Time varying images

Up to now we have said very little about how images vary over time. But of course they often do. Let’s think of an image as a function of $x, y$ and $t$, namely $I(x, y, t)$. In the first half of the lecture, I’ll define a general computational problem of estimating motion in an image, and how to solve it. In the second half of the lecture I will sketch a solution that is known to be used in the brain.

An example image motion, consider a vertical intensity edge drifting to the right over time. The figure below shows a small space-time cube through which the edge passes, and it shows an XY slice and an XT slice through the cube. This edge drifts to the right with speed $v_x$ so $v_x$ is the slope of the edge in the XT slice (where slope is measured $\frac{dx}{dt}$, not $\frac{dt}{dx}$). It could be that there is also a motion component in the $y$ direction. However, this component is impossible to measure for this example, since the image intensity does not vary in the $y$ direction.
See the lecture slides for two other examples of $I(x, y, t)$. One is just a moving bar instead of a moving edge. The second is more interesting and shows a real video of a person walking from left to right. An XT slice reveals the motion pattern of the person’s legs.

**Image motion constraint equation**

The computational problem we are trying to solve is to estimate the local velocity $(v_x, v_y)$ wherever we can, namely at whatever pixels $(x, y)$ and times $t$ where have intensity information that allows us to do so. The intensity information we’ll use are just the partial derivatives $\frac{\partial I}{\partial x}, \frac{\partial I}{\partial y}, \frac{\partial I}{\partial t}$. This turns out to be enough for a basic version of the problem. We do need intensity changes, since if intensity is constant in an XYT patch then there is no information about motion.

Suppose that the points in a small local patch have image velocity $(v_x, v_y)$, and I’ll say what that means below for ”points” to ”move”. For now, let’s not worry about the units, whether the space units are pixels (photoreceptors), mm on the retina, or visual angle and or whether the time units are seconds, or some frames in a video. We simply say that each point will translate a distance $(v_x \Delta t, v_y \Delta t)$ in a time interval $\Delta t$. Based on that, what can we say about how the image intensity changes over this time?

If the image intensities $I(x, y, t)$ are smooth enough that we can compute local derivatives, and write a Taylor series expansion of the intensities near $(x, y, t)$ as

$$I(x + v_x \Delta t, y + v_y \Delta t, t + \Delta t) = I(x, y, t) + \frac{\partial I}{\partial x} v_x \Delta t + \frac{\partial I}{\partial y} v_y \Delta t + \frac{\partial I}{\partial t} \Delta t + \text{H.O.T.}$$

where H.O.T. stands for ”higher order terms”, namely higher than first order derivatives. The partial derivatives are evaluated at $(x, y, t)$.

We now make an important assumption about the motion, namely that the image intensity of a moving point doesn’t change over time – this is sometimes called intensity conservation. Thus, when a point moves from $(x, y)$ to $(x + v_x \Delta t, y + v_y \Delta t)$ from time $t$ to time $t + \Delta t$, respectively, we have

$$I(x + v_x \Delta t, y + v_y \Delta t, t + \Delta t) = I(x, y, t).$$

This lets us cancel these two terms in the Taylor series above. If we further ignore the higher order terms, then we have:

$$\frac{\partial I}{\partial x} v_x \Delta t + \frac{\partial I}{\partial y} v_y \Delta t + \frac{\partial I}{\partial t} \Delta t = 0$$

or

$$\frac{\partial I}{\partial x} v_x + \frac{\partial I}{\partial y} v_y + \frac{\partial I}{\partial t} = 0. \quad (1)$$

The latter is called the motion constraint equation. It expresses the relationship between the spatial and temporal derivatives of the image in terms of the image velocity $(v_x, v_y)$. In particular, it expresses a relationship between what we want to estimate – $(v_x, v_y)$ – and the image quantities that we can directly measure, namely partial derivatives of intensity.

