Initialize

```
In[1]:=   Off[General::spell1];
          SetOptions[ArrayPlot, ColorFunction → "GrayTones", DataReversed → True,
            Frame → False, AspectRatio → Automatic, Mesh → False,
            PixelConstrained → True, ImageSize → Small];
          SetOptions[ListPlot, ImageSize → Small];
          SetOptions[Plot, ImageSize → Small];
          SetOptions[Plot3D, ImageSize → Small, ColorFunction → "GrayTones"];
          SetOptions[DensityPlot, ImageSize → Small, ColorFunction → GrayLevel];
          << VectorFieldPlots`
          nbinfo = NotebookInformation[EvaluationNotebook[]];
          dir =
            ("FileName" /. nbinfo /. FrontEnd`FileName[d_List, nam_, ___] ⧴
               ToFileName[d]);
```

Outline

Last time

■ **Shape from shading for a lambertian surface**

■ **Formal ambiguities in the generative model, the bas-relief transform**

■ **Task analysis: "Shape for X" problem**

■ **Task analysis: discounting illumination & viewpoint -> constrains shape from shading**

   Genericity principle: Visual percepts corresponding to primary variables should produce images that are insensitive to small variations in the secondary variables.

   A generalization of the general viewpoint constraint, where viewpoint is a secondary variable and 3D shape is the primary variable. Related to accidental view arguments for shape interpretation.

   But is also applicable to illumination direction as a secondary variable, with shape or material as primary.

Today

   • Early motion measurement--types of models

      correlation

      gradient

      feature tracking

   •Functional goals of motion measurements

   • Optic flow

      Definition: motion field vs. optic flow

      Cost function (or energy) descent model

      A posteriori and a priori constraints

      Gradient descent algorithms

      Computer vs. human vision and optic flow

         -- area vs. contour

Introduction to motion

It is convenient to divide motion processing up into several stages:

> • Early motion measurement,
>
>> optic flow
>
> • Selection & integration, e.g. global constraints
>
> • End uses of motion measurements, e.g. direction of heading

This lecture looks at mechanisms of early motion measurements, and in particular the approximation of the 2D motion field using optic flow.

The mechanisms of early motion measurement are often interpreted as preliminary steps to obtain local estimates of some aspect of motion flow. Each point of a surface describes a 3D trajectory that maps onto a 2D image trajectory. The problem of motion flow is to estimate at each point in space and time a vector that points in the direction of the image point's trajectory, and whose magnitude is the speed. This is the projected velocity field.

Estimates of the projected velocity field can then be used to estimate useful scene related information such as direction of heading and spatial layout. But the sort of image measurements needed depends on the later uses of these motion measurements. For example, the full vector field may not be needed, because one might be satisfied with only direction information. And even this might be superfluous if we only need to know whether something has moved or not. Let's consider the jobs of vision that use motion.

Functional goals of motion measurement

Before embarking on a detailed study of theories of motion measurement, we should stop and list some of the uses of this information. What is motion information good for?

> • change detection - "first alert"
>
> • image or gaze stabilization
>
> • segmentation--breaking camouflage
>
> • object tracking
>
> • three-dimensional structure-from-motion, object recognition
>
> • relationship of the viewer to the environment
>
>> -- time to contact
>>
>> -- direction of heading
>>
>> -- 3-D layout

In[17]:=
```
nsize = 128; subnsize = 32;
noise = Table[RandomReal[], {nsize}, {nsize}];
subnoise = Table[RandomReal[], {subnsize}, {subnsize}];
Manipulate[
 dnoise = noise;
 dnoise[[x1 ;; x1 + subnsize - 1, y1 ;; y1 + subnsize - 1]] = subnoise;
 ArrayPlot[dnoise, PlotRange → {0, 1}, PixelConstrained → {1, 1}],
 {{y1, 20, "Horizontal"}, 1, nsize - subnsize, 1},
 {{x1, 20, "Vertical"}, 1, nsize - subnsize, 1}]
```

