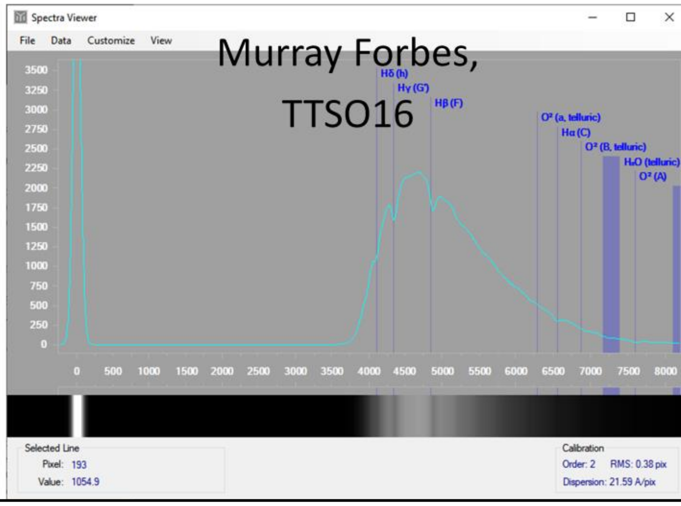
A possible improvement to alignment of spectra in Tangra

Murray Forbes,
TTSO16

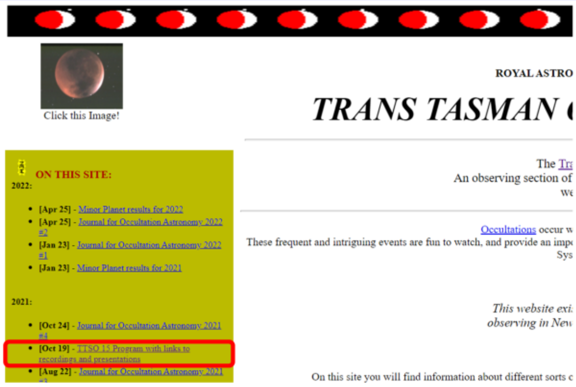


While this talk is a bit ‘off-topic’ for TTSO16, I hope it will be of interest to you as it’s about one of the features of Tangra.
This presentation is in two parts; The first is a run-down on how to use the spectroscopy feature and the second is about a glitch I found in it and how I hope to fix it.
P.S. The graph shown here is from Tangra’s reduction of a spectrum I recorded of Sirius.

Previous talk at TTSO15

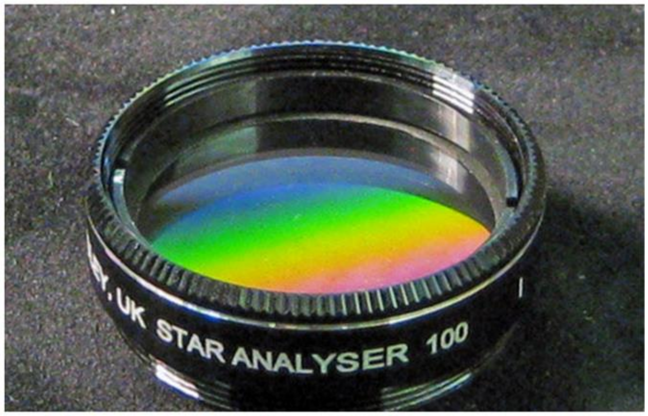
Trans-Tasman Occultation Alliance website:

- http://www.occultations.org.nz/index.htm



Those of you who attended the last TTSO will (hopefully) remember my presentation on Tangra and OccuRec – two programs written by Hristo Pavlov and made available as open source for the occultation community to add features and make bug fixes. As I covered the process and my contributions to date as of last year, I don't propose to repeat the details here. If you didn't attend TTSO15, or did but would like to refresh your memory, just go to the Trans-Tasman Occultation Alliance's web page and look for the TTSO15 program in the 'On this site' frame.

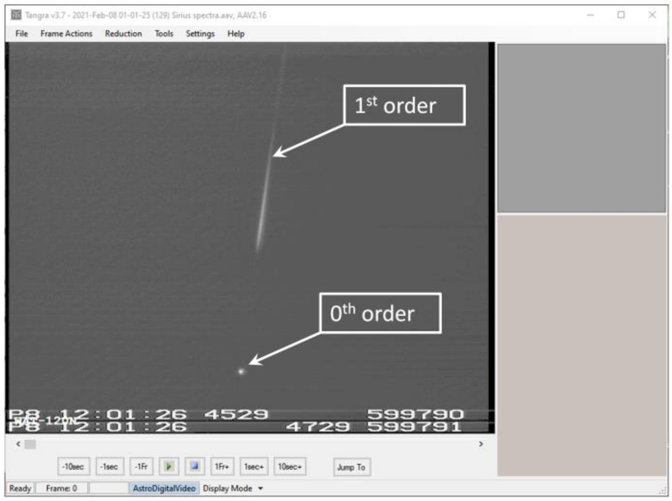
The SA100 SpectroGraph



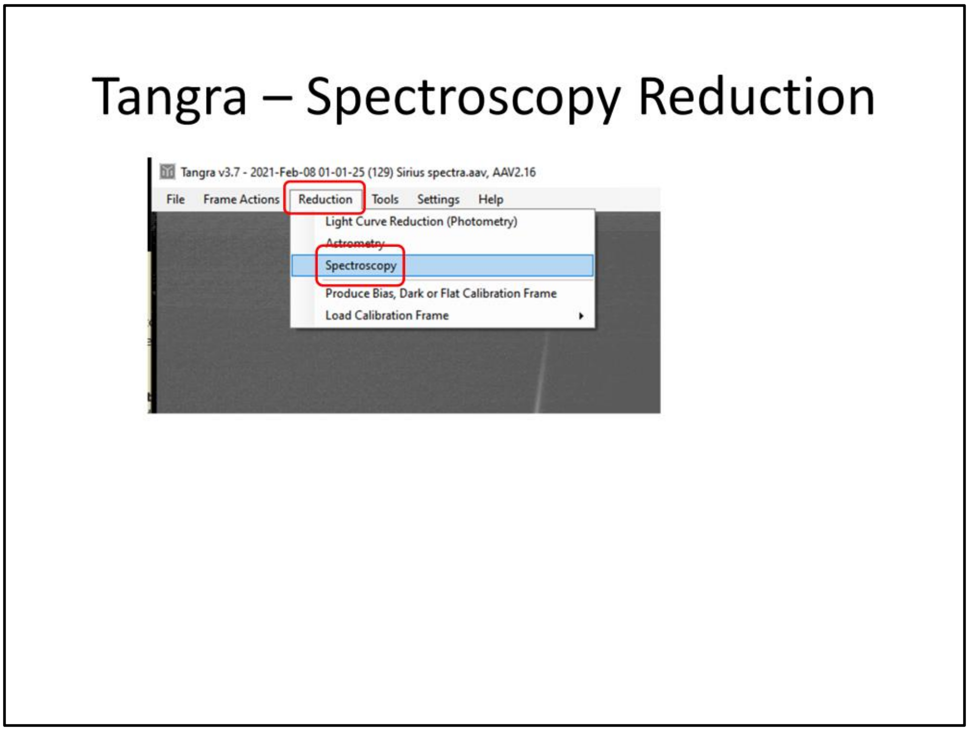
https://www.patonhawksley.com/astronomy-1

I observed Sirius using my usual occultation equipment, but with one addition. This is the Star Analyser 100, which is a slit-less transmission diffraction grating with 100 lines per millimetre and blazed in the first order. It is mounted in a standard 1.25 inch diameter threaded cell and I fitted it immediately in front of my Watec 120N+ video camera. If anyone’s interested in getting in to spectroscopy I would recommend this as a good starting point as the SA100 is very affordable, at only ~100 Pounds.

Tangra - Image of Sirius

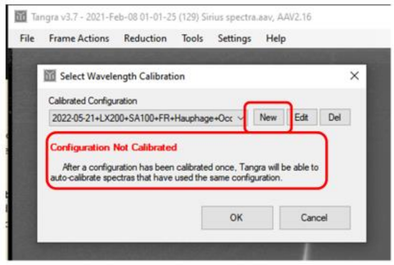


This slide shows a single frame from my recording of Sirius. The ‘zeroth order’ is the image of the star itself while the ‘1st order’ is the spectrum. This is my first attempt at using the SA100 and I haven’t yet used the locking ring provided to rotate the SA100 so the spectra is in the recommended horizontal orientation with the star on the LHS.



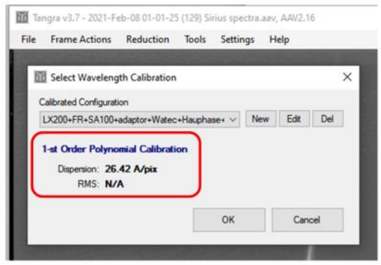
Once you've opened your video recording in Tangra, the 'Spectroscopy' function is in the 'Reduction' menu.

Tangra – Spectroscopy Reduction



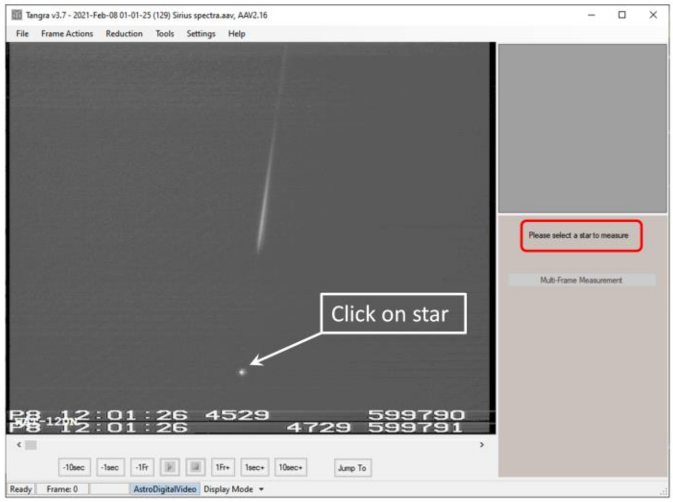
This opens the ‘Select Wavelength Calibration’ form. The first time you use this the ‘Configuration Not Calibrated’ message is displayed and <click> you’ll need to create one via the ‘New’ button. I’ll cover how to do this a bit later in the presentation.

Tangra – Spectroscopy Reduction

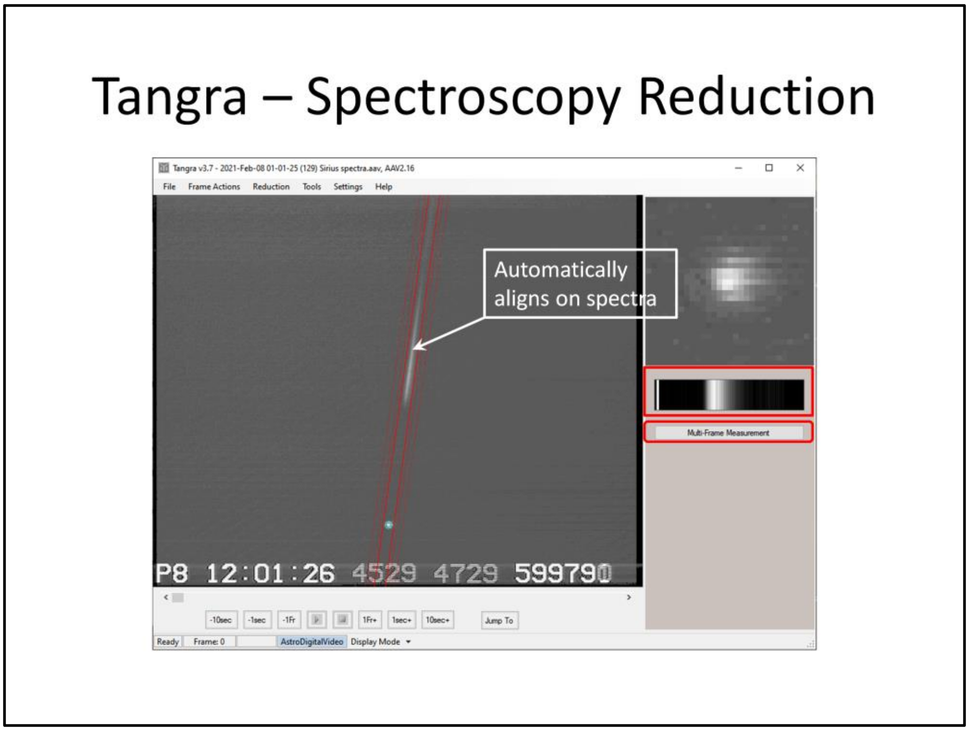


For now, let's use a calibration I'd done previously. Note: The dispersion factor depends on the particular equipment configuration used, and especially on the distance between the SA100 and the camera.

