COMP 546 Assignment 2 Winter 2019

Prepared by Prof. Michael Langer

Posted: Sunday, Feb. 10, 2019
Due: Sunday, Feb. 24, 2019 at 23:59.

The same general instructions apply as in Assignment 1, including the late policy.

This assignment will give you some hands on experience with using Gabor functions to measure image shifts which is fundamental to problems of binocular disparity estimation and monocular motion estimation.

Question 1 (50 points)

For this question, you will need to create a 1D sine Gabor and a 1D cosine Gabor which are the products of a 1D sine or cosine function and a 1D Gaussian. Let the vector length of the Gabors be `N` = 32 and let the number of cycles be $k = 2$. Let the standard deviation σ of the Gaussian be one third the wavelength, $\sigma = \frac{N}{3k}$. Define the origin of the Gabor to be at the middle indices of x and y, e.g. $\frac{N}{2}$.

(a)

In Assignment 1 Question 3c, you plotted the RMS responses of the convolution of a DOG with sine functions of different wavelengths. Do the same here for your two Gabor functions, namely convolve them with sine functions of different wavelengths and plot the RMS values.

Which *wavelength* of the signal sine function gives the biggest RMS value? How is this wavelength related to the *wavelength* of the Gabor. For your answer, express the wavelength in pixels, that is, pixels per cycle.

(b)

Generate a 1D signal using `randn(M,1)` which gives independent samples from a standard normal (Gaussian) distribution, that is, mean 0 and variance 1. Think of this 1D signal as a row from a noisy texture image. It doesn't matter that values can be positive or negative, as you can just think of these as offsets from a mean. You should make `M` much bigger than the Gabor widths `N`.

Convolve (or cross-correlate) the sine and cosine Gabors with this 1D signal. This yields two functions which we call L_c and L_s, which stand for 'cosine' and 'sine' Gabor responses. *Plot these two functions, as well as the response function* $\sqrt{{L_c}^2 + {L_s}^2}$. Briefly discuss how the values of these three response functions vary with position, in particular, how these spatial variations depend on the wavelength of the Gabor.

(c)

In the previous question you generated a 1D signal and filtered it with a cosine and sine Gabor. In this question, we will consider two signals, one shifted with respect to the other. We refer to the responses of the Gabors to the first signal as L_{c1}, L_{s1}. A second signal is defined by circularly shifting the first signal by `d` samples. (See Matlab `circshift.`) The responses from filtering the second signal with sine and cosine Gabors are just shifted versions of the responses from the first signal. So, for each shift `d`, the vector L_{c2} is just the shifted version of L_{c1} (except near the boundary) and the vector L_{s2} is the shifted version of L_{s1}. We leave out the subscript `d` on the L_{c2} and L_{s2} but note that there is one of each of these functions for each `d` and they are just shifted versions of each other.

Given the linear response vectors L_{c1}, L_{s1} and the vectors L_{c2} , L_{s2} for each signal shift `d`, compute the following non-linear response functions which will depend on shift `d`:

- $r_{diff} = \sqrt{(L_{c1} - L_{c2})^2 + (L_{s1} - L_{s2})^2}$
- $r_{sum} = \sqrt{(L_{c1} + L_{c2})^2 + (L_{s1} + L_{s2})^2}$

The definition of the response function r_{diff} is similar to the 2D one given in the lecture slides, namely it measures *differences* in the two filtered signals. The response function r_{sum} measures *sums* rather than differences.

For both r functions, plot the mean value r as a function of the shift `d`. The values of the shifts `d` should range from `–N` to `N` where `N=32` is the Gabor window width.

You should get a peak mean response (either maximum or minimum) for `d=0;` in this sense, these r functions can be considered *tuned to zero disparity*. *Briefly discuss if either of these disparity-tuned models seem more sharply tuned than the other* by comparing the sharpness of the peak of the mean response function near `d=0`.

If you have computed the functions correctly, then your responses should also have side lobes (small oscillating tails). *Briefly discuss why these side lobes occur. The reason is rather subtle so think it and choose your wording carefully.*

As an aside, note that the r_{diff} is much more commonly used in computer vision approaches and r_{sum} is much more commonly used in biological vision modelling.

