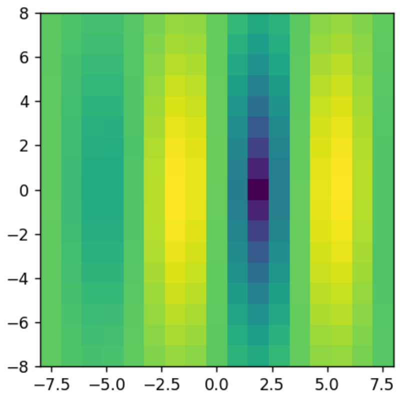
Question 1

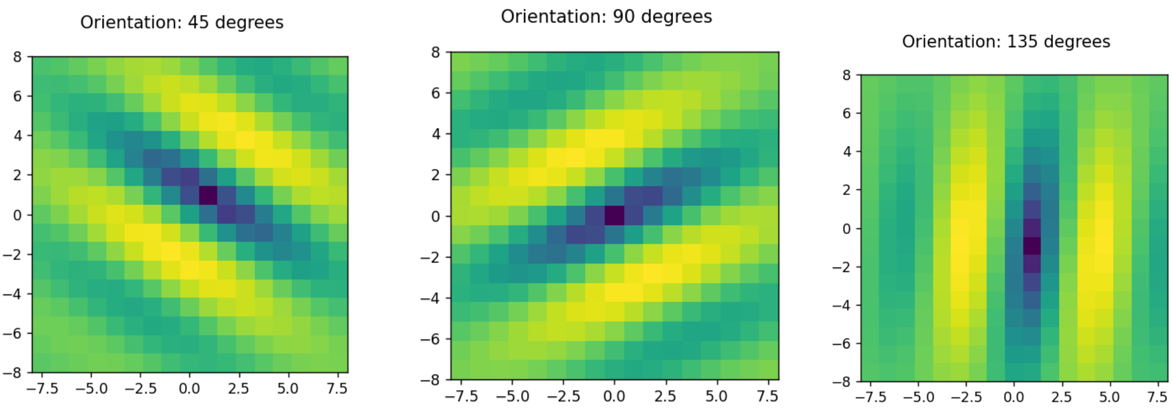
Part a

Plot the mean responses of the cells as a function of image velocities (vx, vy). The responses will form a 17x17 matrix which you should plot as a 2D image. Briefly discuss the relationship between these mean responses and the motion constraint equation that is defined by the normal velocity (2,0)



The response of the cells indeed shows a minimum when the image velocity corresponds to the target velocity (2,0). This is because at this velocity the shift in time corresponds to the exact shift in space at which the cell is tuned. Cells tuned to a normal velocity of (2,0) are optimized to detect motion exclusively in the x-direction at a rate of 2 units per time interval. When the observed image movement aligns with this expected velocity, the cell's response is minimized, indicating an exact match between the anticipated and actual motion in the image.

Part b

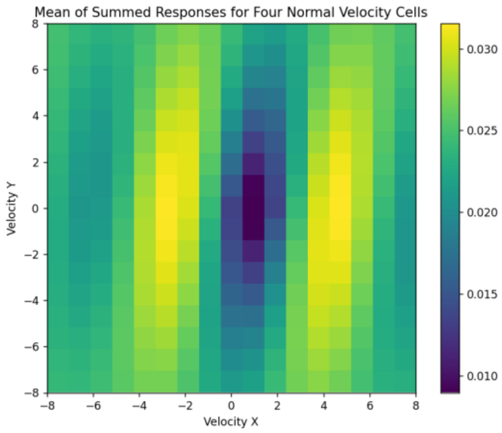


The minimal responses in the heatmaps for cells at 45, 90, and 135 degrees arise when image motion aligns with each cell's normal velocity, which is consistent with the motion constraint equation. In the case of the target velocity we adjusted the normal velocity at which each cell is tuned to be s.t. the target velocity will be on the perpendicular line to the normal velocity. This would mean that since the cell reports minimal responses along that line it should also report small responses for the target velocity (2,0). This works in theory, unfortunately it seems that these cells don't exactly follow the theoretical pattern, and while they report minimal activation when the image velocity is the same as the normal velocity they don't follow the perpendicular pattern perfectly. Still the response of these cells at the target velocity is pretty close to 0.

