How to use SfM-PCNN-LAD Extraction

To use the code, a few software is needed: Photoscan (1.4.5 and above), Matlab (2018), Python (3.5 and above) and an Android APP named AngleCam Lite.

(Although this seems a little complex, but it is actually not, trust me!)

# Step 1: Taking photos

Install the Android APP AngleCam Lite, ether get it from Google Play: <https://play.google.com/store/apps/details?id=com.derekr.AngleCam&hl=en_US> or download the APK from the following link: <https://github.com/jianboqi/SfM-PCNN-LADExtraction/blob/master/AngleCamLite/AngleCam%20Lite.apk>

The user manual of AngleCam Lite can refer to :

<http://anglecam.derekr.com/download/Introduction_EN_20170324.pdf>

Before using this APP for angle measurement. Please perform the following setting:

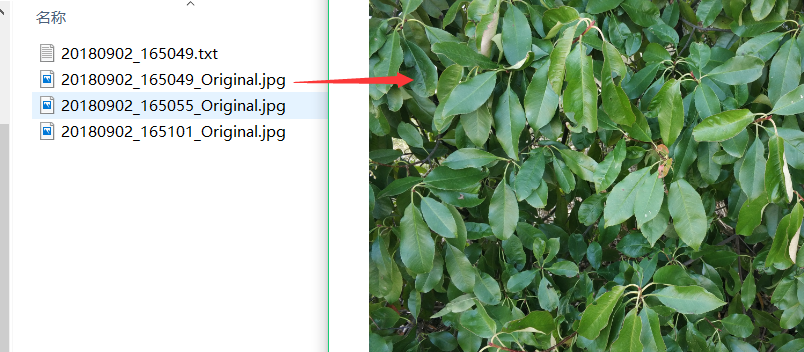
* Open “Setting”, “Orientation setting (方向设定)”, Change the orientation to “Landscape (横向)”
* “Storage setting”, click “Save original file (储存源文件)” and “Save text file (per photo) (储存文字说明档案（每张照片）)”. This option will ensure that each photo has one corresponding txt file that describes the meta data (Pitch, Roll, Yaw, etc.)

When taking the first photo, please ensure that the roll angle equals to 0°, approximately, which will greatly facilitate the calculation.

