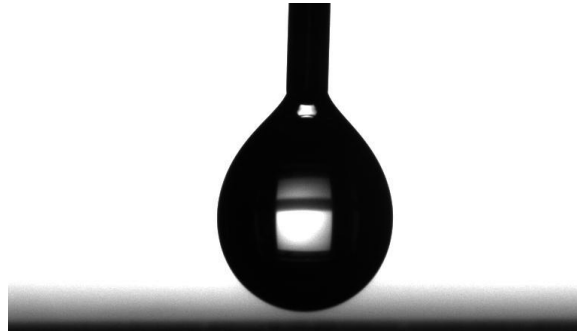


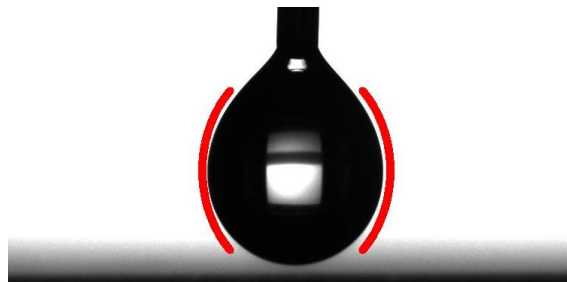
Start with droplet image.



Display these circle 'cross-hairs' on the camera display



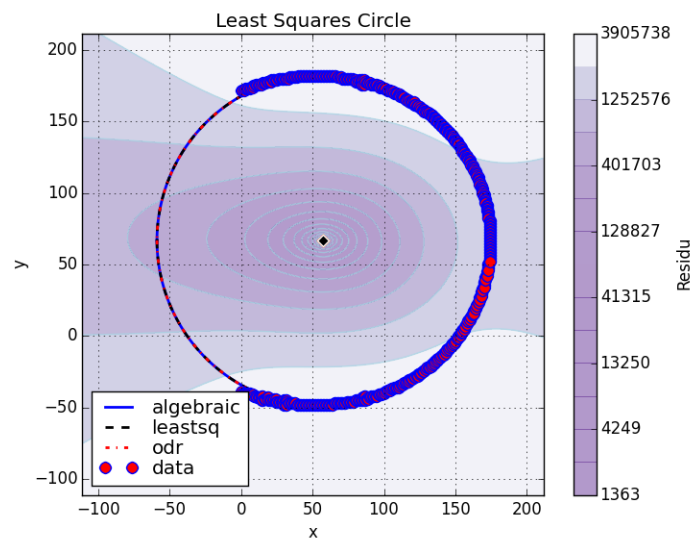
So the user sees that they need to center the droplet and move it into place. Once they do, it's way easier to crop and filter to get an outline of the bottom half of the droplet. I did it manually in the image



So then I can just do some automatic cropping and filling because the program assumes that the droplet is centered between the crosshairs. So now after filtering, the droplet looks like this:



The edges get detected and fed into some semi-circle least-squares program to find a circle that fits the curve. The data and fits are here (ignore the residue stuff for now, and note that the drop is laying on its side so the bottom is on the right):



The radius calculated is $R = 116.101$ pixels. You can see in the fits, though, that the higher up on the droplet (left of center, on this image) the less it is fitting with the correlations. So that data might want to be ignored for this radius calculation. Right now the amount of droplet used for the edge-finding (3rd figure) is arbitrary. I could probably scan through a whole range and see if they converge upon a single radius value. Meaning, I focus more on just the bottom of the droplet.