A solution to this might be changing the dimension of the Gabor or the frequency to account less for the variations related to the noise. When tried this leads to slightly better results but not perfect.

Part c

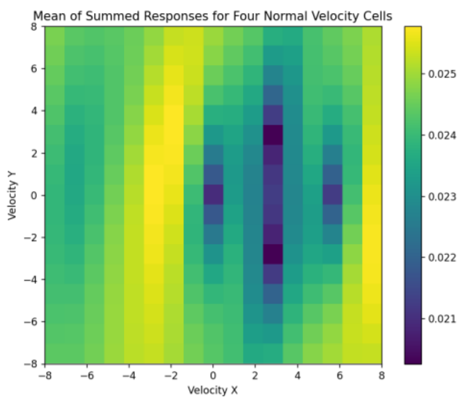
We can now try summing these outputs together to get a higher order cell



The minimum of this higher order cell is at (1,0) 1 square away from the intended target velocity, this is due to the inconsitencies in the previous part. However the cell still has a very low reponse near the target velocity of (2,0). If we were to iron out the problems in the previous point, by changing frequencies or perhaps with an image that is not noise the higher order cell should work as intended.

Part d

To adapt the higher order cell to higher target velocities we would in practice just have to change the normal velocities of each of the 4 cells accordingly respecting the motion equation. By theory the higher order cell should then have a near zero minimum at the intended target velocity. For example this is my result for a (6,0) tuned higher order cell



We can see that the cell doesn't exactly work as intended, it reports a near zero value near the target velocity of (6,0) but also other minimum values in 3 other points.

Two are the reasons for this behaviour:

1 the inconsistencies that presented themeselves in the previous points have a worse effect the further we move from the origin as the behaviour of the gabor is less clear.

2. Most importantly the response of each singular velocity tuned cell presents valley and peaks, when distancing ourselves further we incur in minimum points that are results of valleys of the cells that are not the central intended ones. To fix this we could increase the size of the gabor based on the magnitude of the target velocity and again change the frequency of the gabor to present less valleys and peaks.

Question 2

*Describe how your **Q2** function works.*

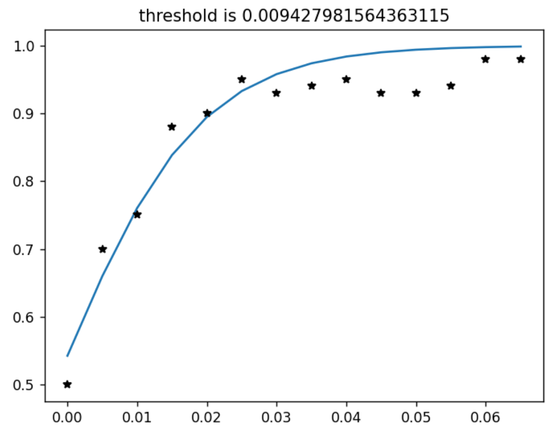
The Q2 function works by first calculating the results of a convolution (with mode "valid") between an horizontal gabor filter and the output image of the run_glass function storing it in an array of horizontal responses. It also does the same for the response of a vertical gabor filter.