Out[20]=
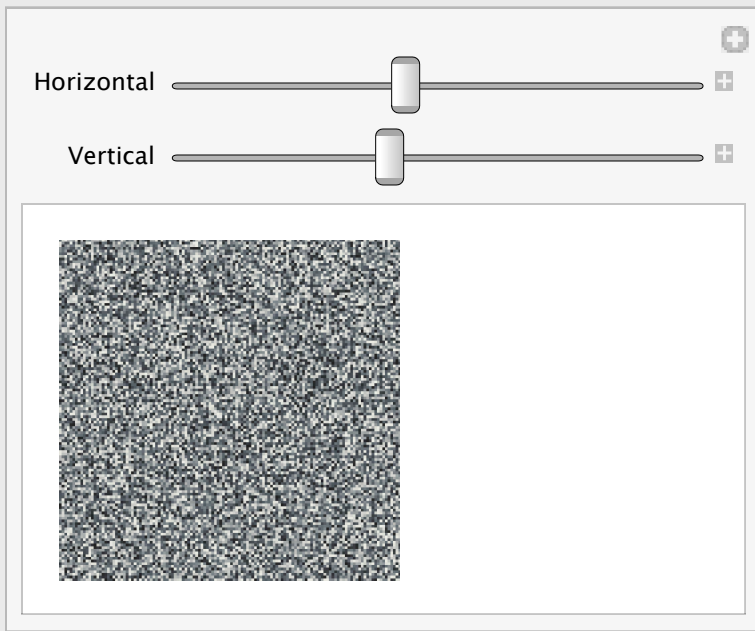
Keeping in mind these goals provides some important constraints on our consideration of the models of early visual motion measurements. One distinction that we mentioned in the context of shape from shading is whether we need to estimate a dense field or a sparse one. A sparse set of velocity estimates would be important for efficiently describing the trajectories of objects--i.e. at a given instant, one velocity vector for each object could be useful for deciding the chance of collision. On the other hand, a dense velocity field could be useful for segmenting an object from a background to determine its shape.

Let's first look at how a change in image data is locally related to the velocity of a point in the scene.

Models of early motion measurement

As we will see shortly, the problem of estimating motion flow is complicated by a number of factors, one of which is that we don't have the motion flow directly available at the image. What we do have, are estimates of intensity change in time at each location, which we will call optic flow. Let's set this lecture in context and outline several approaches to motion measurement.

■ Correlation models

Correlation-like models use image correlations to find spatial shifts that can account for image intensity shifts over time. The work dates to the pioneering results of Reichardt and colleagues in Germany on the fly and beetle visual systems (Hassenstein, B., & Reichardt, W., 1956). Later, mechanisms of directional selectivity were studied in vertebrate retina of the rabbit (Barlow, H. B., & Levick, R. W. , 1965).

Some phenomena of human motion perception can be explained by modified versions of these early correlation or correlation-like models (Van Santen, J. P. H., & Sperling, G. , 1985; Adelson, E. H., & Bergen, J. R. , 1985). The possible synaptic basis of early motion selectivity has a long history (e.g. Koch, C., Torre, V., & Poggio, T., 1986; He, S., & Masland, R. H. (1997), Barlow, H. (1996)).

■ Gradient methods

Gradient models were developed in computer vision. Rather than being derived from an analysis of biological mechanisms, gradient models were developed from a computational analysis of the problem, in conjunction with the use of known mathematical tools for solving the resulting differential equations.

Below, we will introduce the computational problems of motion detection by looking first at gradient models. We will see that a Bayesian formulation can account for a large number of human perceptual results. We will also see how gradient models are related to biologically plausible receptive field properties and mechanisms.
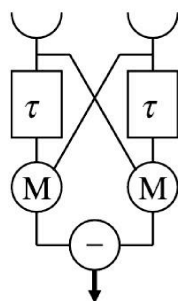
(a) Reichardt detector (b) gradient detector
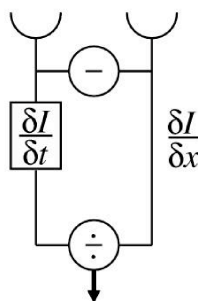
For a recent discussion in the context of the fly visual system, see: Borst, A. (2007). Correlation versus gradient type motion detectors: the pros and cons. *Philos Trans R Soc Lond B Biol Sci, 362*(1479), 369-374. (pdf)

■ **Feature tracking**

Feature tracking. Both correlation and gradient models are low-level in that they are based on local image intensity changes. One could imagine a different kind of system that locates features (conceivably complex, like an "apple") at time t, find that feature again a small time interval later, and from there computes the motion vector.

There is psychological evidence for both correlation and feature tracking models. But the "features" may be defined rather generally. A fair amount of psychophysics has been done arguing for two separate motion systems in human vision (Braddick, O. J., 1974).

