanco has been known to alter the filenames for a reason known only Minerva would know. So be on the lookout for:

```
C0000123.ISQ;1
```

To get around this I recommend running a script something like this (or a sed equivalent):

```
from glob import glob
from os import rename

[rename(f, f.replace(';','')) for f in glob('*.ISQ;1')]
```

2.3 Running

Running the program is as simple as calling the processDirectory function.

```
% Will process all files found by rdir function
processDirectory(directory, structEleSize, voxelSize, minimumGrainSize);
```

3 Files and Functions

3.1 cleanWheat

cleanWheat is a function which takes as input a filename location on disk of an ISQ raw image, it processes it and outputs a binary 3D image and a greyscale 3D image which has been cleaned and segmented.

3.2 countGrain

countGrain takes cleaned image data, separates each identified grain and computes statistics on a grain-per-grain basis. It returns two statistics objects, one with raw pixel data counted and another with computed metric values.

3.3 filterSmallObjects

filterSmallObjects attempts to remove all objects which are smaller than the specified parameter during setup. This uses pixel size **not** metric sizes for this.

3.4 imSurface

imSurface is a library originally by David Legland. It measures the surface area in pixels of a 3D object.

3.5 processDirectory

processDirectory is the main controlling function of this software, it moves image data around from function to function, gathers image results/measurements and saves it to disk from here.

3.6 subdir

subdir is a function which recursively finds files, it is used to find files in sub-directories by using the '*' wildcard in the directory name parameter.

- This function was redone as the previous 'rdir' has operating issues with certain versions of windows.

3.7 readISQ

readISQ originally developed by Johan Karlsson, we have modified it to make speed increases and added specific slice loading, this helps for increased speed when processing larger images

3.8 segmentRachis

segmentRachis finds locations of nodes along the rachis of spikes of wheat, oats etc. Use of this data is primarily for locating joining points of split scans.

3.9 watershed3D

watershed3D incorporates traditional watershedding techniques and has adapted them to work in 3D. It also makes use of modernised distance-based watershed methods, by way of chessboard distance technique.

3.10 writeTif

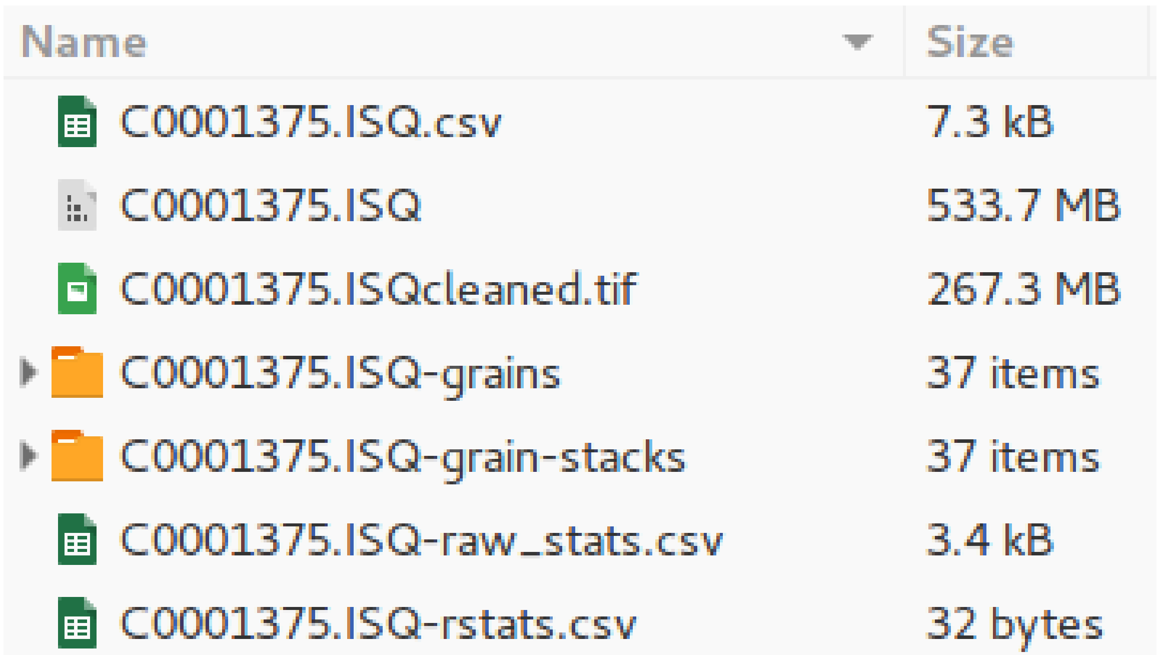
writeTif writes image stacks to disk as TIF formatted files.

4 Output

From successful running of this software output will be:

- A statistics of grains CSV with metric values
- A statistics of grains CSV with raw values
- A TIF file of the segmented image
- A statistics file of the rachis top and bottom points.
- A folder of 2D cross sectional images, for each grain
- A folder of 3D TIF files, each a individual grain

The output folder should look similar to this:



Name	Size
C0001375.ISQ.csv	7.3 kB
C0001375.ISQ	533.7 MB
C0001375.ISQcleaned.tif	267.3 MB
C0001375.ISQ-grains	37 items
C0001375.ISQ-grain-stacks	37 items
C0001375.ISQ-raw_stats.csv	3.4 kB
C0001375.ISQ-rstats.csv	32 bytes