Tangra – Spectroscopy Reduction

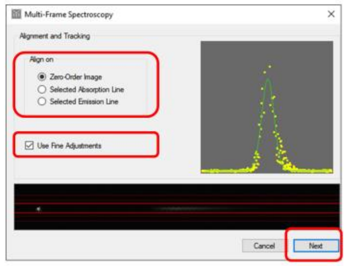


Once the calibration has been selected, Tangra prompts you to click the mouse cursor on the star.



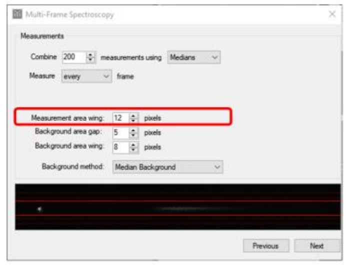
Tangra then draws a cyan circle around the star and attempts to automatically align on the spectra, as shown by the red lines. <click>The brightness between the innermost red lines is shown over here, with the width of the trace artificially broadened for easier viewing. <click>The next step of the process is started by clicking the ‘Multi-Frame Measurement’ button.

Tangra – Spectroscopy Reduction



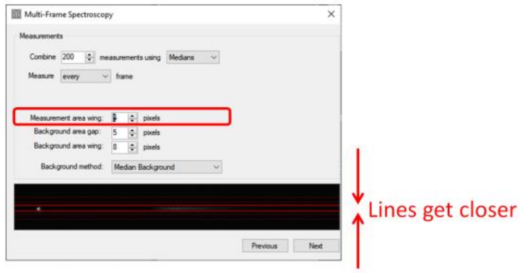
<click>You now have the choice of aligning from one frame to the next using either the star’s image, or an absorption or emission line.
<click>I don’t know exactly what the ‘Fine Adjustments’ does yet, but once I figure it out I think I’ll add a tooltip to give a better (longer) description.
<click>Once you’re happy with your choices, click on the ‘Next’ button.

Tangra – Spectroscopy Reduction



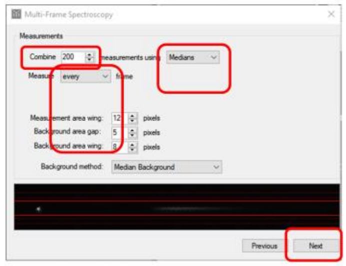
The next form allows you to adjust the width of the measuring wing, the outer wings used to measure the background and the gap between the two.
<click>I've found you get a better result if you reduce the measurement wing to

Tangra – Spectroscopy Reduction



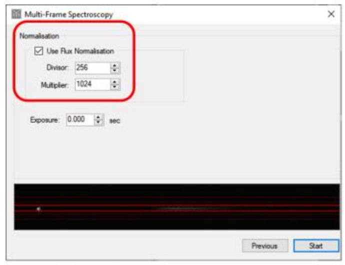
Just include the star image and spectra line.

Tangra – Spectroscopy Reduction



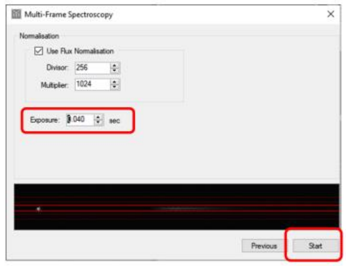
You can also choose how many frames will be combined to make the spectrum, <click>if only every nth frame will be used, and <click> if the frames will be combined by averaging or taking the median. Medians take much longer to calculate than averaging but they're better if you have any outlier frames that, for instance, had a satellite streak across them.

Tangra – Spectroscopy Reduction



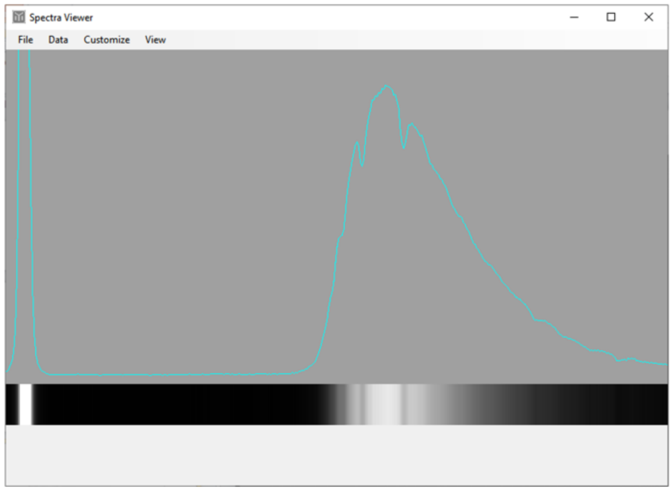
As far as I can tell, if the ‘Flux Normalisation’ checkbox is ticked then the intensity of every pixel is scaled by the multiplier and divisor before being combined with the previous frames. Here for instance, every pixel is multiplied by 1024 and then divided by 256 before being added to the same pixel in the previous frames.

Tangra – Spectroscopy Reduction



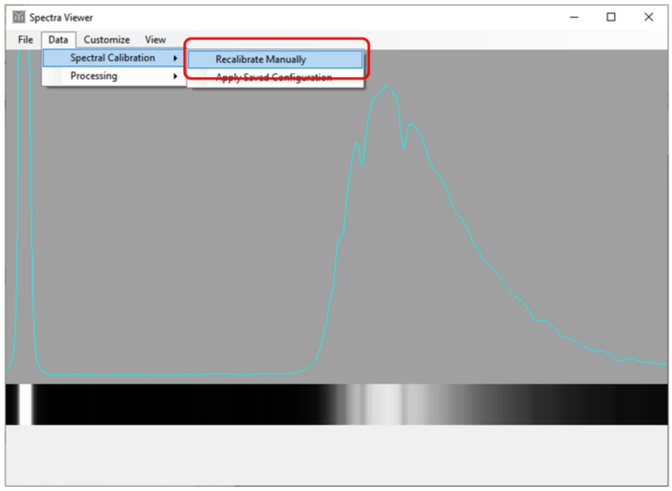
Tangra requires you to enter the exposure time before the <click> ‘Start’ button will work, although Tangra doesn’t seem to use the value for anything. For example, nought point nought four seconds is the standard PAL image rate of 25 frames per second.

Tangra – Spectroscopy Reduction



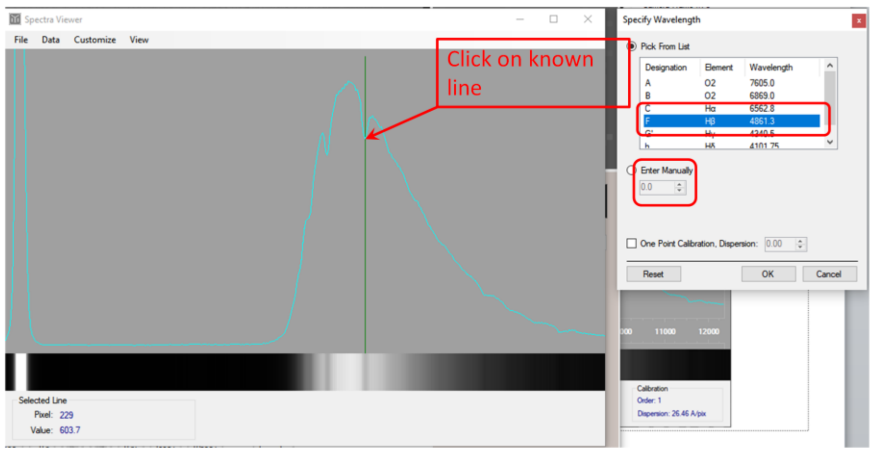
So after running for a while, the ‘Spectra Viewer’ window opens with the averaged spectra, in both a plot and intensity bar. It may, or may not, have a wavelength scale along the horizontal axis.

Tangra – Spectroscopy Reduction



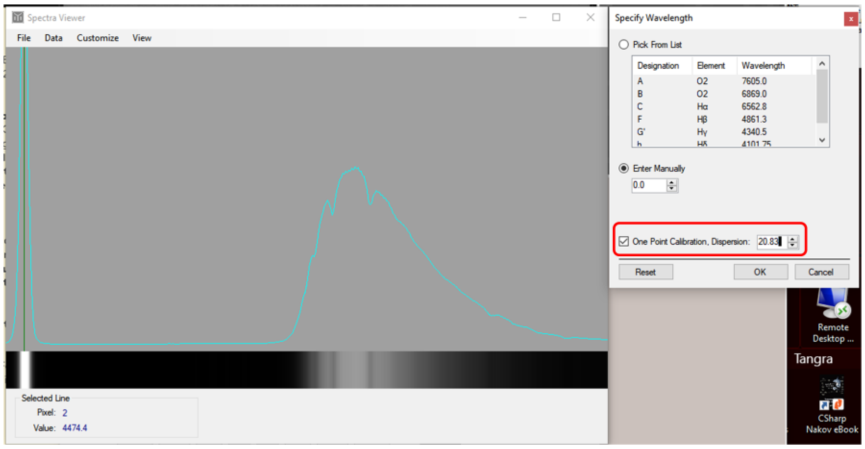
If it doesn't have the scale, then you can either apply a previously saved calibration or manually do the calibration procedure now. Here I'm going to run through the calibration procedure, as promised earlier.

Tangra – Spectroscopy Reduction



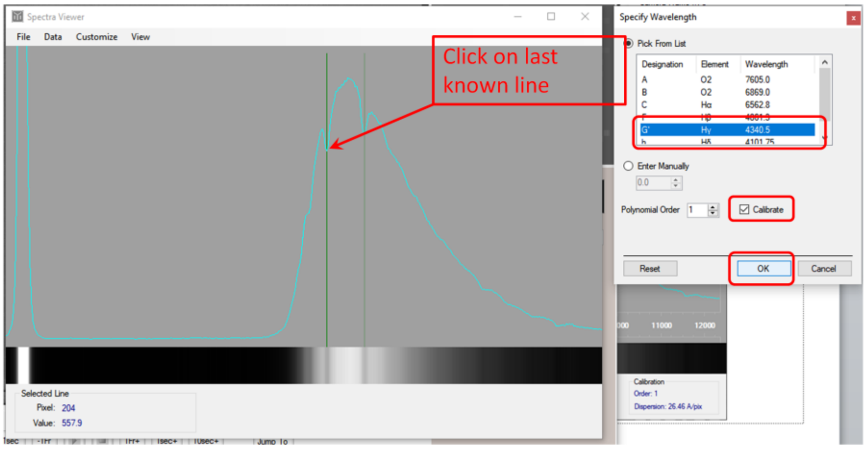
The first step is to click the cursor on a known absorption or emission line, causing the vertical green overlay line to appear. The position of this line can be fine-tuned by using the keyboard's left or right arrow keys. Then select that feature's wavelength from either <click>the drop-down list or <click> enter it manually.

Tangra – Spectroscopy Reduction



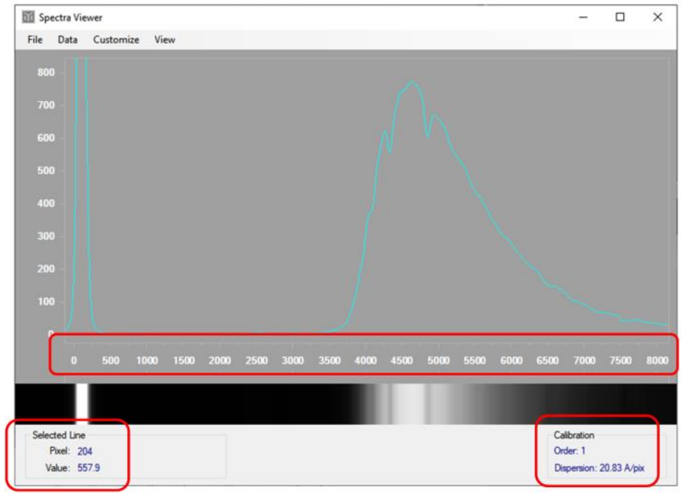
At this stage you have the option of doing a 'One Point Calibration', *if* you already know the dispersion. Here I've entered a dispersion of 20.83 Angstroms per pixel.
In this case, I also used the manual entry to enter the wavelength's zero position. This is in the centre of the zeroth order, i.e. the star's image on the LHS.