Object tracking has been an important problem in computer vision, where discounting view, illumination and occlusion has been taken into account (e.g. Isard & Blake, 1998; Hager & Belhumeur, 1996).

■ **Orientation in space-time**

In the next lecture we will look at models of biological motion detectors that are in effect like "oriented edge detectors" but where the "edges" are in space-time, rather than space-space. We'll also see how they relate to gradient models.

Geometry & kinematics

Generative model describing how the velocity of a point in 3D gets mapped to the velocity of its image point. The nodal point of a lens is where a light ray passes undeviated. Let r_o be a vector representing a point on an object in the 3D scene. Let r_i be the vector representing its corresponding point in the projected image (e.g. on to the retina).
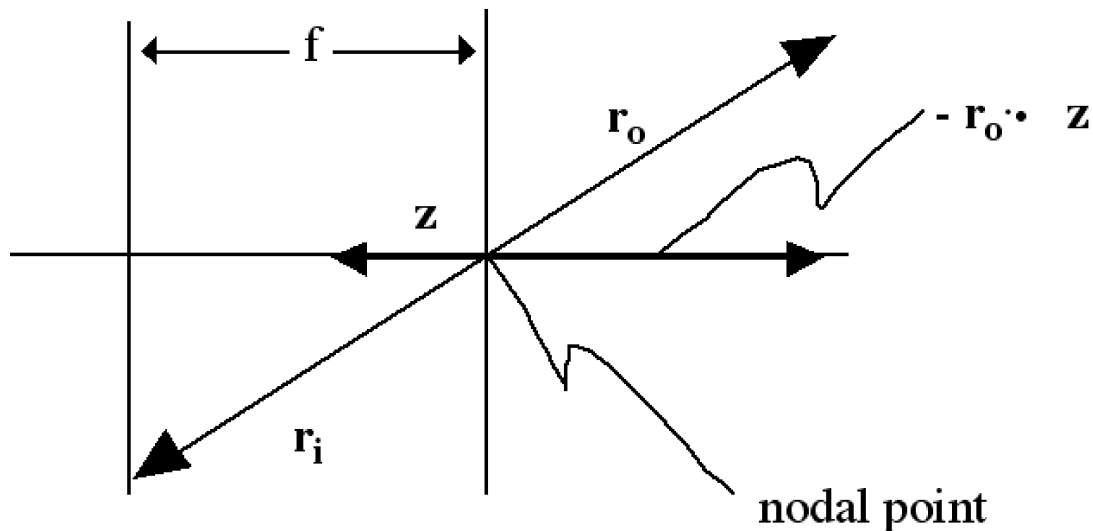
Using similar triangles we have:

$$r_i = \frac{f}{r_o \cdot \hat{z}} \, r_o$$

Object velocity is:

$$\frac{d\,\mathbf{r_o}}{dt} = \mathbf{v_o}$$

And the projected image velocity is:

$$\frac{d\,\mathbf{r_i}}{dt} = \mathbf{v_i}$$

With some vector calculus manipulations, one can arrive at:

$$\mathbf{v_i} = \frac{(\mathbf{r_o} \times \mathbf{v_o}) \times \mathbf{z}}{(\mathbf{r_o} \cdot \mathbf{z})^2}$$

(e.g. See Horn's book Robot Vision)

It is important to underscore that this is just a statement about the kinematics of projection geometry. The equation tells us how to calculate the *motion field* from a description of the 3D coordinates and velocities.

We don't have access to the projected image velocity of 3D world points. The available image data are intensity changes that are not directly related to geometry. How can we estimate the motion flow from retinal or image intensity changes? This, of course, is a scene-from-image inference problem.

## Geometry & photometry: Motion field vs. Optic flow

Image data involves measurements of light intensity over space and time, so it is not immediately obvious how to relate intensity change to geometry changes either in the world or on the retina. We've considered just the geometry to see the relationship between velocity in the world and velocity on the retina--ie. how we can calculate the projected velocity in the image from a knowledge of the 3D velocity of a point.

Our inverse problem is that we would like to go from the retinal flow of brightness (optic flow) to an estimate of the motion field. To emphasize the difference between optic and motion flow, consider a rotating sphere--an illustration adopted from Horn.

