Last lecture we discussed simple cells in V1. We considered sine and cosine Gabor models for such cells. The linear response of these cells to an image was defined by the inner product of a Gabor function with that image.

One technical point: As we saw with retinal ganglion cells, simple cell cannot have a negative response (real cells cannot have a negative number of spikes per second), and so we need two versions of each cell where the weights of one version are just the negative of the other, and so the two versions of each cell have the same linear response magnitude but opposite sign. In the model, the two responses are half wave rectified so one is positive and the other becomes 0. This allows the model effectively to represent both positive and negative responses of the linear cell.

Complex cells

Hubel and Wiesel found a second class of cells in V1 that are also sensitive to oriented intensity patterns (lines and edges), but these cells were quite different from the simple cells. Whereas a simple cell has a well defined excitatory and inhibitory region, this second class of cell does not. These cells are not sensitive to the precise position of the oriented pattern (edge, line) within the cell's receptive field. (Many of these cells are sensitive to motion of an oriented pattern as well. I will discuss this in an upcoming lecture.) Hubel and Wiesel called this second class of cells *complex cells*.

There are many ways one can model a complex cell. One way is to take a set of simple cells that have the same orientation and are distributed over a range of shifted positions. The responses of each simple cell are half wave rectified, and the complex cell could be defined by taking the sum of the these half wave rectified values. The complex cell's receptive field would be the union of the receptive fields of the simple cells that it reads from. It would respond to lines or edges at various positions within that receptive field, but you couldn't say that one position was excitatory or inhibitory. I I sketched out such an example cell in the slides (model 1).

The second and third models that I mentioned in class is defined by a sine Gabor and cosine Gabor pair that have the same frequency (k_0, k_1) and envelope size σ and are both centered at (x_0, y_0). That is, the receptive fields now coincide. The linear response of the sine and cosine Gabors are defined by the inner products of each with the image, and form a pair:

$$(< cosGabor(x - x_0, y - y_0), I(x, y) >, \ \ < sinGabor(x - x_0, y - y_0), I(x, y) >)$$

Think of the pair as a vector in a 2D space. Model 2 and 3 differ in what they do with these two responses.

Model 2 defines the complex cell response as follows:

$$| \ < cosGabor(x - x_0, y - y_0), I(x, y) > \ | + | \ < sinGabor(x - x_0, y - y_0), I(x, y) > \ |$$

that is, the sum of the absolute values. This computation was illustrated in the slides in a slightly different way, namely by taking the sum of two pairs of sine and cosine Gabors, which are each *half wave* rectified. Mathematically, we have the following (for the cosine Gabor). Letting $[\ ]_+$ be the half-wave rectification operator, we write:

$$\begin{aligned} & | \ < cosGabor(x - x_0, y - y_0), I(x, y) > \ | \\ = & \ [\ < cosGabor(x - x_0, y - y_0), I(x, y) > \ ]_+ \ + \ [\ < -cosGabor(x - x_0, y - y_0), I(x, y) > \ ]_+ \end{aligned}$$

and similarly for the sine Gabor.

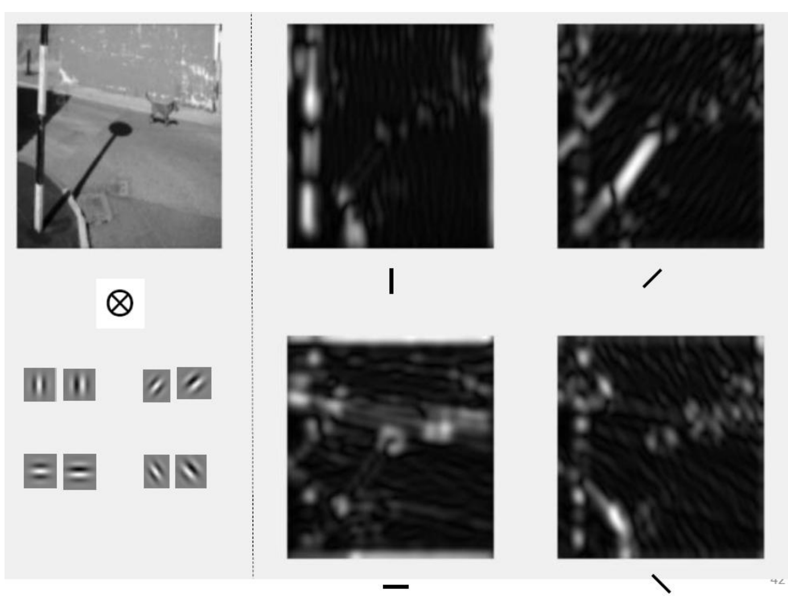
Model 3 is similar except that we take the squared values of the linear cosine and sine Gabor responses, rather than the absolute values. Taking the square value might seem to buy us nothing at first glance, but in fact it does make the math a bit cleaner which hopefully you'll appreciate soon. The basic idea is to treat the linear responses of the cosine and sine Gabor as a 2D vector (we have a pair of values), and to consider the Euclidean length of this vector:

$$\sqrt{< cosGabor(x-x_0, y-y_0), I(x,y) >^2 + < sinGabor(x-x_0, y-y_0), I(x,y) >^2}.$$

This is the third model of the complex cell's response, and it is the most commonly used model.

