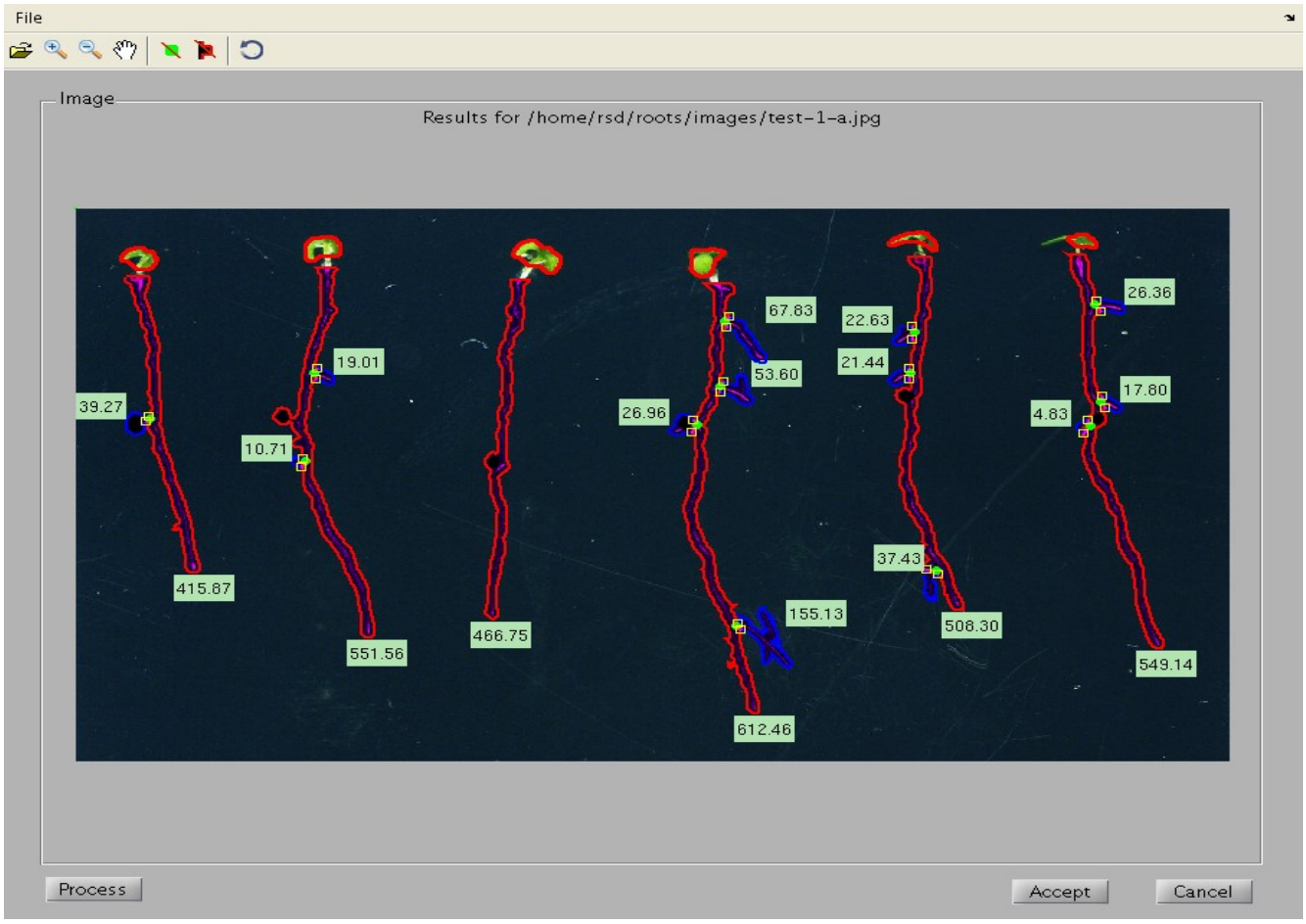

File
Image
Results for /home/rsd/roots/images/test-1-a.jpg
424.93
19.01
10.71
521.89
464.22
67.83
53.77
614.70
21.77
20.41
37.43
155.13
503.19
26.36
17.62
220.22
Process
Accept
Cancel


File
Image
Results for /home/rsd/roots/images/test-1-a.jpg
39.27
415.87
19.01
10.71
551.56
466.75
67.83
53.60
26.96
612.46
22.63
21.44
37.43
155.13
508.30
26.36
17.80
4.83
549.14
Process
Accept
Cancel

There is a constant in roots.m, clear_black_spots. When set to one the first image above results. When set to 0 image the second one results. You will notice that the first image above, a large section of the root is cut off due a discontinuity caused by a clear black spot.