Tangra – Spectroscopy Reduction



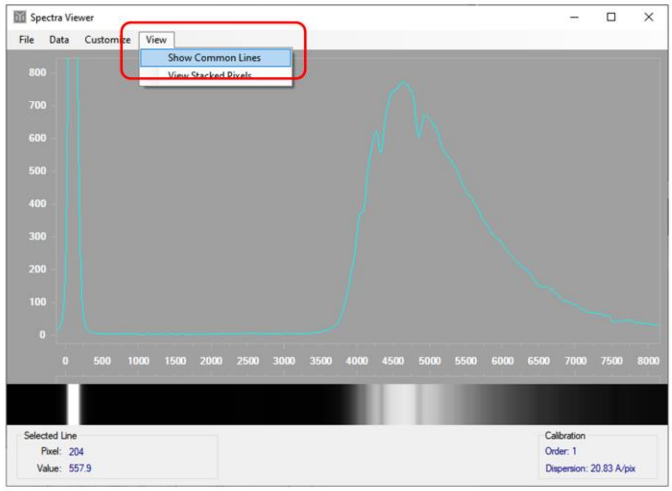
Assuming you wanted to select more than one known feature, just keep selecting features with the cursor and <click> entering their wavelength until you get to the last one.
<click>Next tick the ‘Calibrate’ checkbox and choose the ‘Polynomial Order’. Enter ‘1’ for a straight line, ‘2’ for a quadratic fit etc.
<click>Finally, click the ‘OK’ button.

Tangra – Spectroscopy Reduction



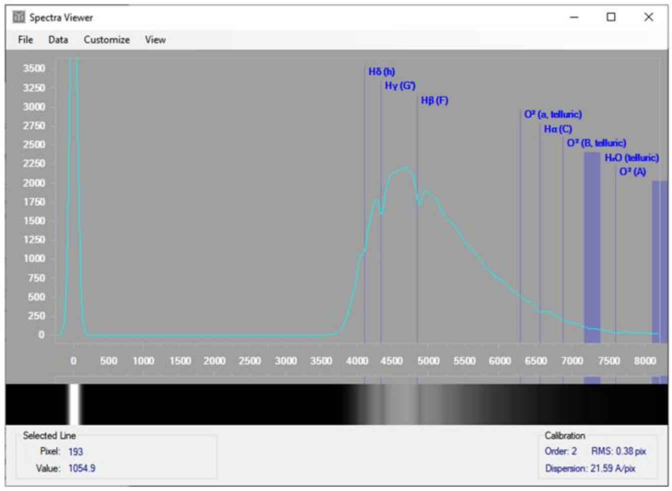
The wavelength scale is now drawn on the horizontal axis, <click>with the results of the calibration shown on the bottom, RHS of the window.
<click>If you click the cursor anywhere on the graph, the information in the ‘Selected Line’ area on the bottom, LHS of the window will change to display the wavelength and intensity of that line.

Tangra – Spectroscopy Reduction



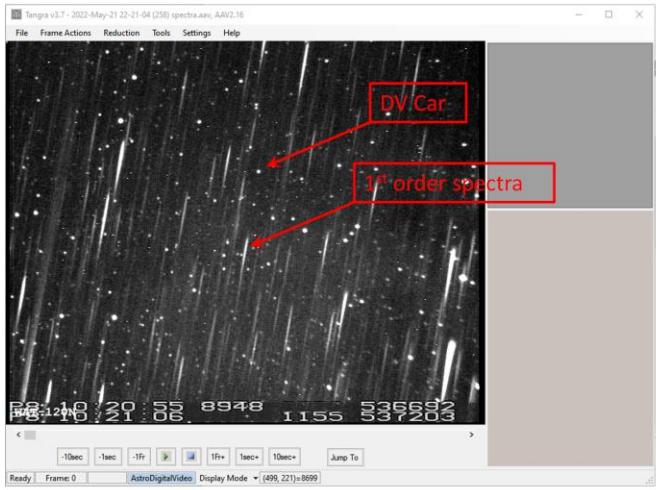
The final feature in Tangra I'm going to mention is the 'Show Common Lines' option in the 'View' menu.

Tangra – Spectroscopy Reduction



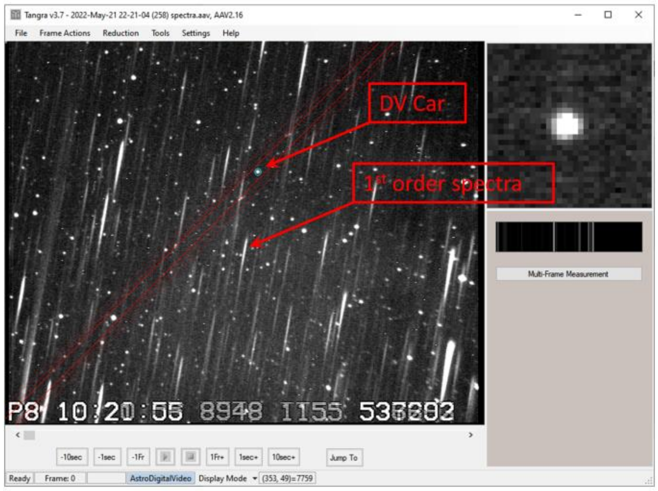
Choosing that brings up this overlay on a calibrated graph. Here the first four lines of the Hydrogen Balmer series are shown, along with the Oxygen and water absorption lines or bands caused by the Earth's atmosphere.

So what's the problem?

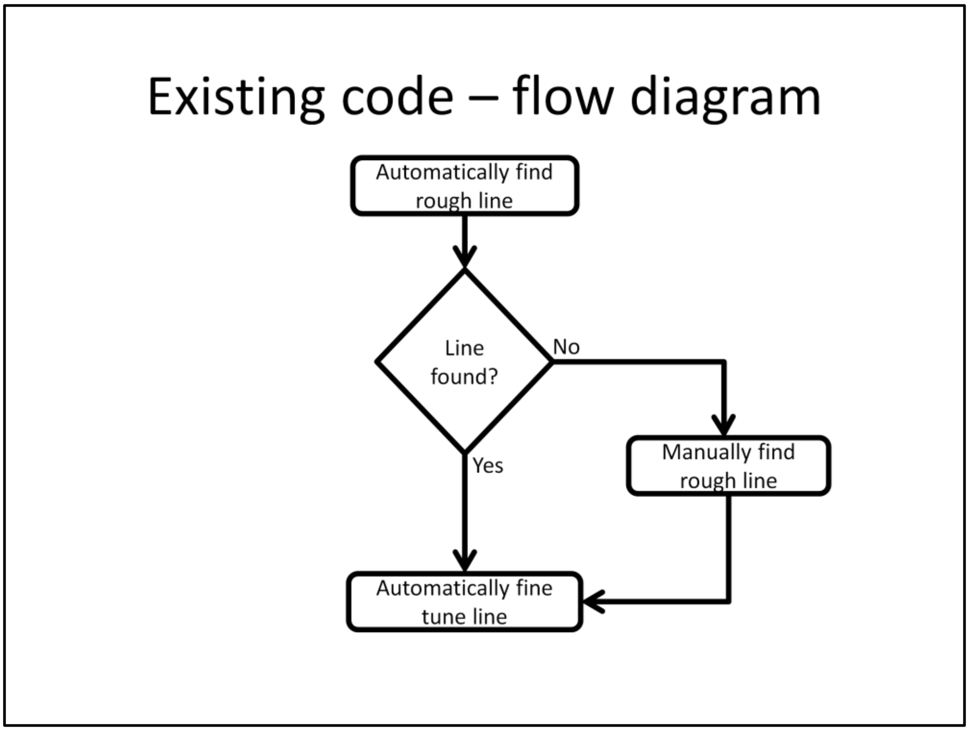


Now we're at the second part of my presentation, where I describe the problem I found and how I hope to fix it.
I've been observing DV Carina, an eclipsing binary star system, for a few years now using my occultation equipment. At magnitude 10.4, it is a lot fainter than Sirius and so quite a few stars are visible in the frame when my integrating video camera is set to see DV Car. So this means when I place my SA100 into the optical chain, I get a lot of spectra. The image here shows my first attempt at this – once I get things working properly I will investigate different orientations of the camera relative to the SA100 so that both my target star and comparison stars spectra do not overlap with other stars.
P.S. Without the SA100 in the optical train I use a Slow5 (x16, or 16 / 25 = 0.64 s) integration setting. With the SA100 I had to increase the video camera's integration setting to its maximum value Slow9 (x256, or 256 / 25 = 10.42 s), so magnitude 10 to 11 looks to be the limit of my equipment for spectroscopy.

So what's the problem?

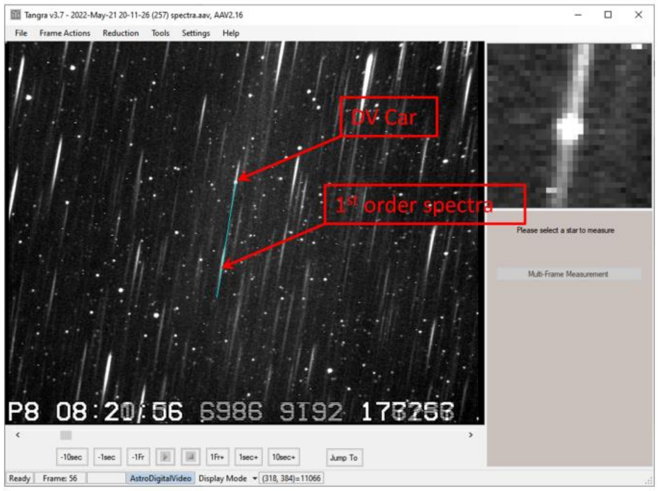


Recall how clicking on the star is the first step in the reduction process, where Tangra then automatically tries to find the spectra. In this case, its failing and seems to be tricked by the bright star to the lower left of the frame.



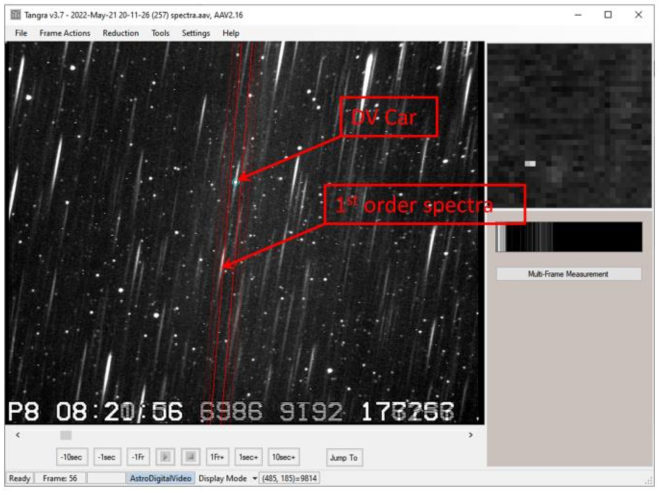
This flowchart is a summary of Tangra's code for finding the spectral line in the image. Although Tangra failed to find the spectra in the frame shown in the previous slide, in most cases Tangra doesn't offer to let me manually pick the line, i.e. it thinks it did find the line.

So what’s the problem?