Example

Consider again the example image from the previous lecture. Now we take the responses of both the sine and cosine Gabors of each orientation, and we compute the complex cell responses at each pixel. We compute eight maps – four orientations of a sine Gabor and four orientations of a cosine Gabor – and we cross-correlate each of the Gabors with the image. For each of the four orientations and for each image position, we define a complex cell response, which is what the images show. The four images below right now represent non-negative values only, so zero response is black. Roughly the same regions as last lecture give a large response. For example, the vertically oriented complex cells gives good responses on the pole, and the right diagonal oriented cells gives a good response on the cast shadow. The main difference between the responses below and what we saw last lecture with the simple cells is that the position information in the complex responses is less detailed. This is exactly what complex cells encode: they encode that there is some oriented structure in a local neighborhood but they don't indicate exactly where.



Estimating binocular disparity

Recall that the left and right eye images are send to the LGN but the signals are not combined there. The left LGN carries the signals from the right visual field and the right LGN carries signals from the left visual field. However, the left and right eye's signals for each field are fed into different LGN layers and then relayed separately to V1 where they provide the inputs to binocular simple and complex cells. I will not discuss binocular simple cells. Instead I will discuss just binocular complex cells. Before I do, let's consider what computational problem these cells are solving.

Consider a visual direction (x_0, y_0) in retinal coordinates, that is, relative to each eye's coordinate system. Suppose that the left and right images near this direction have similar intensities, except for a horizontal shift (see analglyph slide), that is, a binocular disparity. This shift might vary over the image, because the depths will vary and the shift depends on depth.

[EDITED: March 9] Near (x_0, y_0), the visual system could attempt to estimate the shift to be the value d that minimizes

$$\sum_{(x,y)\in\mathcal{N}(x_0,y_0)} (I_{left}(x+d,y) - I_{right}(x,y))^2$$

where the sum is over (x, y) coordinates in a neighborhood of (x_0, y_0). Note that $d > 0$ corresponds to a *leftward* shift of I_{left}. For the correct d, this would remove the disparity between the left and right images so they would be properly registered and their point-to-point difference would be 0.

The idea of this computation is that if you shift the left image by the correct disparity d, then the shifted left image should correspond pixel-by-pixel to the right image – at least in the local patch where the disparity is roughly constant. In that case, the above sum of squared differences should be 0 for the correct d. For other values of d, sometimes the left image will be brighter at a pixel than the right image and sometimes it will be darker, so the intensity difference at that pixel will be non-zero. We square the intensity differences because we only case how much it is different from 0, not whether it is positive or negative.[1] The idea for estimating disparity d near (x_0, y_0) for a particular left-right image pair is to choose the d value that minimizes this sum of squared differences.

While the above computational model works well (and is the basis for many computer vision methods for binocular stereo, the model is not biologically plausible. In the brain, binocular disparity estimation occurs in V1 which analyzes images using Gabor-like cells. We next consider a model based on such cells. We restrict ourselves to vertical oriented cells. (In Assignment 2, you will explore why.)

Up to now we have considered monocular complex cells in V1 which were constructed rom simple cells. We now consider binocular complex cells which are constructed from simple cells, namely Gabor cells for the left and right eyes. Using a similar idea as the computer vision method above, we could estimate the disparity d by finding a d shift that minimizes the follwing sum of squared differences:

$$(< cosGabor(x - x_0 - d, y - y_0), I_{left}(x, y) > - < cosGabor(x - x_0, y - y_0), I_{right}(x, y))^2$$

$$+ (< sinGabor(x - x_0 - d, y - y_0), I_{left}(x, y) > - < sinGabor(x - x_0, y - y_0), I_{right}(x, y))^2$$

[1] We could have alteratively taken the absolute value, and indeed some computational models do that.

Here the d shift is for the sine and cosine Gabor for the left eye, that is, the Gabors for the left eye are centered at $(x_0 + d, y_0)$ and the Gabors for the right eye are centered at (x_0, y_0).