When this situation arises, Manny suggests that user intervention is required. The user should be able to tell the program where the beginning and end of the root is.

In the algorithm there is a threshold, a flag that says if there is a black spot clear the map. If you don't clear the map.

The algorithm works like this: First you calculate the entropy, then calculate the derivatives. From this you get a map, where the roots might be. You save this, then you get a second map that also tells you where the heads are. It easy because then you calculate simply by looking at where the green spots are in each shot. Then you intersect the two maps, because each root needs to have a head. The intersection of these is your final map. Anything that is not in the intersection, we forget about it. The noise, and other junk will be outside of the intersection.

Getting the black spots is fairly easy because the energy of the blackspot is very low. We can filter the map by looking at these blackspots. It is easy to see in the above image, that any edge detection algorithm will think thats an edge. What happens is, can throw a third filter on top that says this is a black spot remove it. Now my contour is chopped in half. The mask is coming

Manny and I would like to know if the current algorithm allow you to select he spot and say go over it. Koray: masks are like separate object.

Is the feature there already to point out the black spots.?

get_roots.m

Masking operation. Take the image, convert it to double numbers.

```
imd = double(im);
imen = imd(:,:,1)+imd(:,:,2)+imd(:,:,3);
```

takes the image and converts it double numbers
energy is the sum of the rgb components, so now the energy becomes a mapping in the size of the image.

```
spotmask = imen<black_spot_energy;
```

this is a binary mask. The enegery is going to be 1, where the energy of the pixel is lower than the blackspot energy (block_spot_energy = 25 defined inside globals.m). IF the sum of the rgb components is maximum 25 then this pixel is corresponding to a blackspot.

In an image each RGB value goes from 0 – 255, therefore the total energy can go up to 765. energy value of 25 is very low, like 3%.

Disc of size 3, when you get a binary image something like this (kind of noisey:

0	1	1	1	0	
0	1	0	1	0	
0	1	1	1	0	(this is the black spot)
0	1	0	1	0	

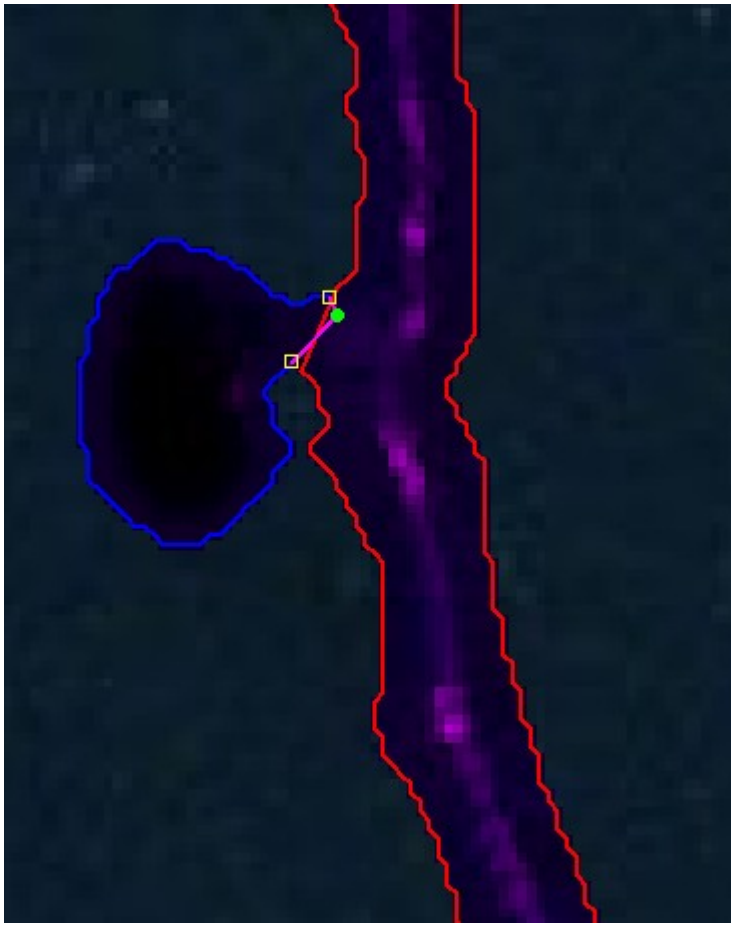
dilate it with a disc element with radius 3., its just going to make the
(I am under the impression that typically dilation means give any pixel with 1 neighbor becomes 1. It sounds like Koray's dilation, turns everything to a 1 inside the disc)
imdilate() is a morphological operation, basically an OR operation (makes everything 1 in the region). In essence, makes the area not a black spot.