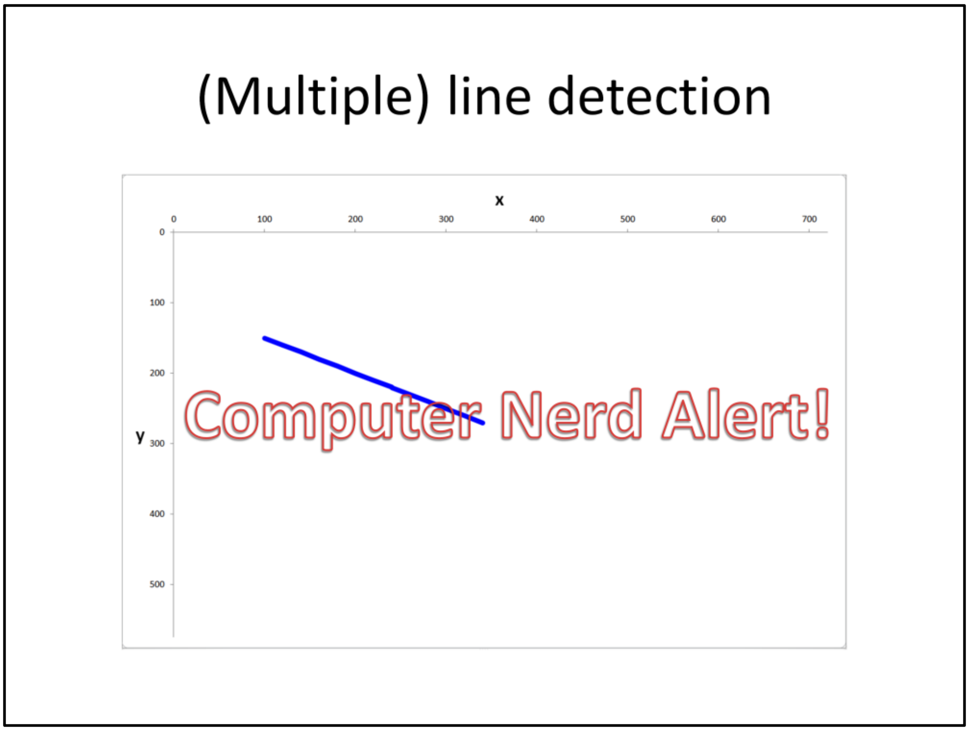


If I pick a different frame to start the reduction process occasionally Tangra will prompt me to manually pick the line. This image shows a frame *as* I was selecting the right spectra, by dragging the blue line around with my mouse.
P.S. This is taken from a different video than in my previous slides, taken on a different night with the camera rotated by a different amount - the spectra all have the same angle as before but the star field has rotated approximately 90 degrees.

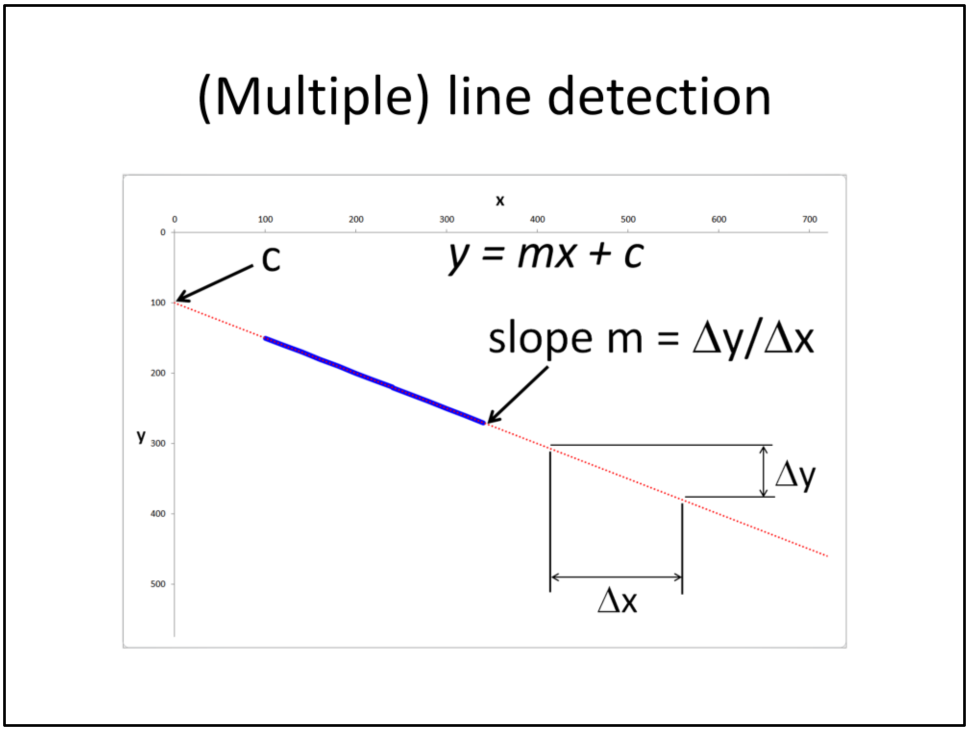
So what’s the problem?



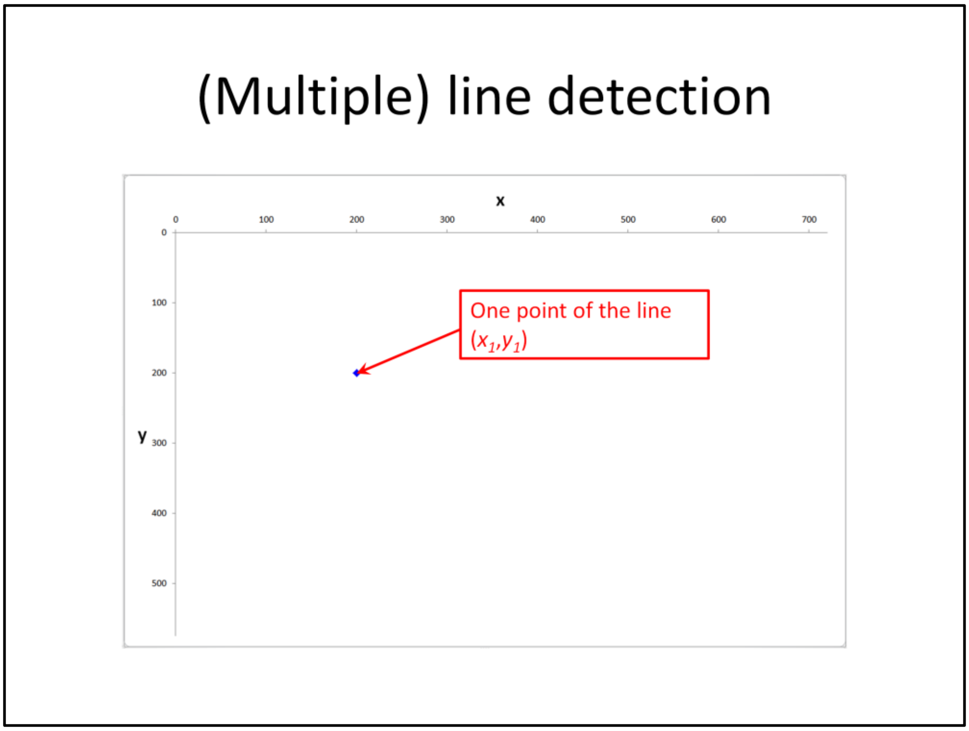
Unfortunately the fine tuning step, done by Tangra after my manual alignment, still fails.



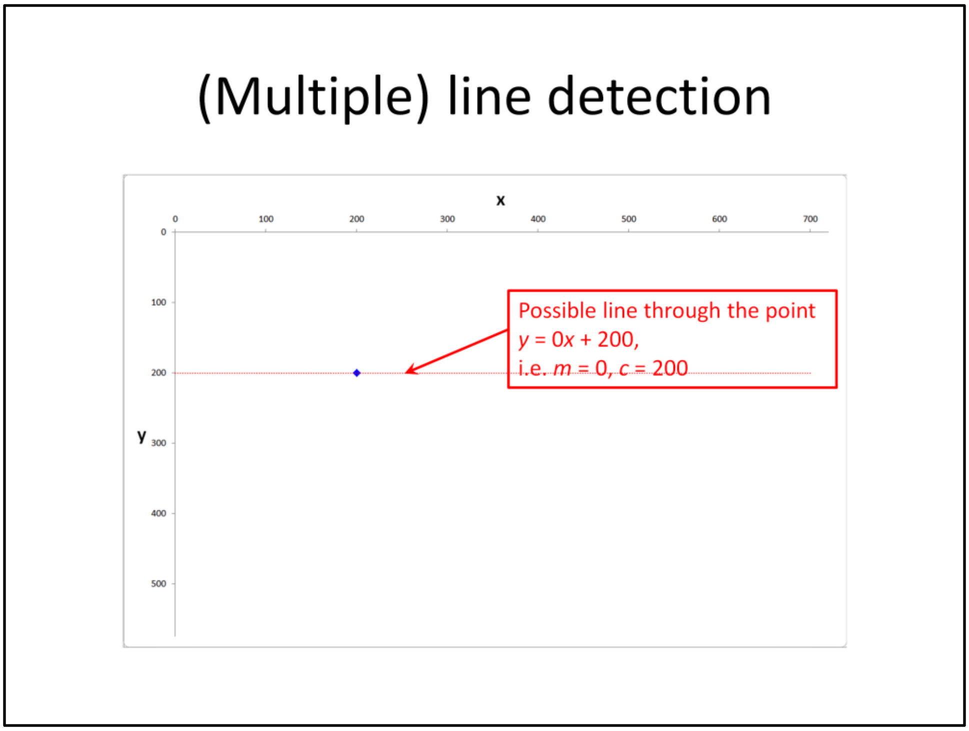
Just a warning that this part of my presentation gets a little technical, with some equations and image processing – stuff that as a science nerd I find fascinating. <click>What Tangra’s trying to do boils down to finding a line, or line segment, within an image – for instance the blue line segment shown here. While this type of problem is easy for people to solve – remember the previous lines of the spectra in the Sirius and DV Car images – it’s not so easy figuring how to program a computer to do it.



We can describe the line mathematically as y equals m times x plus c, where m is the slope of the line and c is the value on the vertical y axis where the line crosses this axis. Hopefully you remember this from back in secondary school maths.

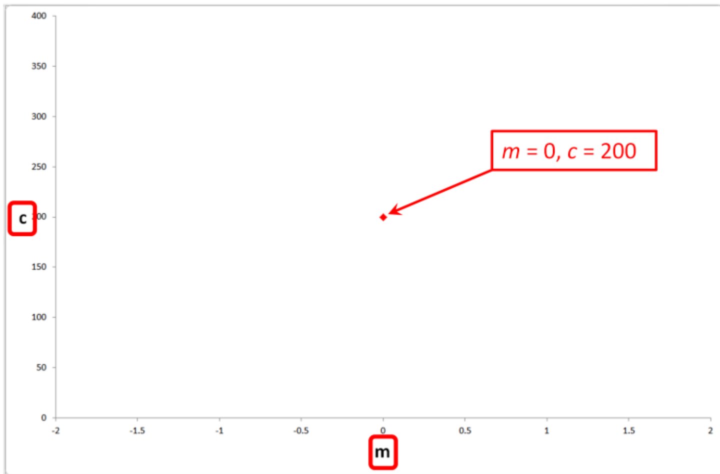


Now the procedure of my proposed 'improved' code works by scanning over the image one pixel at a time looking for a bright pixel that *could* be part of a line. Here I've shown one pixel of the line from the previous slide.

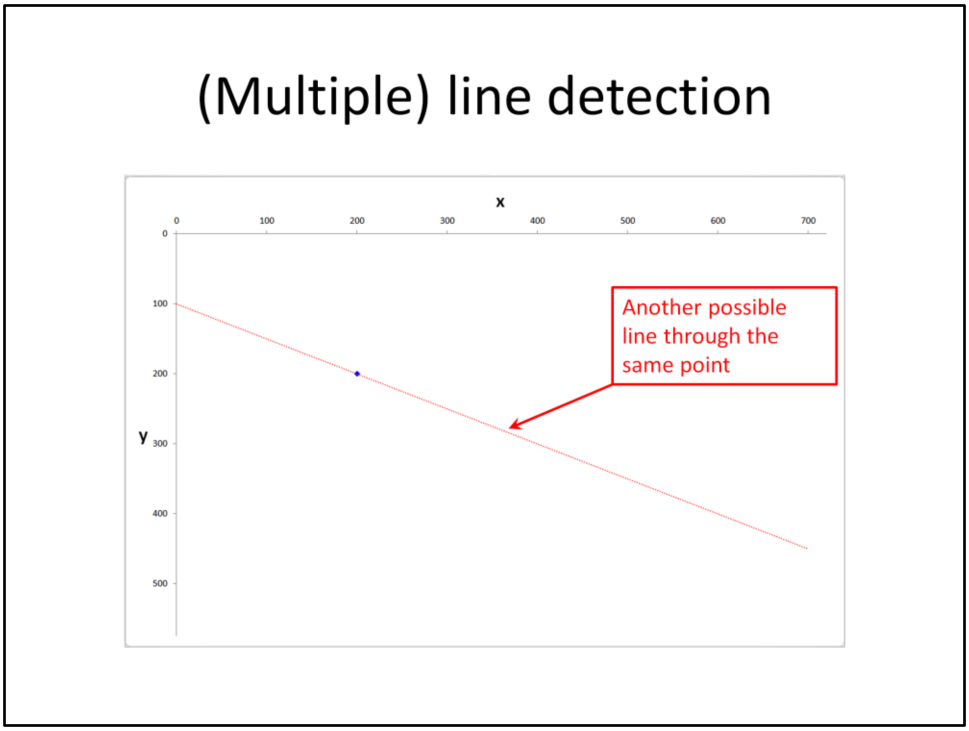


At this stage we, well actually the computer, doesn’t know which line ran through the pixel. It could have been the line shown here. This example is a horizontal line, i.e. it has a slope of zero and crosses the y axis with a value of 200.