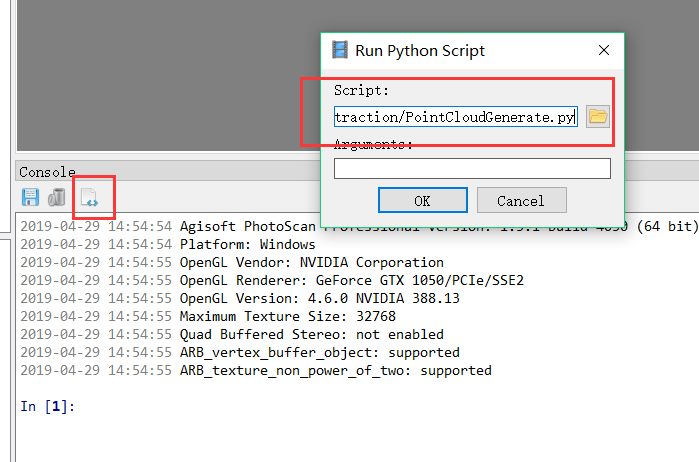


# Step 2: 3D point cloud generation

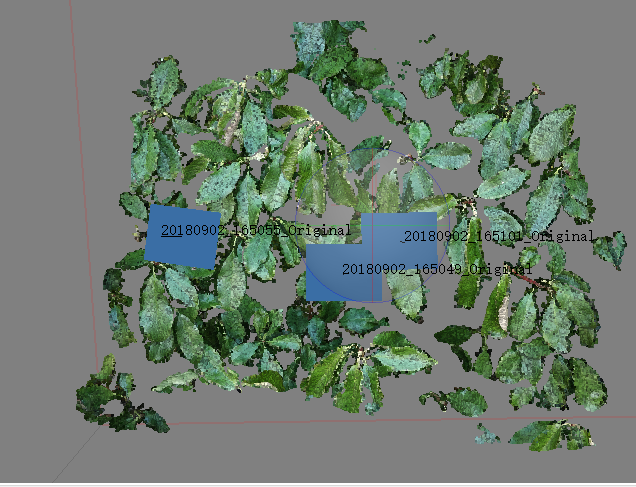
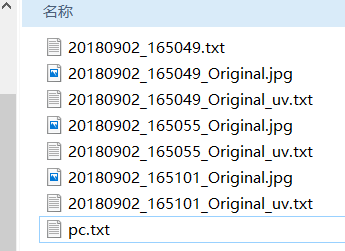
Usually, at least 2 photos (with significant overlapping) are needed for generating a point cloud, but we recommend 3 photos to get a more reliable result. Put the three photos into a folder, such as “Demo”, as illustrated in the following figure. The three photos are exported from AngleCam Lite. The text file “20180902\_165049.txt” is the meta data for the first photo.



Now, Open the Photoscan software, run the script from the Console, and choose the “PointCloudGenerate.py”. This will let you to choose the image files, i.e., the three photos.



After the execution of the script, the Photoscan will generate the point cloud. In this “Demo” folder, you will find some new files have been created. “pc.txt” stores the point cloud. Other “\*\_uv.txt” files store the project pixel coordinate of each 3D point in each image. This files are only intermediate results.

# Step 3: Leaf detection in Images

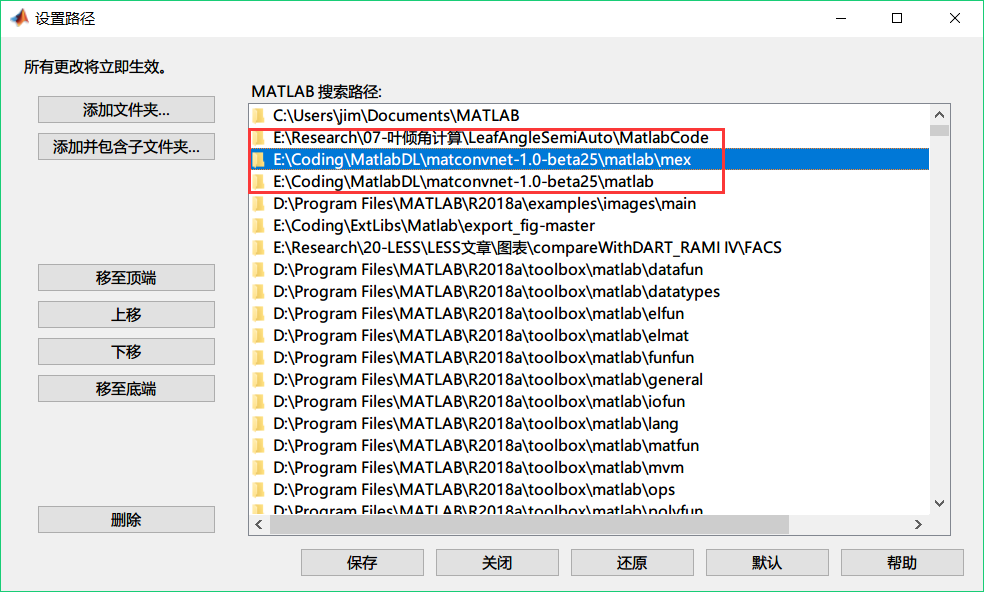
In this step, Matlab are used. Launch Matlab 2018. Before using the PCNN, “MatConvNet” is needed: <http://www.vlfeat.org/matconvnet/>. Please follow the instruction: download the source code of MatConvNet and compile it with Matlab.