The idea is that if we place a cosine Gabor template at $(x_0 + d, y_0)$ in the left image and at (x_0, y_0) in the right image and if d is the true disparity in the images – then the linear responses of the left and right eye cosine Gabors will have the same value, so if we subtract one from the other then we get 0. Similarly, the linear responses of the sine Gabors will have the same value, so if we subtract one from the other we get 0. The shift d that minimizes the sum of squared differences in the above expression would the best estimate of the disparity.

The above model works well in theory. Unfortunately, it doesn't describe binocular complex cell responses in V1. Rather, complex cells in V1 that are tuned to a disparity d have a *maximum* response (not a minumum response) at that disparity. So we need to change the model slightly so that it has this property. We do so by summing rather than taking a difference:

$$(< cosGabor(x - x_0 - d, y - y_0), I_{left}(x, y) > + < cosGabor(x - x_0, y - y_0), I_{right}(x, y))^2$$

$$+ \; (< sinGabor(x - x_0 - d, y - y_0), I_{left}(x, y) > + < sinGabor(x - x_0, y - y_0), I_{right}(x, y))^2$$

The intuition here is that when the shifted distance d corresponds to the correct disparity then the two cos Gabor responses will be identical and the two sine Gabor responses will be identical, as above. Now, when we sum them and square them, rather than getting a perfect cancellation, we get a big response. Several models[2] along these lines were proposed in the 1990's. These cells have peak responses to images of some disparity d to which the cell is tuned, and they are sensitive to particular orientation (usually vertical), and they don't care about the specific position of the (vertially) oriented structures within their receptive fields, just like the monocular complex cells we discussed earlier.

In particular, note what the above model of a binocular complex cell predicts would happen if the visual system were shown only one image – for example, if one eye were closed. That eye would have $I = 0$, and so it would not contribute to the response and the model would predict that the cell behaves just as a monocular cell – namely for the image given to the other eye. For example, if $I_{left} = 0$ everywhere, then the response would be

$$\sqrt{(< cosGabor(x - x_0, y - y_0), I_{right}(x, y) >)^2 + \; (< sinGabor(x - x_0, y - y_0), I_{right}(x, y) >)^2}$$

In the slides, I give an example of a response of binocular complex cell that is "tuned" to zero disparity ($d = 0$). This cell has its four Gabor receptive fields (sine and cosine, left and right eye) centered at the same position (x_0, y_0). I showed the responses of this cell to a single vertical line in the left and right eyes as a function of the x position of the line. Three different plots showed the responses for three image disparity values namely 2, 10, and 18. You can generate those plots yourself by running the code in

`http://www.cim.mcgill.ca/~langer/546/MATLAB/complexCells.zip`

in particular, run `binocularComplexCell.m`. You will need to input a disparity value for the vertical line. You might also run the code `monocularComplexCell.m` . You don't need to inspect

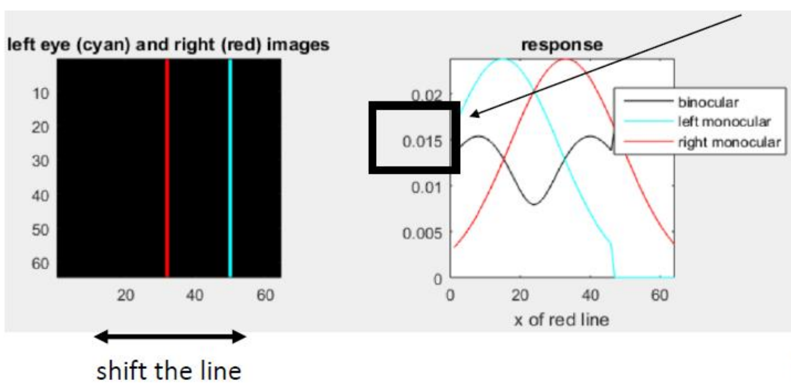
[2] e.g. Ohzawa, Freeman, DeAngelis, Qian, Fleet and others

what the code is doing; indeed I suggest you *don't* since it is implemented using Gabor formulas that look different (but mathematically equivalent!) from what I wrote above.

Below I show the responses for the case that the disparity between the left and right image is 18 pixels. The cell is tuned to zero disparity, so we would not expect the cell to have a good response to an 18 pixel disparity.

The arrow points to the value of the peak response value for this binocular complex cell, namely about 0.015. This is about one third the value of the response for the case of disparity = 2 pixels. (See the slides for that plot).