(Multiple) line detection

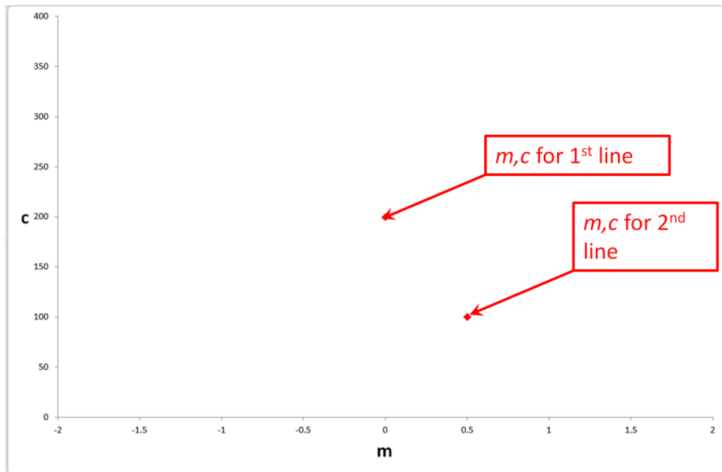


Previously we were showing the pixels and lines on a graph using x and y values for the axis, giving a direct representation of the image. We could instead represent a *line* from the previous graph as a *point* on a new graph that uses the m and c values for the axis's. So a line that had a slope of m equals zero and an intercept c of 200 becomes a point with those values as its coordinates.

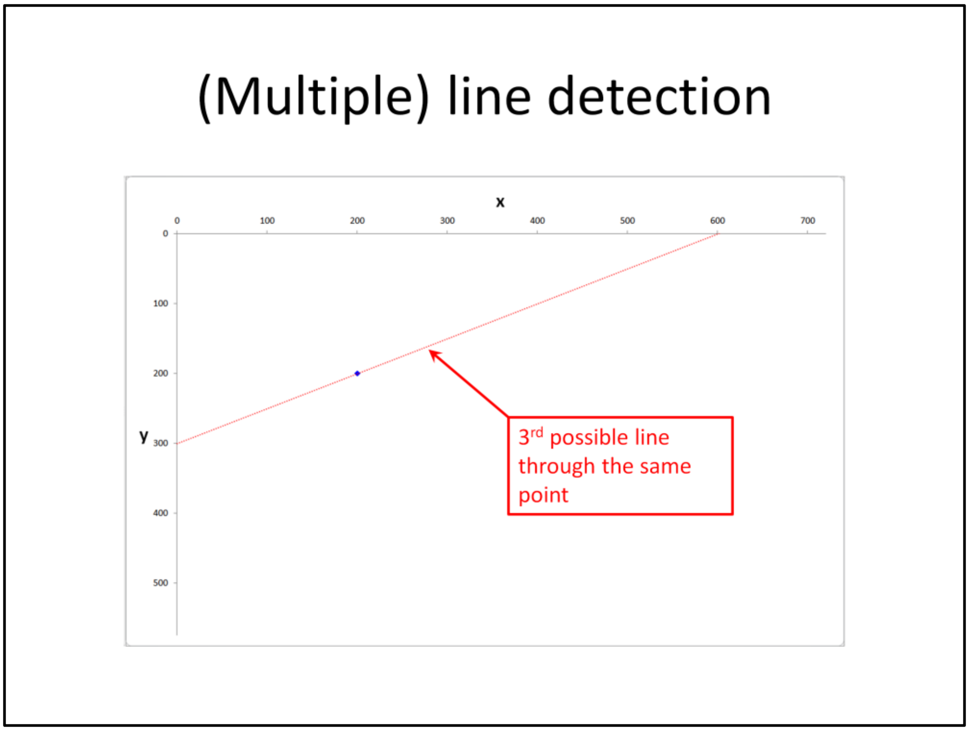


Remember that point on the m, c graph was for one of the lines that runs through the pixel, which *might* be the actual line that the pixel is part of. Here I've shown another line running through the pixel that might be the actual line. This has a different slope m and intercept c than the previous line and ….<click>

(Multiple) line detection

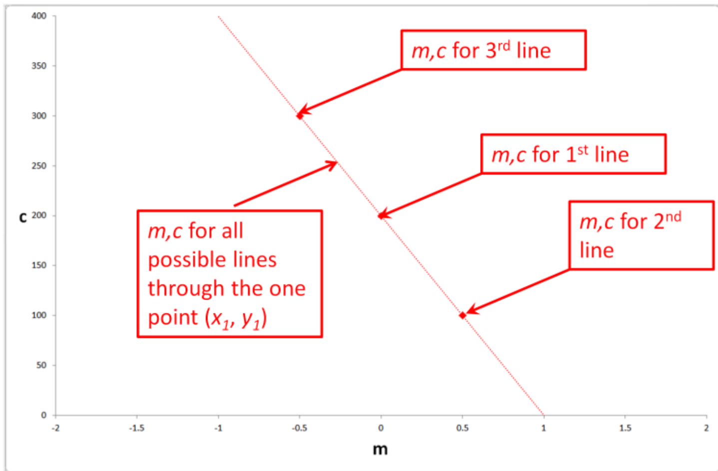


Can again be represented by a point in the m, c graph



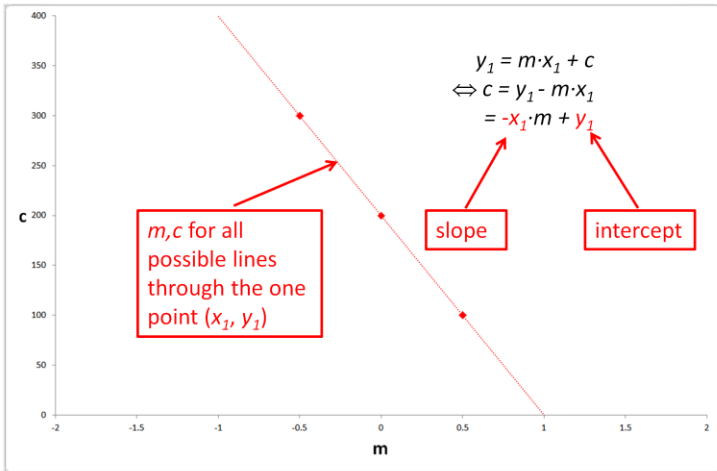
If we do one more possible line through the pixel then …<click>

(Multiple) line detection

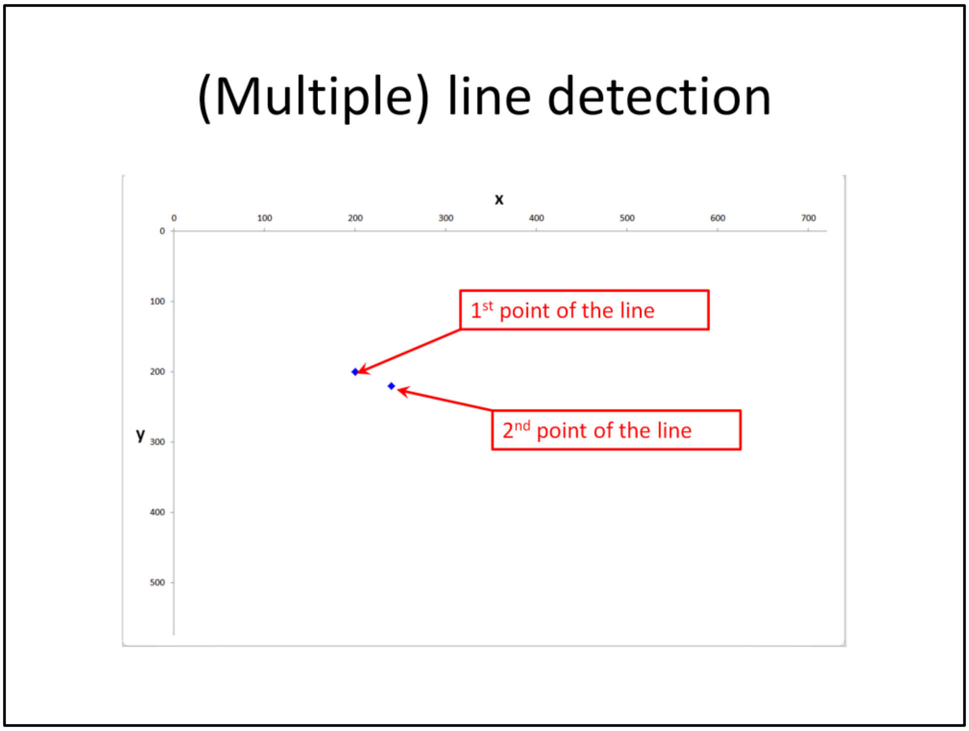


Then it should now be clear that all of the possible lines through that pixel can be represented by a single line in the m, c graph.

(Multiple) line detection

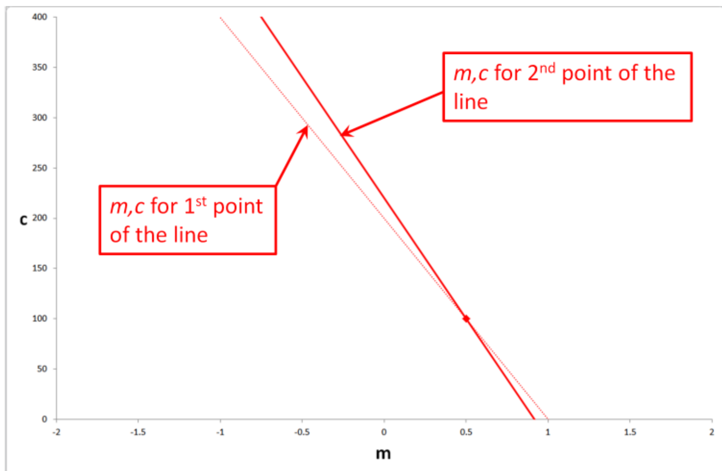


In fact, you can prove this by simply re-arranging the equation for a line with slope m and intercept c that goes through the point (x1, y1) to a line with a slope of minus x1 and intercept of y1.



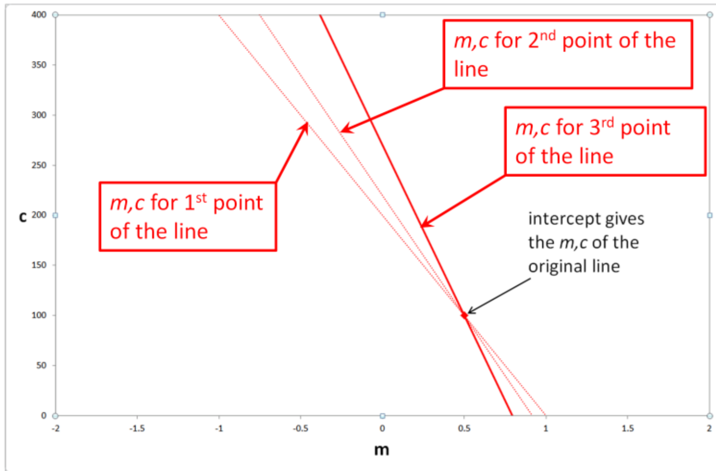
So what happens if we now consider a second pixel that is part of the line segment we're trying to calculate in the image?