This constraint is closely related to the constraint on binocular disparity that we discussed last lecture. With binocular disparity, we assumed that the left and right eye images $I_{left}(x, y)$ and $I_{right}(x, y)$ were the same except for local horizontal shifts by the disparity $d$ which was the quantity that we wanted to estimate. Here with image motion, we assume that (projected) image positions...
of 3D points are moving over time and that the image intensity of each (projected) point is not changing. Here the quantity we want to estimate is the local velocity \((v_x, v_y)\).

Given a function \(I(x, y, t)\), one can compute the three partial derivatives. But can one estimate for \((v_x, v_y)\) from these local derivatives at \((x, y, t)\) alone? Unfortunately not, since Eq. (1) only gives one linear constraint at each point and this equation has two unknowns, namely \(v_x\) and \(v_y\). All we can say is that \((v_x, v_y)\) lies on a particular line in the 2D space of \((v_x, v_y)\). See figure below. The shortest such candidate velocity vector (shown in blue) is normal to the line, and hence it is called the *normal velocity*.

Another way to express the same constraint is to consider XYT space and write Eq. (1) as

\[
\left( \frac{\partial I}{\partial x}, \frac{\partial I}{\partial y}, \frac{\partial I}{\partial t} \right) \cdot (v_x, v_y, 1) = 0.
\]

which says that the 3D vectors \((v_x, v_y, 1)\) in red are constrained to be perpendicular to the 3D intensity gradient vector. For this example, these vectors lie on a vertical line for this example because \(\frac{\partial I}{\partial y} = 0\) and so \(v_y\) can be anything. As an exercise: what is \(v_x\) in this example?

**Aperture problem and its solution**

The ambiguity of the motion constraint equation is often called the *aperture problem*, since we can think of viewing the image through a small aperture in space-time such that only the first order
partial derivatives can be computed. Note by "aperture" here, I’m not talking about a camera aperture like in lecture 2. Rather I’m just talking about holding a solid surface in front of the eye such that the surface contains a small hole (aperture).

The aperture problem is more general than this, though. It applies anytime one has a moving 1D pattern. For example, the illustration below shows a set of oblique parallel stripes that are moving, either horizontally, vertically, or obliquely. Given only the motion in the aperture, one cannot say what the "true" velocity vector is. This problem is also related to the barber pole illusion: [http://www.opticalillusion.net/optical-illusions/the-barber-pole-illusion/](http://www.opticalillusion.net/optical-illusions/the-barber-pole-illusion/)

To avoid the aperture problem and estimate a unique velocity vector, one needs two or more such equations. The natural way to do so is to assume that the velocity vector \((v_x, v_y)\) is constant over some local image region, and to combine constraints of Eq. (1) from two nearby points whose spatial gradients \((\partial I/\partial x, \partial I/\partial y)\) differ. Since the two points would define two different lines in \((v_x, v_y)\) space and the true velocity must lie on both these lines, one can solve for the true velocity vector by computing the intersection of the two lines. This is called the intersection of constraints (IOC) solution.

As examples, see figure below. The red vector is the IOC solution and the blue vectors are the normal velocities. The one on the right is counterintuitive because both lines have a normal velocity that is downward to the right, but the true solution is upwards to the right. This is surprising because one might have expected that the true solution should be "between" the normal velocity motion vectors defined by the two given constraints somehow.
Retinal receptive fields and time-varying images

The visual system does not compute local derivatives and solve for the intersection of lines, in the way I just described. To model how the visual system estimates image motion, we need a solution that is based on cells that respond to time varying stimuli.

Let’s begin the retina. When we discussed the center-surround structure of ganglion cells in the retina, we ignored the temporal dimensions except that we noted there are two classes of retinal ganglion cells – the midget and parasol – and that the parasol is sensitive to time varying stimuli. Let’s say a bit more about these two types of cells.

In general, the response (rate) of a ganglion cell at any time \( t \) will depend on the image in some local spatial neighborhood and on some local time interval in the past. Consider the XT slices for three cells below. Their temporal receptive field lies in the range \( t < 0 \) and this is meant to illustrate the receptive field weights for determining the response (firing rate) at time \( t = 0 \). The receptive field must be non-negative only for \( t < 0 \) since the cell’s response cannot depend on something that hasn’t happened yet.