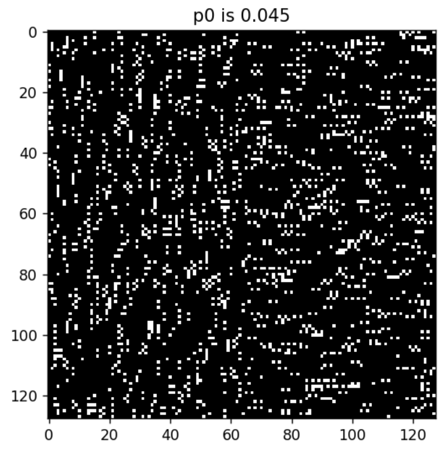
Now once these two arrays are sored the function then needs to compute whether its more likely that a lr or tb split occurred. To do so we assume that the value of the responses will be highest when the correctly oriented gabor filter is applied in the correct side following the correct split. We calculate two scores one being the sum of the responses in the case of an lr split and the other being the sum of the responses in the case of the tb split, we assume that the split leading to a higher score is the correct one.

To calculate these scores for both lr and tb we compute the sum of the vertical responses on one side (lets say left) and horizontal responses on the other. We also compute the opposite situation in which we apply the horizontal filter to one side (left again) and horizontal to the other. We then compute the max between the two sums and assume that is the score for the split (lr or tb).

*Submit the plot that is generated by **runGlass** for your selected parameters. The plot should show the data (fraction correct) as well as the logistic function curve fit. Mention in your text what are the p0 values used; these values should also be evident from your psychometric curve plot.*

The p0 values used are the list from 0.005 to 0.065 with steps of 0.005 as this was the result that best showed the initial part of the curve, the num_trials is set to 100 to get more accurate results. After 0.07 the curve stabilizes to a fraction correct value of 1

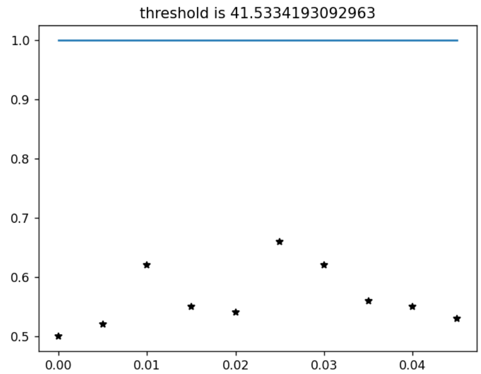


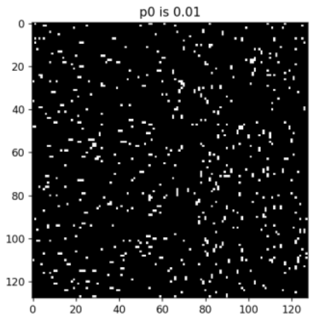


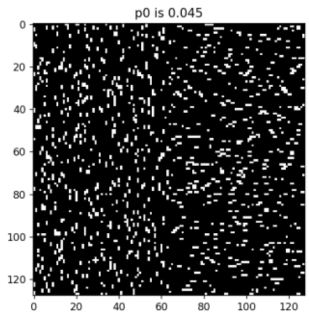
Change the experiment by making the dot spacing either closer or further apart.

1 pixel only:

This is a plot for a shift of 1 pixel. We use p0 values up to 0.045 since the function isn't able to reach a satisfactory fraction correct, I assume this is because a shift of 1 is too small for the selected gabor that we are using with 4 cycles in 32 pixels and therefore is not correctly identified. If we were to increase the frequency, say 16 cycles in 32 pixels we would indeed be able to reach a value of 1 in fraction correct.