(Multiple) line detection

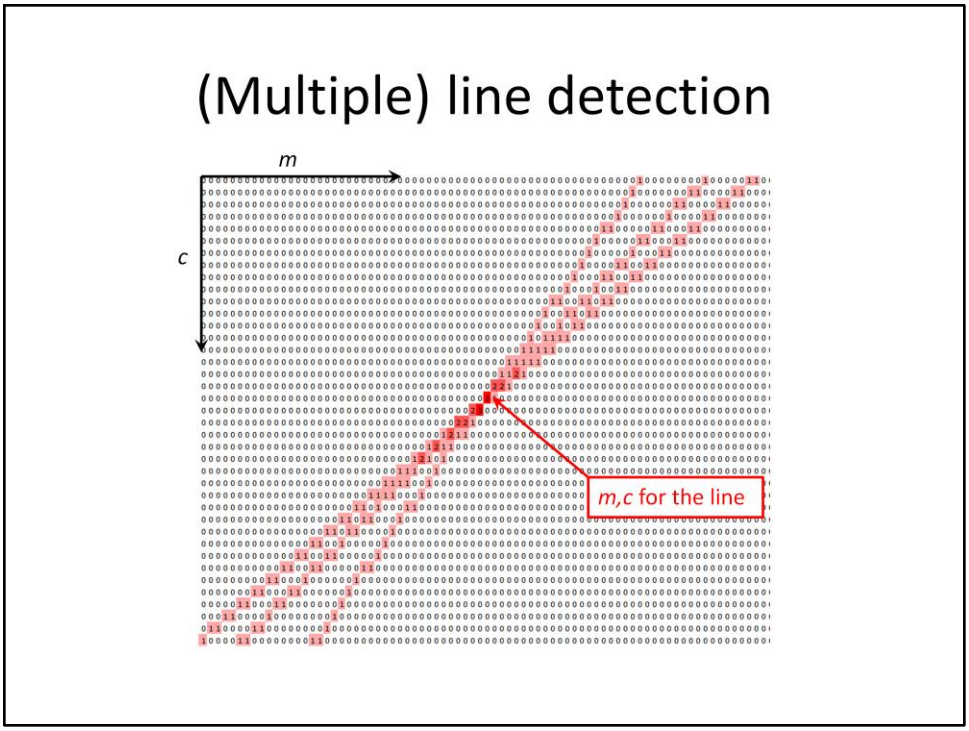


We again get a single line in the m, c graph that represents all possible lines through the second pixel.

(Multiple) line detection

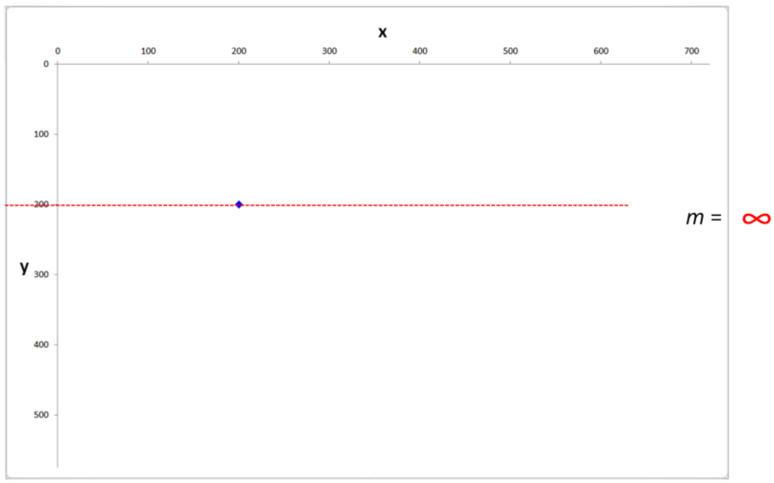


And if we repeat this yet again for another pixel of the line segment, we get a third line in the m, c graph. Notice that the three lines all intersect at the same place. In fact if you were to repeat this for every pixel in the line segment, all of the resulting lines in the m, c graph intersect at the same point. Even better, the m, c coordinates of the intersection are the slope m and intercept c values of the line segment.



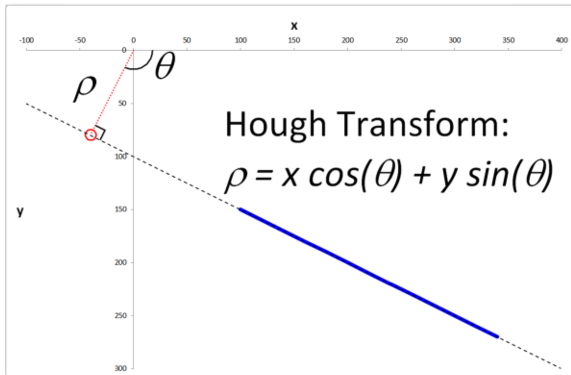
The previous slide seemed to suggest we needed to calculate the intersection of a large number of lines in m,c graph – which at first glance appears to be swapping the original problem for another that's not much easier. However, if instead of trying to solve the equations, let's just draw the lines on an image by adding the number one to each pixel in the image if it's part of a line. This means the intersection of the lines will have the most ones added together in that pixel – so now the problem is just the trivial one of finding the maximum value, i.e. brightest pixel, in the image. To make it easier for us humans to see in this presentation, I've colour coded the pixels using a heat map, with the zeros having a white background, a one with a faint red background and the maximum (brightest) pixel with a bright red background.

(Multiple) line detection

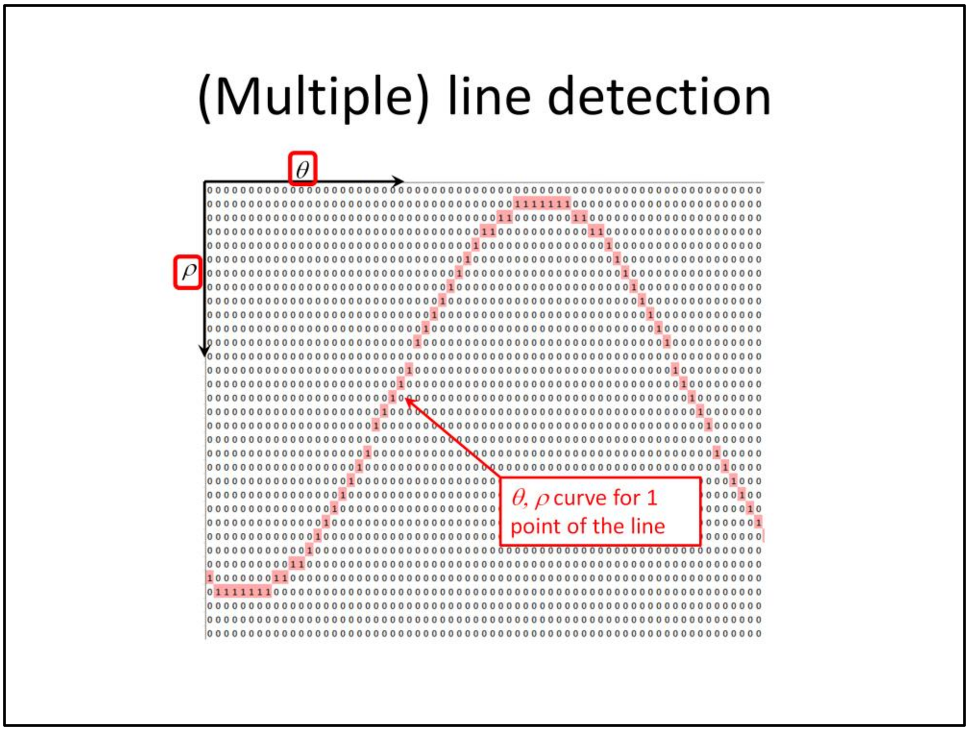


However there is a major problem with detecting a line using the slope m and intercept c to describe the line. Let's start with a horizontal line, as shown here. This has a slope of zero.
<click>Obviously as we go to a steeper line the value of m gets larger. The problem is when the line becomes vertical – then the value for the slope becomes infinite (and the computer program usually crashes at this point).

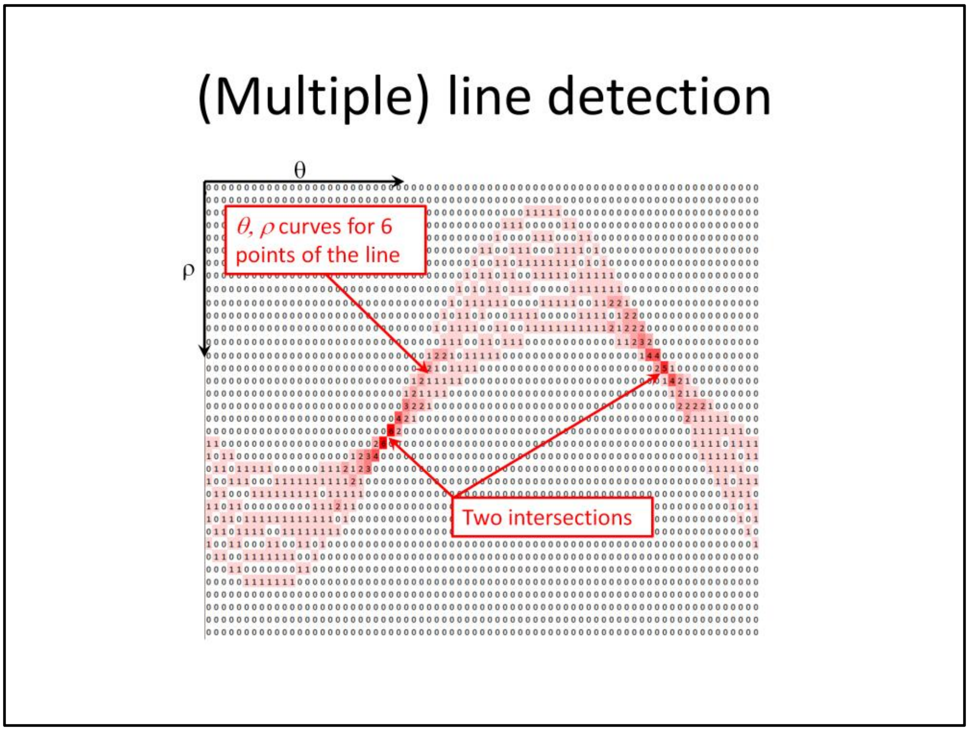
(Multiple) line detection



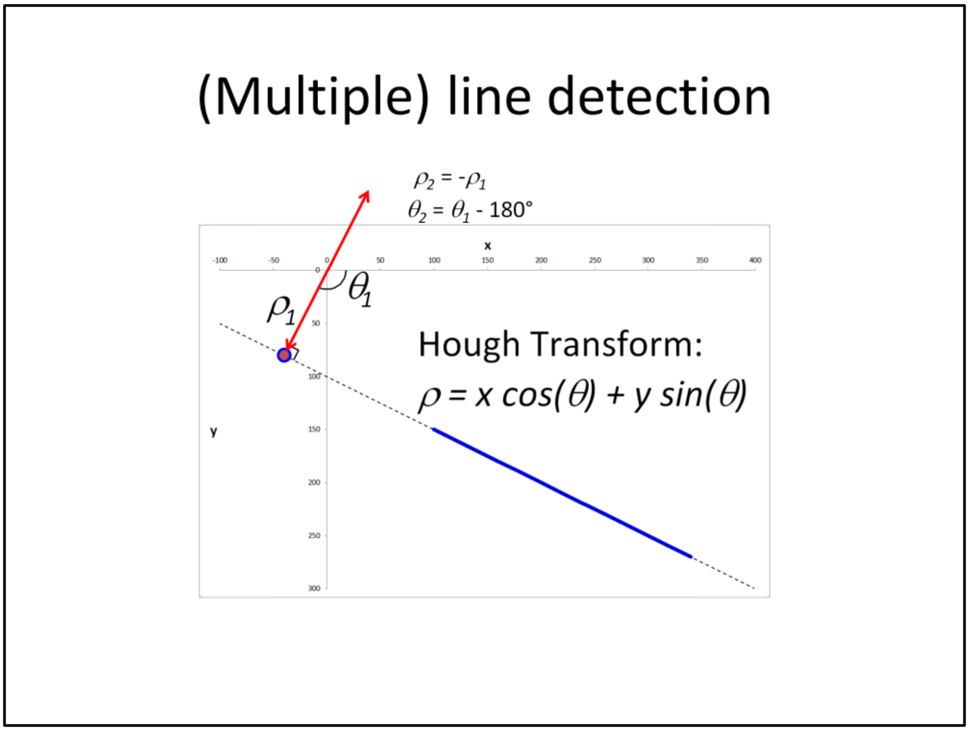
So what we need is another way to describe a line that doesn't have these pesky infinities. Fortunately we can do this by using the values of rho and theta, as shown on the graph here. Rho is the shortest distance between the line and the origin of the plot, shown by the red dotted line. Theta is the angle between the horizontal axis and the red line. Theta always has a value between nought and three hundred and sixty degrees, with theta being zero for a vertical line and ninety degrees for a horizontal line. The equation for a line in terms of rho and theta can be rearranged to a form as shown here, which is more useful for the line detection procedure and is known as the Hough transform.