Before we compare the three cells, note that the Y dimension dependence is similar to the X dimension since these cells are DOGs in XY. Think of rotating the cell’s receptive field around its central vertical axis.

The cell on the left would compute a weighted average of the filtered intensities over some time interval, proportional to \( \sigma_t \). In XYT space, the cell receptive field is spherical shaped, with a central positive (ON) cylinder aligned with the \( t \) axis, and the cylinder is surrounded by a negative (OFF) tube. There are such cells in the retina, namely the midget cells. These cells do not respond to changes in light intensity over time.

The two cells on the right are sensitive to changes in the intensity over time. For each \( x \), there are both positive and negative weights for previous times. For the contribution from the spatial center of the receptive field, the cell will respond well if the intensity was increasing in the recent...
past. For the contribution from the surround, the cells will respond best if the intensity is decreasing in the recent past. Such cells have indeed been found in the retina, namely the parasol cells. Note that the middle cell responds better to slow intensity changes, and the right cell responds better to fast intensity changes. You will not be surprised to learn that people have modelled these time varying receptive fields. I will skip the details since we won’t use those models.

The more important point to make is that, although these cells are sensitive to intensity changes over time, they are not tuned to any particular direction of motion. Because the spatial dependence of these cells is a $DOG(x, y)$ function, the cells are rotationally symmetric in XY. To find directionally selective cells in the brain, we need to go to area V1.

**Directionally selective cells in V1: 3D Gabor model**

Many cells in V1 – both simple and complex – are sensitive to motion direction, and these cells are also sensitive to orientations (XY). How can the responses of such cells be modeled?

For example, consider cells that are most sensitive to vertically orientation spatial structure and to a speed $v_x$ in the $x$ direction. Because of the aperture problem, these cells can only measure the component of motion in the direction perpendicular to the spatial orientation to which the cell is most sensitive. We can model such cells using Gabor functions. But now we are in XYT space and so we need to think about 3D Gabors. In particular we need to think about sine and cosine functions in XYT. Consider a 3D cosine function

$$\cos\left(\frac{2\pi}{N} (k_0 x + k_1 y) + \frac{2\pi}{T} \omega t\right)$$

where $k_0$ and $k_1$ are fixed integers between 0 and $N - 1$, and $\omega$ is an integer between 0 and $T - 1$. Note that we are sampling time just as we are sampling space. This may seem like a bizarre thing to do since, although real retinas sample space discretely (cells are discrete), they do not sample time and so one would think time should be represented continuously. The reason I am using discrete time is that this is a computer science course, and if we want to compute anything with a time variable, then it needs to be discrete.

To see how the above function is a 3D cosine, note the expression in the cosine’s argument has a constant value $c$ along a plane in XYT, namely

$$\frac{2\pi}{N} (k_0 x + k_1 y) + \frac{2\pi}{T} \omega t = c.$$  

The value of the cosine changes with $c$ and one goes from plane to other plane.

To understand this 3D cosine, think of it as a video. Fixing $t$ corresponds to a single frame, and gives a 2D cosine function of $(x, y)$. This cosine has spatial frequency $\sqrt{k_0^2 + k_1^2}$ cycles per $N$ pixels, namely $k_0$ cycles per $N$ pixels in the $x$ direction and $k_1$ cycles per $N$ pixels in the $y$ direction. For fixed pixel $(x, y)$, the video changes like a cosine over time $t$, with temporal frequency $\omega$ cycles per $T$ frames. (As an Exercise, figure out the speed of the wave as it travels over time.)