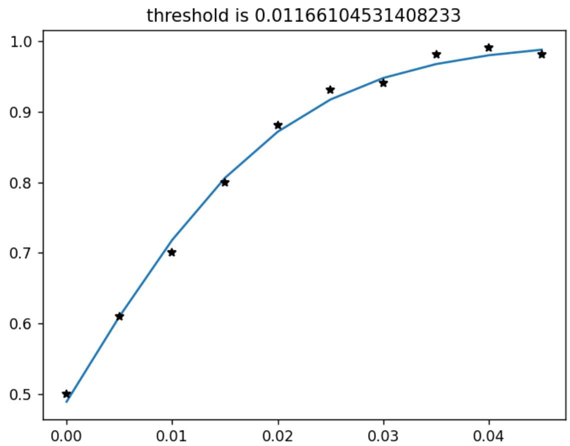


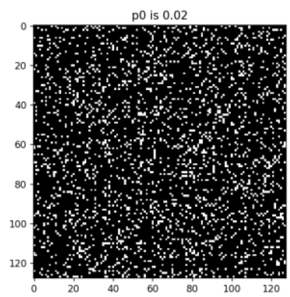


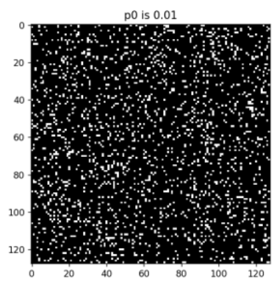
4 pixels:

Note: we increase the value of pnoise to 0.1 as otherwise the value of frac correct goes to 1 too quickly to fit a curve nicely.

This is because a shift of 4 pixels is larger and more easily recognizable by our gabor filter. If we were to increase the shift to 8 however this might be too much for the frequency of our gabor filter and it wouldn't be detected by the convolution.

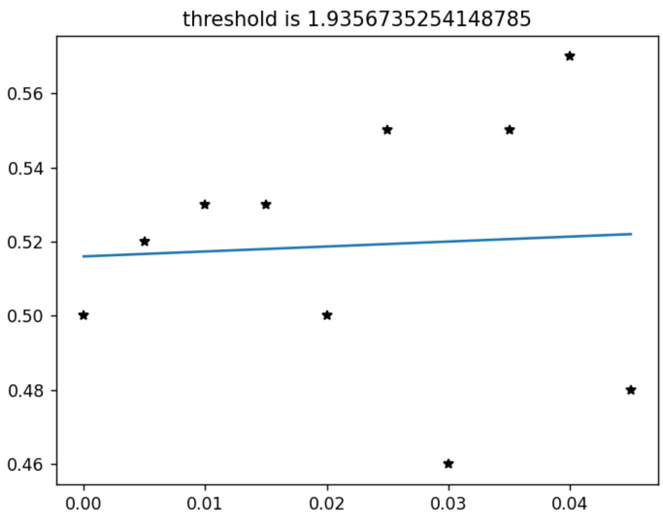


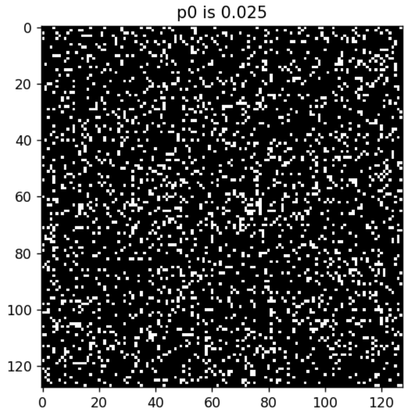




8 pixels:

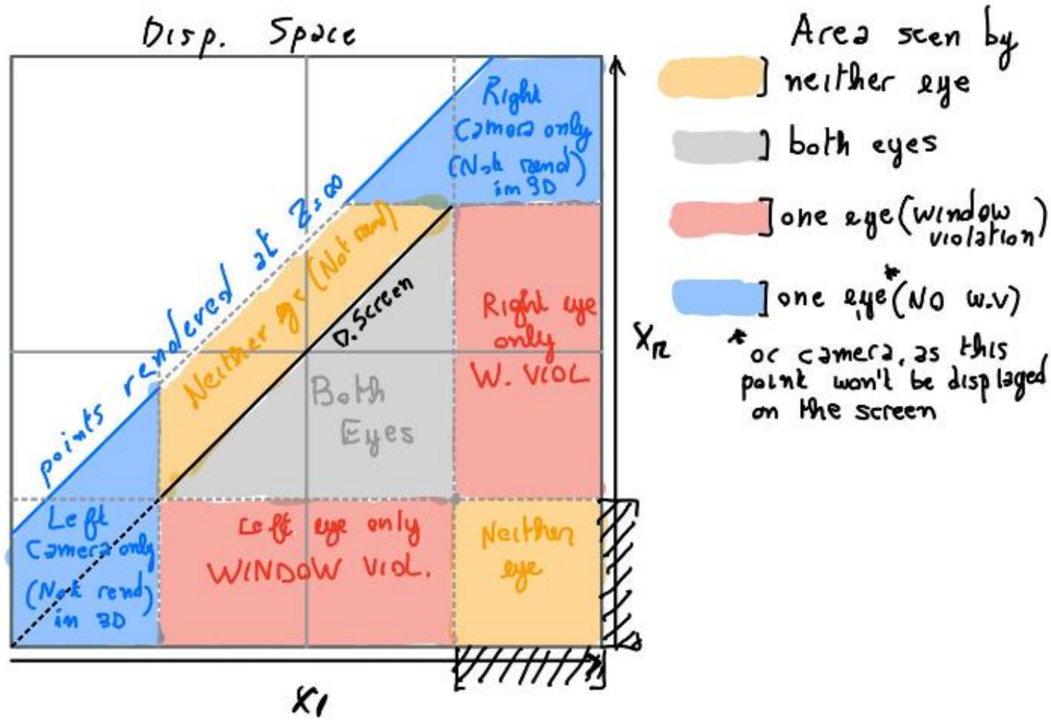
For 8 pixels no matter the noise values and p0 values selected the function is not able to reach a fraction correct of 1. This is probably due to the shift being too large for our specific gabor function to respond to. The plot in this document refers to a noise value of 0.01 and p0 values from 0.005 to 0.045



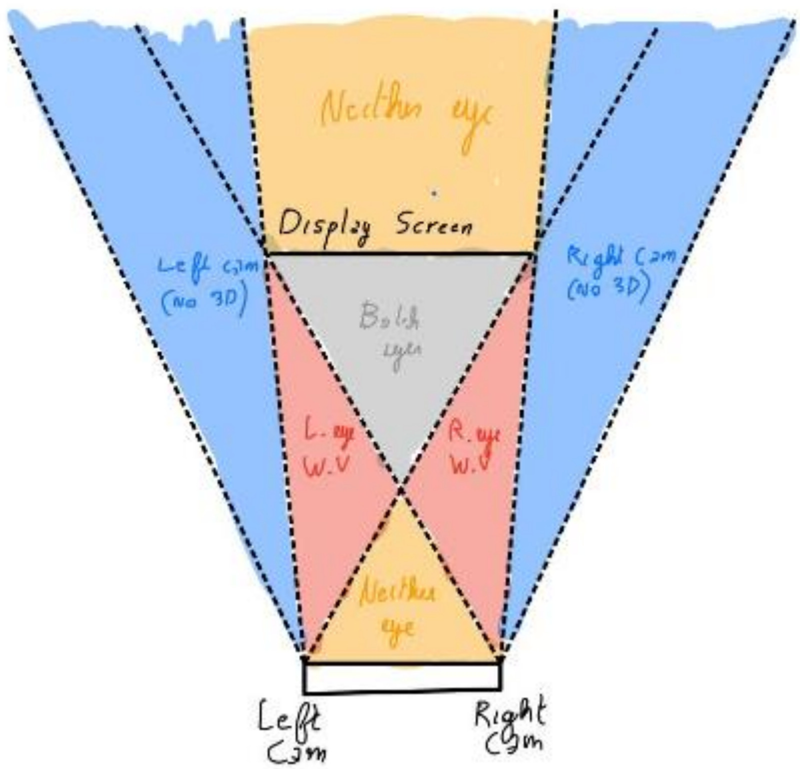


Question 3:

This is the disparity space drawing



This is the original picture with updated colors following the legend


Neither eye
Display Screen
Left cam (No 3D)
Right cam (No 3D)
Both eyes
L. eye W.V
R. eye W.V
Neither eye
Left Cam
Right Cam