If we repeat the same procedure that we did for the m,c plot but using rho and theta instead then for a single pixel in the original x,y image we end up with a sine wave (rather than a straight line).

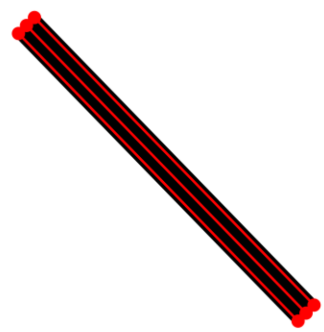


I’ve repeated that here for six pixels on the line, with the same heat map used in an earlier slide. As before, the curves overlap at points that correspond to the correct values of rho and theta for that line. The difference is this time there are two intersections, not one – so what’s the story with that?



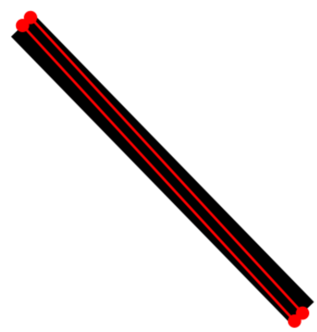
One of the intersections are the values of rho and theta I described earlier, here labelled with a subscript '1'.
<click>The other is like a mirror image, with theta 2 being 180 degrees opposite to theta 1 and rho 2 is the negative value of rho 1.
While we could use either to figure the line, if we have more than one line in our image then we'd have extra (and unnecessary) work to match up the correct pairs. In this case, the easiest thing to do is *not* to plot negative values of rho on our rho, theta image.

The problem with thick lines



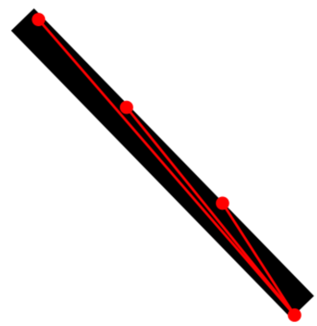
So far I’ve been talking about a line that’s only one pixel thick, while the spectra on our images are the same width as the stars. If, for example, the line was three pixels thick, <click>we could draw a line here, <click>here, or <click>here.

The problem with thick lines



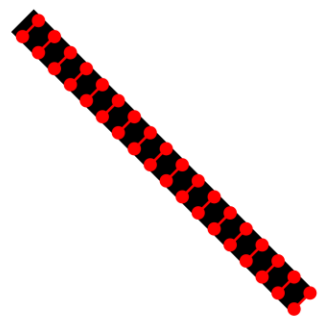
Here’s some more lines we could draw.

The problem with thick lines



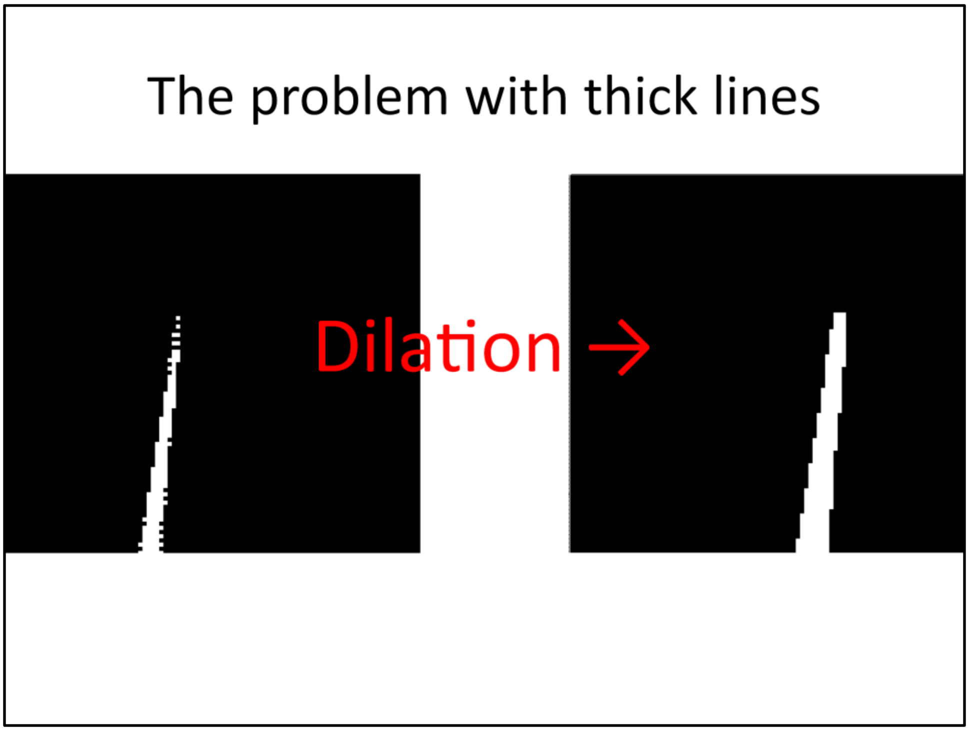
Yet some more.

The problem with thick lines

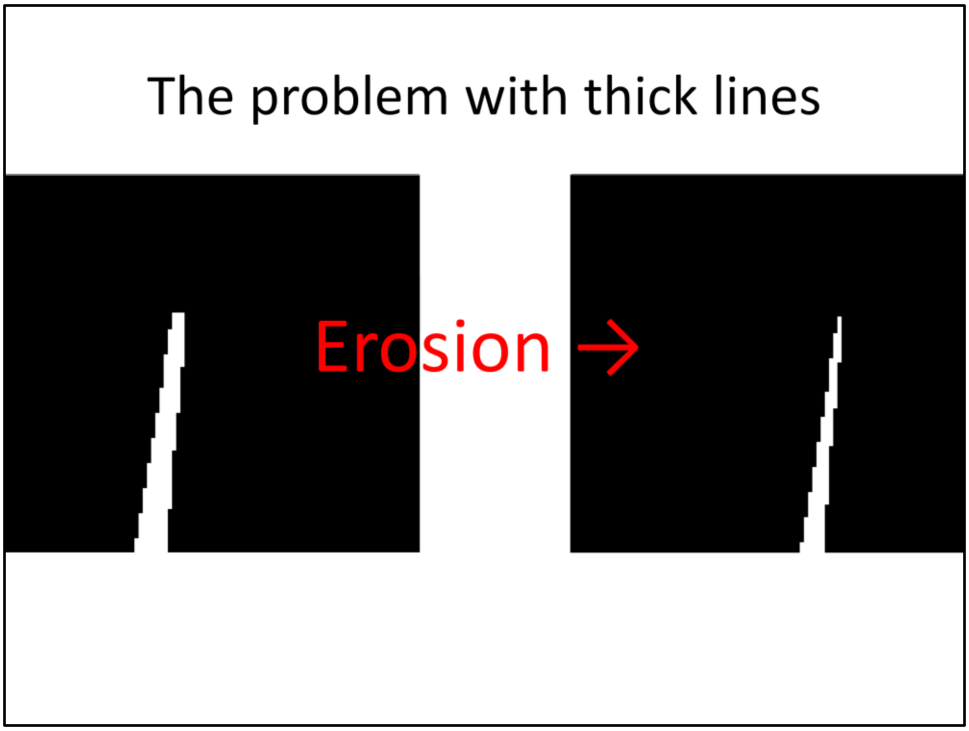


And even more. All of these could be found as peaks in the rho, theta image and make it difficult to find the ‘actual’ line.

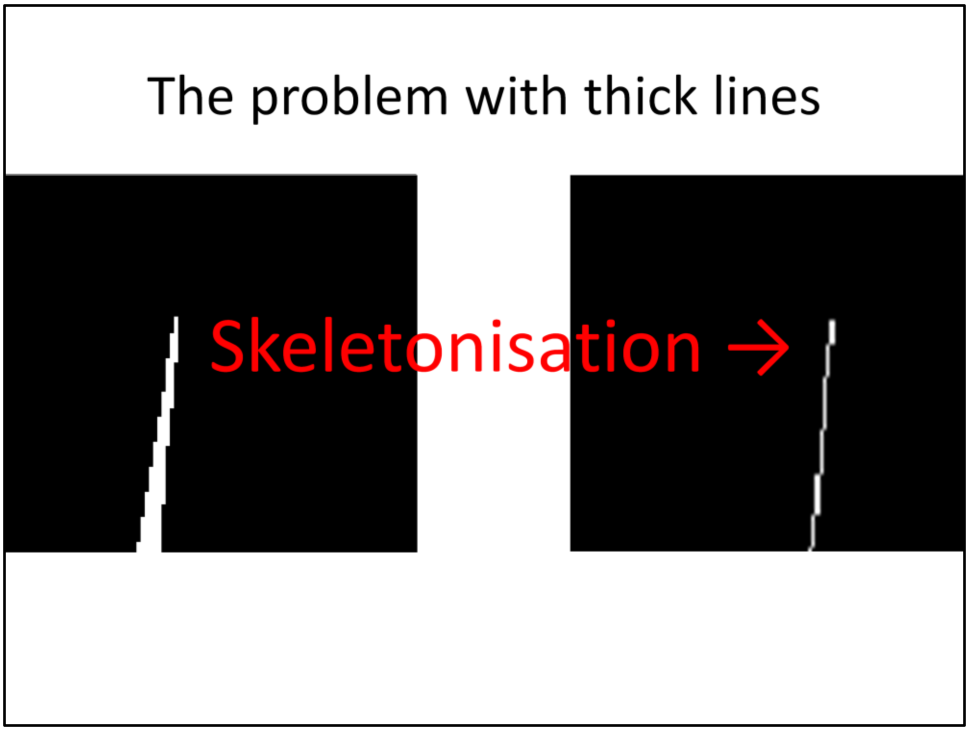
So what we need to do is somehow thin the spectra down to a line that's only a single pixel wide.
The first step is to convert the image, that for my video camera can have value between zero to 255, to a black and white image where zero is black and one is white. This is done by picking some value in the original image where everything below that threshold is black and above it is white. Tangra chooses this threshold based on the peak intensity of the star that you've manually selected and I expect I'll do the same. The image here is the result of converting the image of Sirius and its spectra that I had earlier in the talk to a black and white image.
<click>Now let's have a closer look at the part of the spectra inside the red box.
<click>Notice that there are some gaps near the end of the line and along the edges.