Assuming a surface of uniform material, the sphere would not appear to rotate (optic flow would not change), but motion flow would change. However, if the light source is rotated around the stationary sphere, the sphere appears to rotate (optic flow), while the motion field is actually constant. Because the visual system relies on optic flow information, we must understand how to infer motion flow from optic flow.
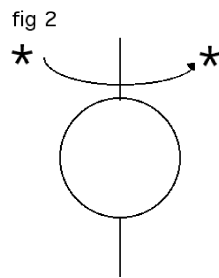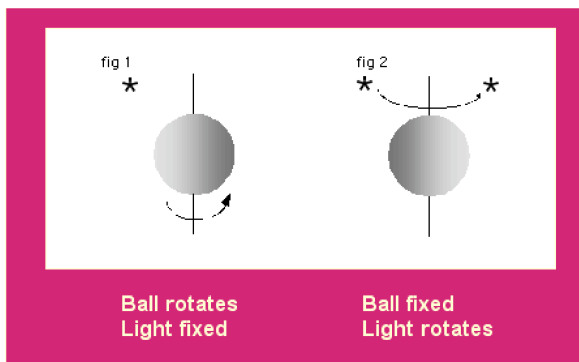
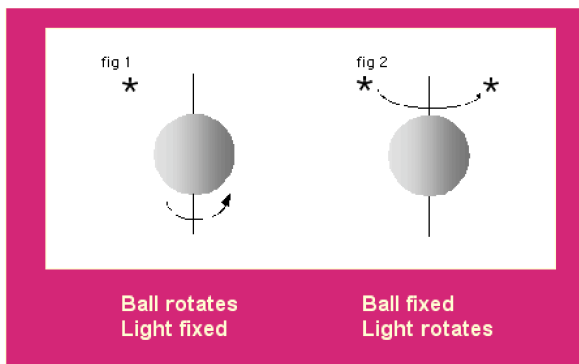### ■ Non-zero motion field

Rotate the sphere

fig 1
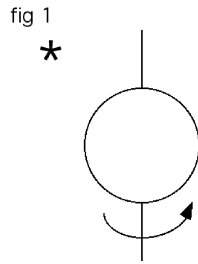
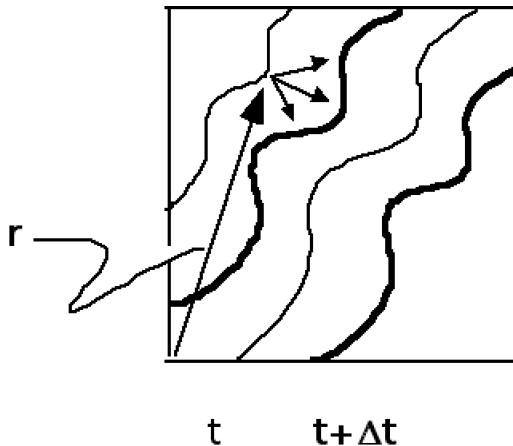## ■ Non-zero optic flow

Rotate the light source

fig 2

■ Demo

Group the two frames below, select them, and then Animate them:

The gradient constraint & the aperture problem

Photometric generative model (inexact) -> gradient constraint

t   t+Δt

If one could pick a point, r in the image, on a curve at time t, and track it to its new location at time t + Δt, the motion flow would be given by dr/dt. In a sense a gradient model can be thought of as a limiting case of feature tracking. In order to track this point, we need to specify some "feature". The simplest is the luminance at point r. Although this could be considered a primitive form of feature tracking, it is so primitive that it belongs in a class by itself. Why is it primitive? The luminance corresponding to a point out in the scene is not guaranteed to remain constant with time in the image. And even if the points do maintain a constant intensity in time, there may be many points at time t+dt that match the intensity of the feature point at time t. But suppose it did, then the total derivative of luminance would be zero:

$$L(x,y,t)$$

$$L(x+\Delta x,\ y+\Delta y,\ t+\Delta t)$$

$$\Delta L \ = \ \frac{\partial L}{\partial x}\Delta x \ + \ \frac{\partial L}{\partial y}\Delta y \ + \ \frac{\partial L}{\partial t}dt \ = \ 0$$

$$\frac{-\partial L}{\partial t} = \frac{\partial L}{\partial x}\frac{\Delta x}{\Delta t} \ + \ \frac{\partial L}{\partial y}\frac{\Delta y}{\Delta t}$$

$$\frac{-\partial L}{\partial t} = \nabla \mathbf{L} \cdot \mu \qquad \mu = (u, v)$$

μ is a quantitative expression for the optic flow at a point, but it is only an estimate of the motion field. Note that even if luminance did remain constant, (not a bad assumption for small Δt), we still have the problem of matching which of possibly many points of the same luminance in the second frame belong to our chosen feature in the first frame. This an example of the correspondence problem, which crops up in several areas of vision including stereo vision.