Also note that the peak response for the binocular cell shown for the 18 pixel disparity images is *less than* the peak monocular responses (green and cyan curves) to the same shifted line images. The monocular responses are what we would get if one eye were closed. To understand why the binocular response is so poor, note an image disparity of 18 pixels corresponds to roughly half a wavelength of the Gabor. (The Gabor's sinusoid is defined to have 2 cycles for 64 pixels, so half a wavelength is 16 pixels). So when the line's position in the left eye sits on a maximum of (say) of the sine Gabor, the shifted line in the other eye's image will fall on a minimum of that Gabor, and so there is a cancellation of values when the left and right sine Gabors are summed. (The cancellation may not be exact because the value of the Gaussian window will typically not be the same at the line and the shifted line in the other image.) The same argument can be made for the responses of the left and right cosine Gabor to the shifted line in the left and right image, namely the responses will be of opposite sign and will roughly cancel.



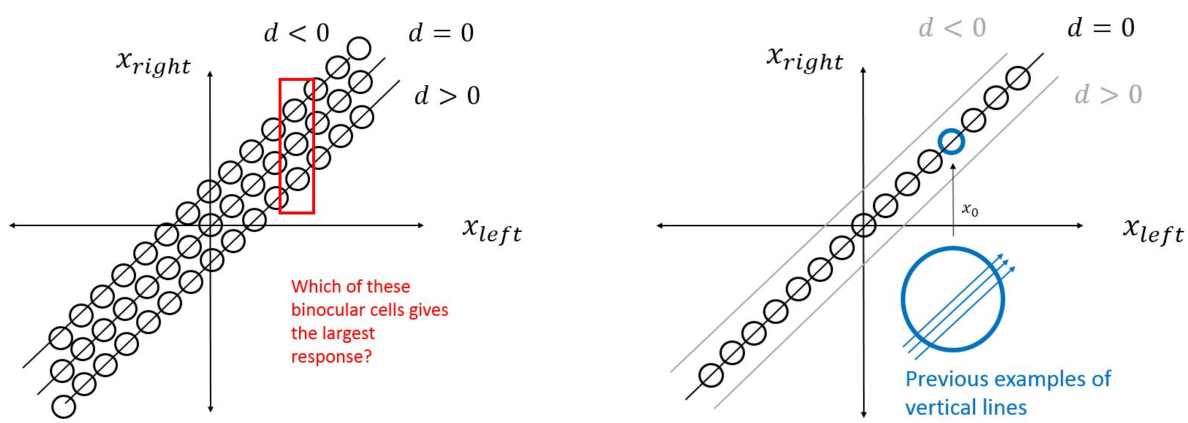
[I ended the class here. I will finish up the rest of this material next lecture, and then move on to motion processing in V1.]

Disparity Space

The above example showed how one binocular complex cell – which was defined to be tuned to disparity of 0 – responds as a function of two parameters: the disparity of the vertical image line, and the location of this line. Let's next consider a slightly different question: how does a family of binocular complex cells (say each with peak tuning to a *different* disparity) respond to a single stimulus? That is a very important question, since the visual system estimates the disparity at an image location (x_0, y_0) by comparing the responses of this family of cells and choosing the disparity the cell that gives the biggest response.

Let's not deal with a numerical example here. (I'll save that for Assignment 2.) Instead let's just sketch out conceptually what it means to have a family of cells that are tuned to different disparities.

The figure(s) below considers just a 1D case where the variable is x. Each binocular cell has two monocular receptive fields, centered at x_{left} and x_{right} and so we can indicate the binocular receptive field with a disk (or square, if you prefer – I use disk because I'm thinking of a Gaussian in each dimension and the product of two 1D Gaussians is circularly symmetric). If the monocular receptive field centers are at the same position in the two eyes, $x_{left} = x_{right}$, and then this cell would be tuned to a disparity of 0. This is just the case of the example above. If the monocular receptive field center for the left eye is to the right of the monocular receptive field center for the right eye, then this cell would be tuned to a positive disparity. Similarly, if the monocular receptive field center for the left eye is to the left of the monocular receptive field center for the right eye, then this cell would be tuned to a negative disparity. See the $d > 0$ and $d < 0$ zones in the plot below.



The idea for the figure on the left is that, for each x_{left} (say) the visual system "considers" the set of binocular complex cells whose left monocular receptive field is centered there. See cells highlighted in red. The best estimate of the disparity would correspond to that of the binocular cell in the (red) set that gave the largest response.

Finally, to explore this disparity space representation a bit more, consider again the vertical white line example. As the position of the line is shifted in both the left and right eyes, it's x value sweeps out a diagonal line in the disparity space. See blue arrows in the big disk. Three different line disparities are sketched there (say 2, 10, 18). Think of these as the three examples given in the slides.

We will return to these ideas again a few lectures from now, and in Assignment 2.