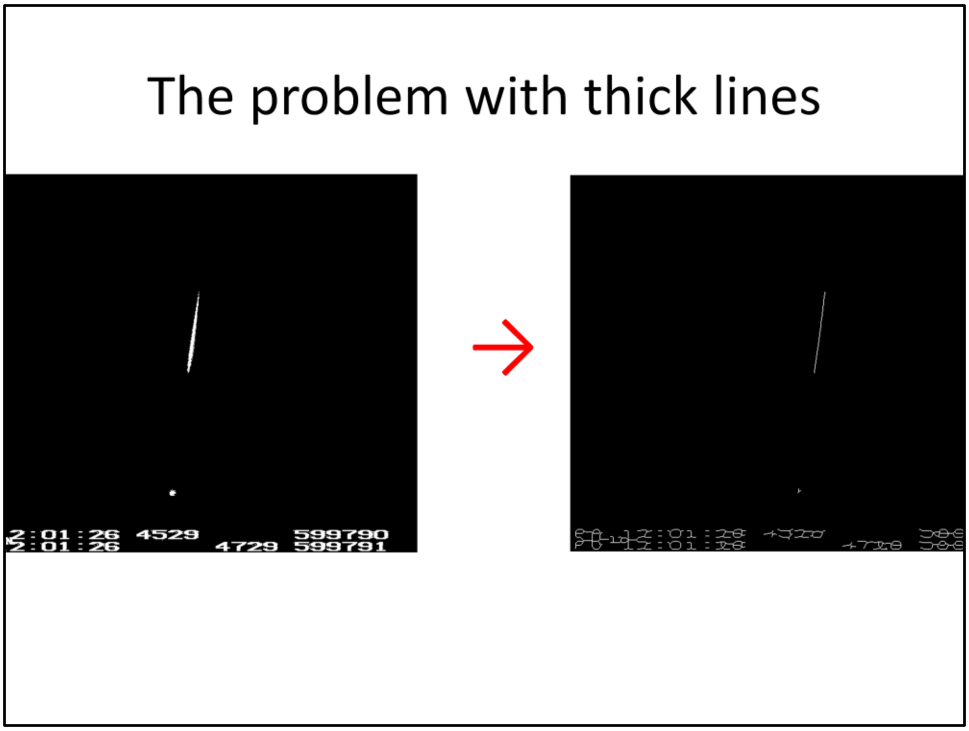
We can fill in these gaps by applying by applying a filter over every pixel in the image that carries out the image processing's dilation operation. This also makes the entire line a pixel thicker on each edge, so …



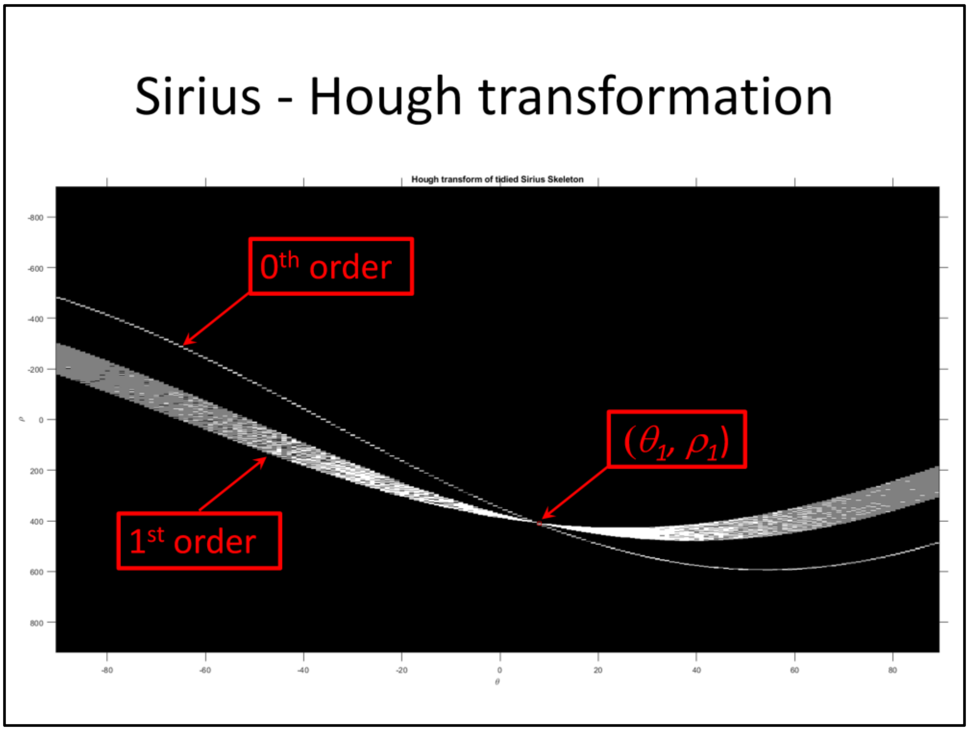
To undo that we apply another image processing operation know as erosion. We now have a clean looking line, even though it's still several pixels wide.



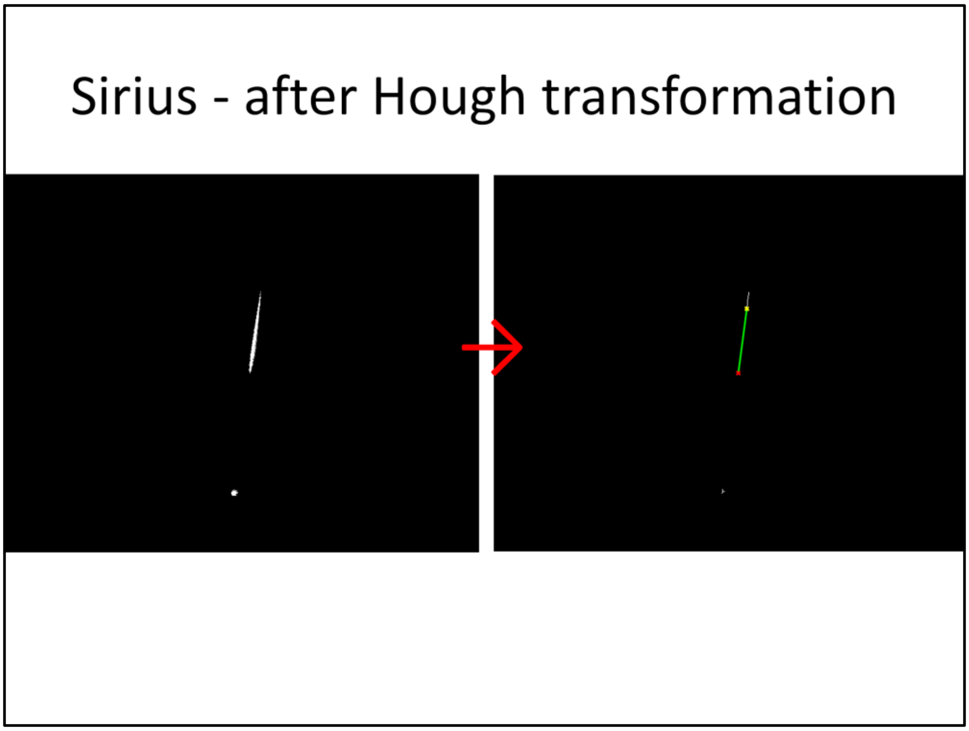
So the final image processing operation is applied, known as Skeletonisation. This has the effect of narrowing the line down to one pixel's width.



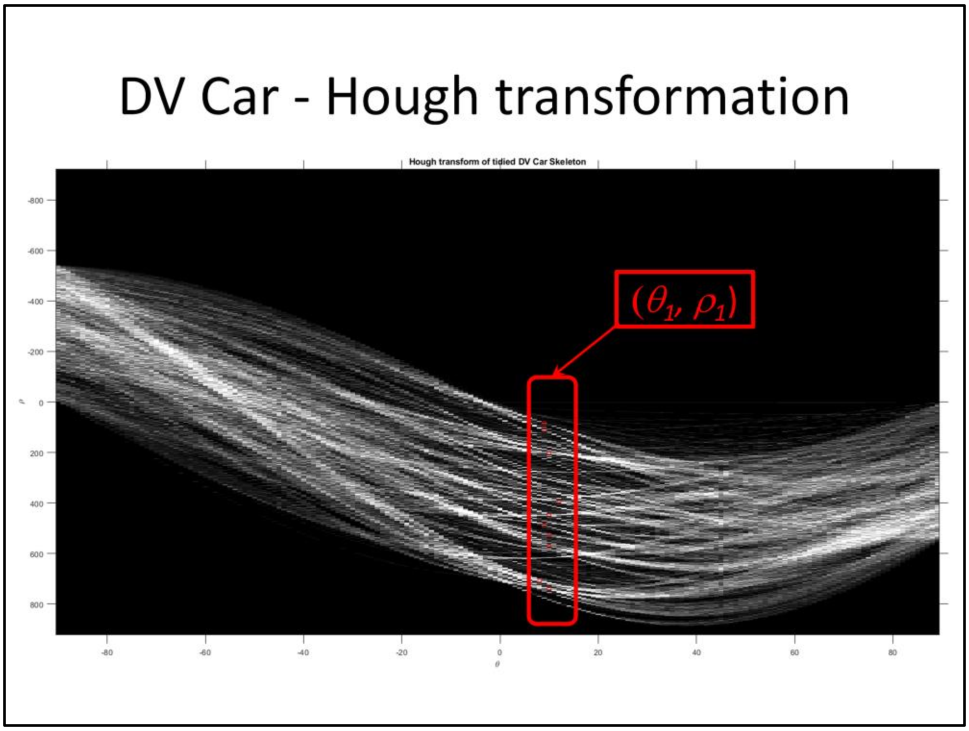
Zooming back out to the full image, I have before and after images. We now have the desired single pixel width line. You can also see the effect of the preceding operations on the time stamp at the bottom of the frame. As this could also cause spurious line detection in the Hough transform procedure I intend to simply black out the OCR region of the image before doing any of this image processing.



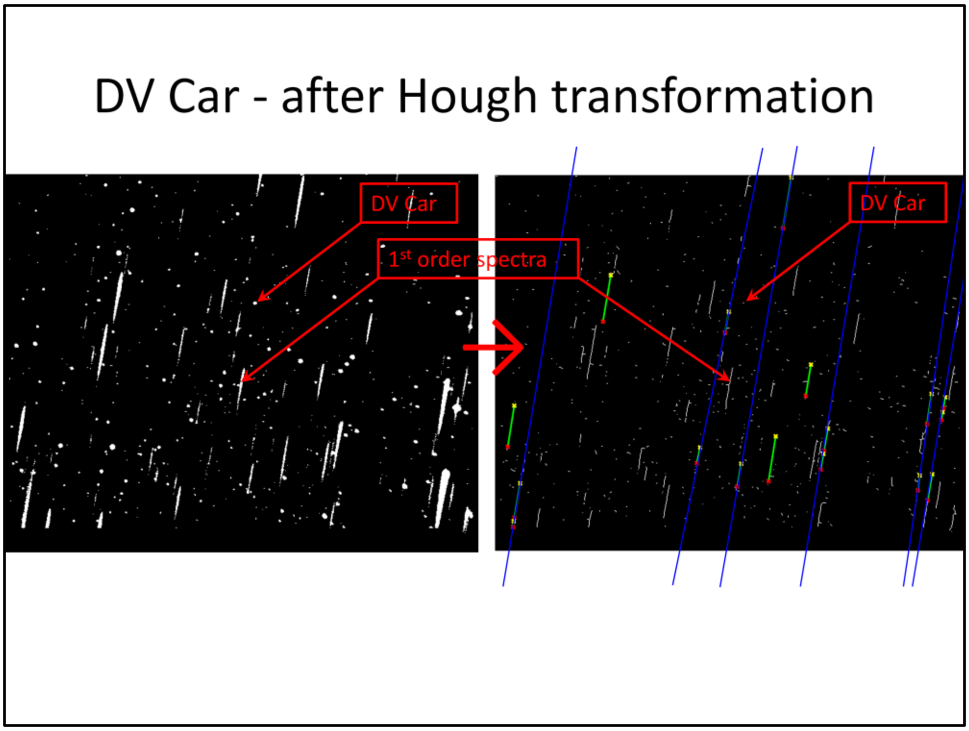
This graph shows the result of doing the complete procedure on Sirius, i.e. removing the OCR region of the original image, doing the image processing dilation, erosion and skeletonisation steps and finally performing the Hough transformation. The band of sine wave curves is due to the first order spectra line and the single curve is from the zeroth order, i.e. the star’s image.



This is another before and after slide, where the green overlay on the RHS image visually shows the line detected by the Hough transformation in the previous slide. So Tangra's existing code and my proposed procedure give the same (good) results in the simple situation of a single bright star and its spectra.



But the proof of the pudding is for the case where Tangra failed – that of multiple stars and spectra in the image, as I had for DV Carina.
This graph shows the result of doing the complete procedure on DV Carina. Here the procedure was arbitrarily restricted to finding the 10 'strongest' lines, which will tend to be the longest lines, which in turn will depend on both the star's brightness and spectral type. Notice they're all clustered around the same angle theta. As the spectral lines are expected to be parallel, I might be able to improve the accuracy of the procedure by taking the median of the 10 angles (τηετα). The median will be used, rather than the average, as it is less sensitive to outliers, i.e. spurious line detections.



Using the results from the previous slide, I can indicate the lines found with the green overlay as before.
<click>Eagle eyed observers may have noticed there are more than 10 line segments – this is because some of the spectral lines spuriously appear to the computer to be part of the same line as shown by the blue overlays. This also explains why the lines aren't all exactly parallel.
In this case DV Carina's spectra is not among those found. Nevertheless, as Tangra already knows the image coordinates of DV Carina (that you manually selected it earlier) then all Tangra now needs to know is the slope m (which is directly related to theta) in order to find the right spectra.

Questions?