If you meet problem such as “cl.exe is not found”, please refer to:

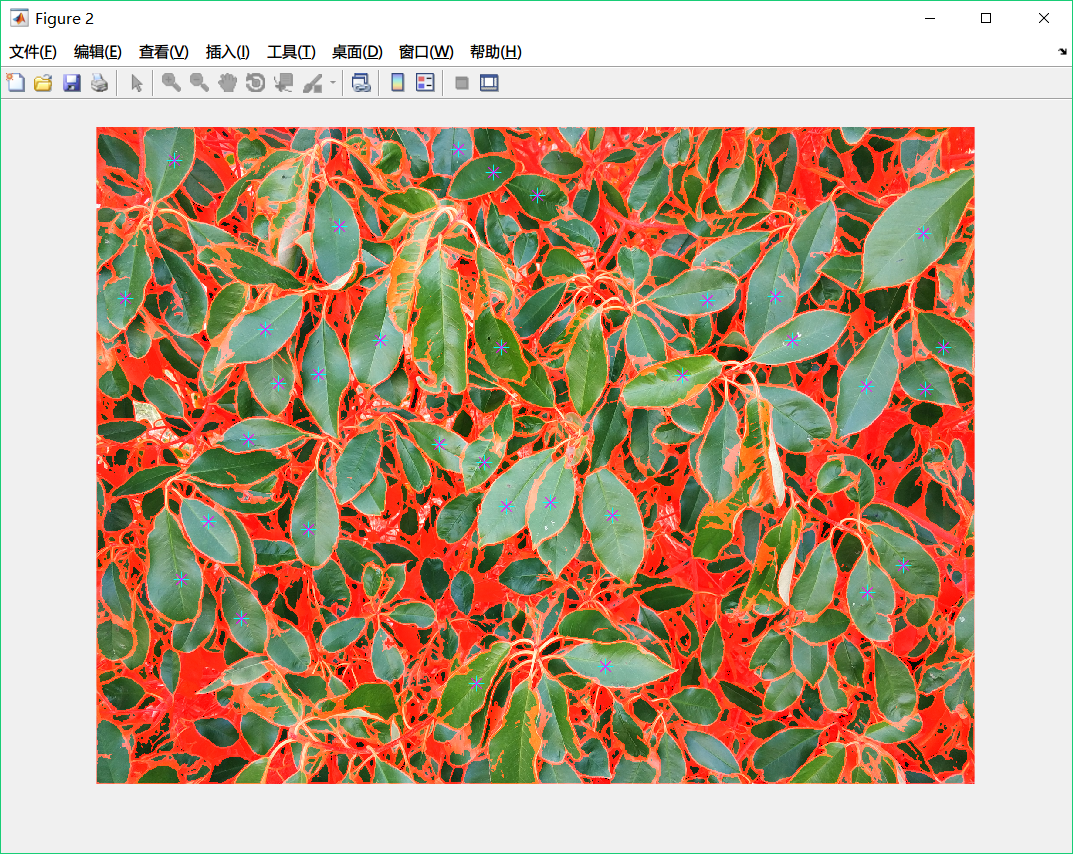
<https://stackoverflow.com/questions/40226354/matconvnet-error-cl-exe-not-found>

“*Install visual studio community edition ( it's the free edition ) (minwg compiler will not work) go to C:\Program Files (x86)\Microsoft Visual Studio search for cl.exe take the one appropriate for your computer architecture and copy it to the folder containing the matconvnet installation run mex -setup c++ and set it to visual studio and you're good to go”*

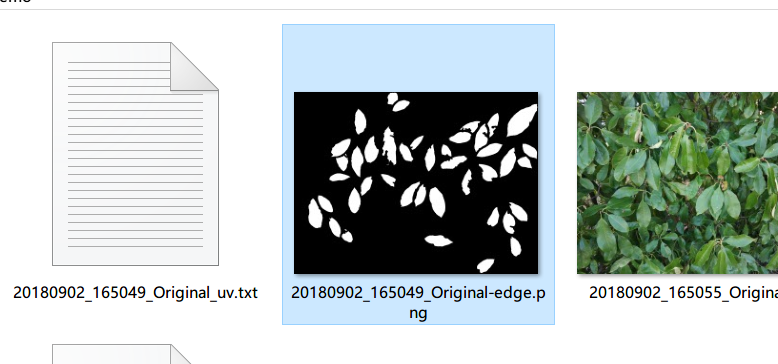
After the compilation, Add folder “matlab” and “matlab/mex” to the matlab path. At the same time, add the folder “MatlabCode” to the matlab path.