Question 2 (50 points)

This question considers *motion* sensitive cells defined by 2D sine and cosine Gabors. You should use the `make2DGabor.m` function in the starter code to make these Gabor cells.

There are a few "heads up's" for this question. One is that in Matlab you should use (y,x) rather than (x,y) as indices into your image since Matlab uses (row, column) notation. This issue also comes up with the `circshift` function which uses (row,column) indexing. Also, the rows in Matlab start from top of the image, not from bottom.

A second heads up is that you will be dealing with both shifts of the image (the motion) as well as shifts of the cell receptive fields (normal velocity tuning). You need to keep track of which is which.

Ok, let's get started…

Let the motion sensitive cell's response depend on two images. Assume that the second image is shifted by a fixed amount:

$$(\Delta x, \Delta y) = (4,0)$$

relative to the first. The shift can be thought of as the velocity (vx, vy) pixels per frame. I emphasize that this is the motion of the image intensities themselves, not the receptive fields of the cells.

For the cells, assume the shift vector of their receptive fields from the first to second frame is normal to the cell's preferred spatial orientation. We will consider three spatial orientations of cells (0, 45, 90 degrees) and range of shifts (normal velocities) for each of the three orientations. Details are given below.

To define the Gabors, use the same values of N and σ as you used in Question 1, except that now the Gabors are 2D rather than 1D.

You will need to vary the spatial frequencies k_x, k_y to get different preferred orientations. The (k_x, k_y) vector is perpendicular to the spatial orientation of the cell and parallel to the preferred shift vector. We require the (k_x, k_y) vectors to have the same overall spatial frequency. We set $k = 2$ to be constant, and choose (k_x, k_y) accordingly so that $k = \sqrt{{k_x}^2 + {k_y}^2}$.

The three cell family orientations as listed as follows, and for each orientation we have a range of shift vectors which are defined by `d` in -N to N.

- vertically oriented cells; they have preferred spatial orientations in directions (0,1) and the shifts of the receptive fields are $(\Delta x, \Delta y)$ = (`d`, 0) pixels.

- horizontally oriented cells; they have preferred spatial orientations in direction (1,0) and the shifts of the receptive fields are $(\Delta x, \Delta y)$ = (0, `d`) pixels.

- right diagonal cells; they have preferred spatial orientations in direction (1,1) and the shifts of the receptive fields are $(\Delta x, \Delta y)$ = (`d`, -`d`). We will not consider left diagonal cells because, for reasons of symmetry, one obtains the same plots for left and right diagonals.

For each of the three orientations and for each of the shifts defined by values of `d`*, compute the mean responses using the* r_{sum} *definition from Question 1.* Note that we are considering the responses to a single image motion only, namely the actual image shift (4,0).

To define the responses, we also need to specify the image intensities. We consider two cases:

- a random 2D Gaussian noise image which you can generate using `randn(M,M);` define this noise for the first frame and then generate the second frame by circular shifting the image by $(\Delta x, \Delta y) = (4{,}0)$; be sure to make `M` >> `N`, that is, the image is much larger than the Gabor.

- a 2D image that for each x has a random Gaussian value which is constant over y; so the image consists of vertical lines with random grey level; again use a large image and generate the second frame from the first by circular shifting.

For both types of intensity pattern, and for each cell orientation, *plot the response values as a function of* `d`.

Briefly explain

- *why the peak response occurs at that particular value of* `d` *in each plot*
- *why the values of the peak responses differ between plots*
- *why side lobes occur.*

In particular, note that the random line image produces an "aperture problem" since the gradients are always in the horizontal direction. *Explain what this problem is, in particular, for the vertical line image. Your explanation should be in terms of using the responses of the cells to infer the visual systems task of inferring the underlying image velocity* (vx, vy).

Finally, I am expecting you to think about how one could build a "higher level" cell that is tuned to a specific shift (vx, vy). This higher level cell is more complicated than the normal velocity cells that you have built. A higher level cell that is tuned to a specific (vx, vy) would need to combine the responses of normal velocity cells. The idea was sketched at the end of lecture 8, when I briefly discussed area MT of the brain.

Good luck! Have fun!