Aperture problem

Let us look at what the generative equation buys us. What is measurable? Our measurables are: Lt, Lx, Ly., but u and v are unknown. So we have two unknowns for each equation at a point (as with the shape from shading problem). The gradient flow equation provides a constraint on the solution to the problem of calculating optic flow. But it does not specify a unique solution. Again we see an example of a vision probblem that is mathematically underconstrained--even apart from any motion field ambiguity, there are too many optic flow solutions that satisfy the data.

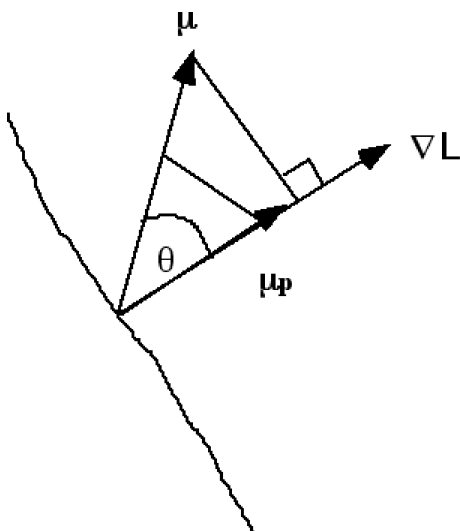Well, what can we estimate? Note that $\nabla \mathbf{L}$ points in the direction of maximum intensity change.

$$\nabla \mathbf{L}$$

We noted this property before in the context of edge detection with gradients in x and y. We can see this by projecting $\nabla \mathbf{L}$ onto a little vector d$\mathbf{r}$ of fixed length that we rotate around so that it makes various angles θ with $\nabla \mathbf{L}$. The dot product

$$\nabla \mathbf{L} \bullet d\mathbf{r} = \left|\nabla \mathbf{L}\right| \left|d\mathbf{r}\right| \cos \theta$$

is biggest when $\cos\theta = 1$, or $\theta = 0$ (e.g. at an edge/contour). So, we can measure the component of optic flow perpendicular to the contour:

$$\frac{-\partial L}{\partial t} = \left|\nabla \mathbf{L}\right| \ \left|\mu\right| \ \cos\theta \qquad \left|\mu_p\right| = \left|\mu\right| \cos\theta \qquad \left|\mu_p\right| = \frac{\frac{-\partial L}{\partial t}}{\left|\nabla \mathbf{L}\right|}$$

This formula makes explicit what we get, and what we lack in our effort to find a unique solution to the optic flow. We can get the components in the direction of the gradient, but not perpendicular to it. This ambiguity is known as the aperture problem. This problem is faced by any purely local measurements of motion. For example, the motion and orientation selective cells in V1, are limited by the aperture problem because of their small, limited receptive field sizes.

The true motion of contour is not known from a local measurement. Only the component (μ_p) of the motion perpendicular to the contour is available
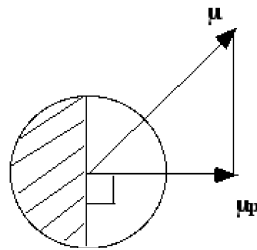
Make a *Mathematica* demo of aperture problem

Constraint lines: A sparse velocity solution

What is the solution to this problem? We need *a priori* constraints, i.e. constraints that are not explicit in the data, but are based on assumptions of what the class of estimates are typically like. If for example, the local measurements come from rigid motion of an object in a plane, the following strategy could be used to estimate the actual velocity