To model the cell responses in V1, we use a 3D Gaussian window on the cosine functions. This gives us a 3D Gabor. Strictly speaking, we can’t properly say whether it is a sine or cosine Gabor or something in between, since it depends on where the spatial Gaussian window is placed relative to the cosine function defined above. If the Gaussian window is centered at the origin, then its a
cosine Gabor. But if it is centered at some unspecified \((\mu_x, \mu_y, \mu_t)\), then we cannot say. Let’s just write the Gabor:

\[
G(x, y, t; \mu_x, \mu_y, \mu_t, \omega, \sigma, \sigma_t) \cos \left( \frac{2\pi}{N} (k_0 x + k_1 y) + \frac{2\pi}{T} \omega t \right)
\]

Also note that a different Gaussian window size is used for \(T\) since space and time are ”apples and oranges”.

As an example, consider a vertically oriented sinusoid shifting to the right. It would have spatial frequency \(k_0 \neq 0\) cycles in the \(x\) direction and because of the vertical orientation, it would have \(k_1 = 0\) cycles in the \(y\) direction. Such a cell has the standard XY receptive field which we’ve seen before several times. An XT slice is as follows (left). An example of a real cell’s receptive field is shown on the right. This is a simple cell, since we can draw the ON (solid iso-curves) and OFF (dashed iso-curves).

There are both simple cell and complex cell versions of these directionally tuned cells. For the complex cell, we take the sum of squared responses of a cosine and sine Gabor, as in earlier lectures.

In XY (single image), the cells will respond best when the image intensity contains oriented structure this is aligned with the ON and OFF regions of the receptive field. Similary, in XYT, the cells will respond best when the moving points sweep along the orientation of the cell (a translating point creates a lines in space-time). It is important to remember that the cells have a preferred XY orientation (line or edge) and so the cells respond best when the plane that is swept out by the line or edge is aligned with the plane of the underlying sinusoid of the Gabor. Also keep in mind the aperture problem: when the moving line (or edge) has a velocity component that is parallel to the line, there will be no image evidence for this velocity component. See again the sketch on p. 3.

So what does this have to do with the computational theory that I presented in the first half of the lecture, namely the intersection of constraints solution? The idea is that each XYT Gabor cell has a frequency \((k_0, k_1, \omega)\) that defines a 3D orientation of (constant value) planes in XYT. This orientation corresponds to a motion constraint equation. [I will try to make this link more explicit in the Exercises.] So, an XYT Gabor corresponds to a motion constraint, in the sense that motion vectors \((v_x, v_y)\) that lies on the motion constraint line should provide a good stimulus for the cell, and motion vectors \((v_x, v_y)\) that lie far from the motion constraint line should provide a poor stimulus. Thus, by examining the set of cells that give a good response, and considering the velocity vector that lies closest to the corresponding constraint lines, one can infer what is the underlying \((v_x, v_y)\).
Velocity tuned cells in area MT (middle temporal lobe)

Cells in V1 are sensitive to only normal velocities. They are the XYT Gabors just discussed. So which area(s) of the visual system contain cells that estimate that there is a particular velocity vector \((v_x, v_y)\)? It turns out there is just one: an area known as MT, which stands for ”middle temporal” (not for ”motion”). Here I briefly sketch out how these cells work. [ASIDE: MT also has normal velocity cells, but I won’t discuss them here. I’ll just pretend that all the normal velocity cells are in V1.]

For any velocity vector \((v_x, v_y)\), there is a set of motion constraint lines that pass through or near that point in velocity space. There is also a set of motion constraint lines that do not pass through the point \((v_x, v_y)\) in velocity space. So, for each directionally selective V1 cell, including cells that are tuned for zero velocity, think of the corresponding motion constraint line. We can build an MT cell that is tuned for motion in some direction \((v_x, v_y)\) by feeding it an excitatory input from those V1 cells whose constraint line passes through or close to \((v_x, v_y)\) and feeding the MT cell an inhibitory input from V1 cells whose constraint line does not pass through or near \((v_x, v_y)\). One can express this all mathematically, and indeed this has been done – quite successfully. e.g. [http://www.cns.nyu.edu/pub/lcv/simoncelli96-reprint.pdf](http://www.cns.nyu.edu/pub/lcv/simoncelli96-reprint.pdf)

As I mentioned in class, in my opinion this is one of the great success stories in computational vision science.