If you want it to be considered as part of the root, make it not a black spot. Make a function, generates map of black spots, turn this binary map into object imobj(). Gives connection elements from binary map. You can select one of them, then it will clear it off the map. The next operation there is defining origin of the image with this map. Once you

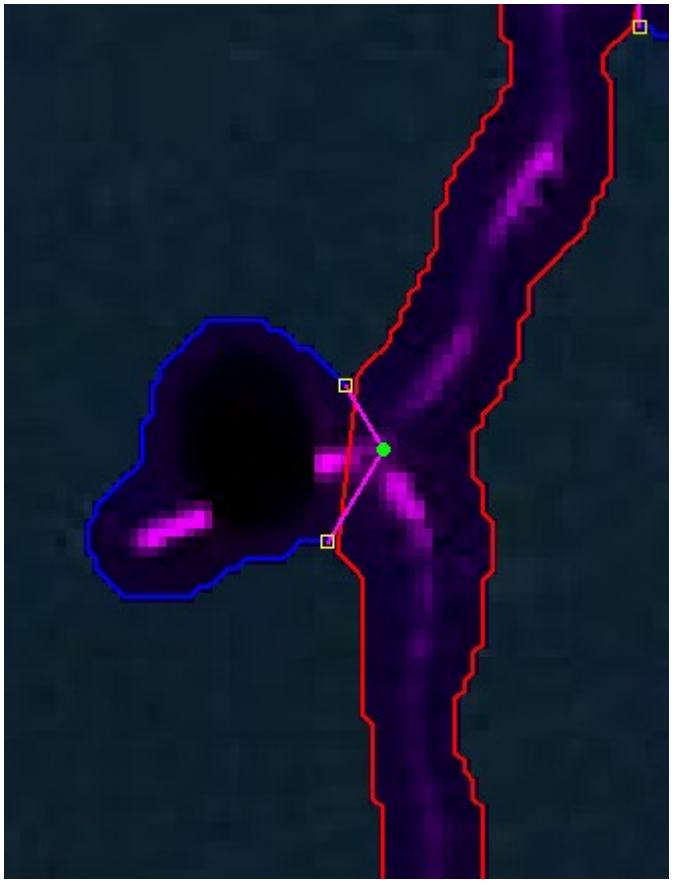
You get a binary map of black dots
turn this binary map, into objects <= find out the name of this matlab function
then select object
bn(obj) = 0

here out map, replaces the black dots with the average color of the image. It is not considered a black spot, but it is not considered a root. If I clear the black spot, then the root finding algorithm will now find the root there. If you set the black spot to zero, in he globals file, the root is actually found. There are other problems, it is an edge, so sometimes it will detect the black spot as a lateral root. Its going to think its a lateral edge because it cuts the plant.

The lateral root algorithm has things to avoid this type of problem, but there are still a few cases that it is considered a root.



Black spot as lateral root



black spot on lateral root

Manny: maybe we can set it up so the user can manipulate the image to fix these errors. For example if there is blackspot, instead of user just saying clear black spot, actually the use can change the color of the pixel – so maybe there is an extending the root function. That will actually put the color of the root connecting the two paths then recalculate root.
(basically a paining function is needed)
Manny: is there a function to do it already?
Koray shows that there is already a function to cut off a lateral root that is not there.