Run command “ImgSegment('Demo')” in Matlab, then a window will be shown with detected leaves. Use your mouse to click the leaves that you want to segment. Then hit “Enter” button.



After that a mask file will be generated. This mask image shows the detected individual leaves. You can use any image processing tool to modify this image, for example, separate one leaf into two leaves or smooth the edge of some leaves.



# Step 4: Leaf Angle Calculation

Finally, we get to the last step. This step uses all the generated data above to calculate the angle of each detected leaf. Change Line 20 “IMG\_DIR = r"Demo"” to the Demo folder. Then run “Python LADExtract.py” to get the results.

In this code, there are two options:

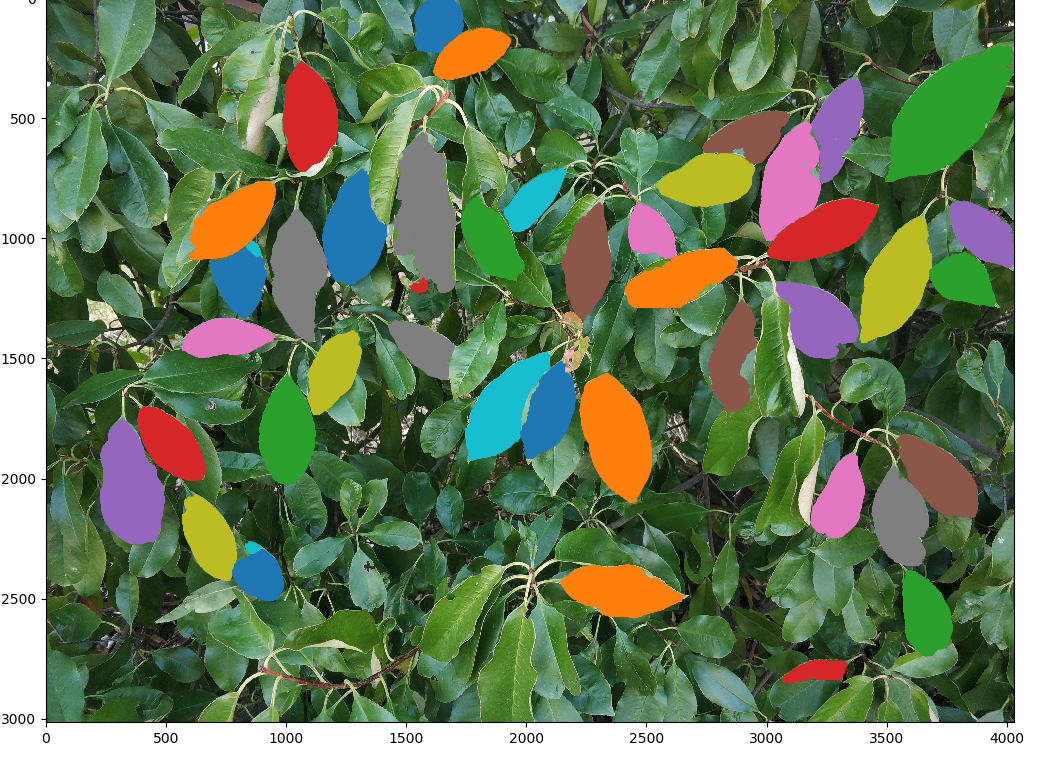
show\_angle\_on\_img = False

show\_proj\_img = False

If you set show\_angle\_on\_img = True, a image will be shown with labeled leaf angles:



If you set show\_proj\_img = True, a image will be shown with projected points:



(If you see this, congratulations to you! We will continue to optimize this script and a website will be provided in the near future.)