actual object velocity

Constraint lines
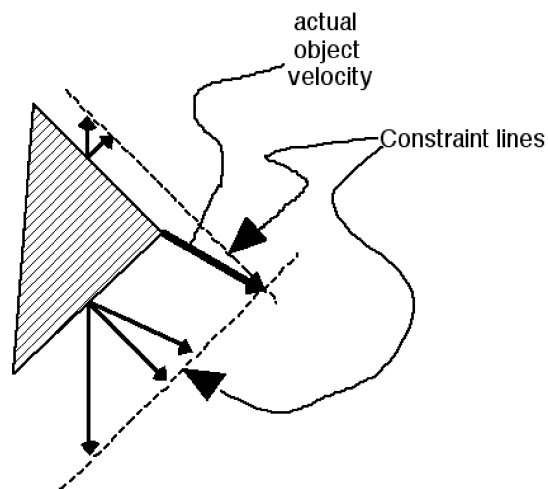
This solution would give us the right motion vector for rigid motion in the plane. We will return to the possible physiological solution to this problem later. One could ask whether there are more general constraints that are not restricted to assuming rigid planar motion.

Smoothness constraint to solve for a dense optic flow field

Combining prior smoothness constraint with generative constraint

An early method due to Horn & Schunck. Dense motion field. "Area-based" rather than contour-based.

One way to solve the problem of being underconstrained, is to add some fairly general constraints on the class of allowable optic flow fields. Horn and Schunck's idea was to assume that flow fields are smooth almost everywhere. This constraint is generic. That is, the hope is that it will more or less work regardless of the class of scenes causing the flow. If one knew ahead of time that the flow would always be caused by a rotating cube illuminated from above, one could use this knowledge to constrain the optic flow solution. This would be a domain specific constraint.

What is a suitable measure of smoothness?

$$(\nabla^2 u)^2, (\nabla^2 v)^2 \qquad (|\nabla u|, |\nabla v|, \text{ etc. also possible})$$

Then, the constraints are that the flow estimates should be consistent with that expected from required the total derivative of the image data to be close to zero:

$$E_i[u, v : x, y] = (\frac{\partial L}{\partial t} + \nabla L \cdot \mu)^2$$

We will call this is an a posteriori constraint, because it is based on the data.

The second constraint is an a priori constraint that the flow estimates be locally smooth:

$$E_s[u, v : x, y] = (\frac{\partial^2 u}{\partial x^2})^2 + (\frac{\partial^2 u}{\partial y^2})^2 + (\frac{\partial^2 v}{\partial x^2})^2 + (\frac{\partial^2 v}{\partial y^2})^2$$

Es is the smoothness cost function.

"a priori" means that we were assuming this about our solution independent of the data, as in a Bayesian prior. We saw this distinction between a priori and a posteriori constraints in shape from shading.

The above two local constraints can be used to construct a global cost function, or more precisely functional (if the optic flow is assumed to be a continuous function):

$$e[u(x, y), v(x, y)] = \int\int (E_i[u, v : x, y] + \lambda E_s[u, v : x, y]) dx dy$$

The goal is to find the optic flow, u(x,y) and v(x,y), that minimizes this functional.

We saw a similar cost function before in the Ikeuchi and Horn shape from shading cost function.

$$e[p(x, y), q(x, y)] = \int\int (L(x, y) - L_R(p, q))^2 + \lambda((\nabla^2 p)^2 + (\nabla^2 q)^2) \, dx \, dy$$

$$e[p(x, y),\ q(x, y)] = \iint (L(x, y) - L_R(p, q))^2 + \lambda((\nabla^2 p)^2 + (\nabla^2 q)^2)\ dx\ dy$$

How can we find a solution?

One way to do this is to use the method of gradient descent. Here we are going to be taking an entirely different gradient-- not the gradient of intensity, but the gradient of the cost function e.

Visualization of gradient descent in 2D -- (Note: "gradient" on a different function!)

Gradient descent technique

Gradient descent is an important numerical technique that crops up in lots of applications including vision.

■ **Cost function or "Energy potential" surface**

Also called error or objective function. If continuous it is called a "functional", in that the energy depends on functions rather than scalar or vector values. If optic flow is a continuous vector field, it is a function, and you'd like to find exactly which optic flow function gives the smallest cost function.

Whatever it is called, the idea is that finding a solution requires finding the argument values that minimize the energy (under certain conditions, this is equivalent to finding the arguments that maximize the posterior probability).

Let's look at a small dimensional example. Here is the energy surface e = z(u,v):

```
z[u_,v_]:= Sin[2 u] Sin[2 v];
```

```
Plot3D[z[u,v], {u,-2,2}, {v,-2,2}, ViewPoint->{1.3, -2.4, 2},
PlotRange->{-3,1}, Axes->False,ImageSize->Small]
```
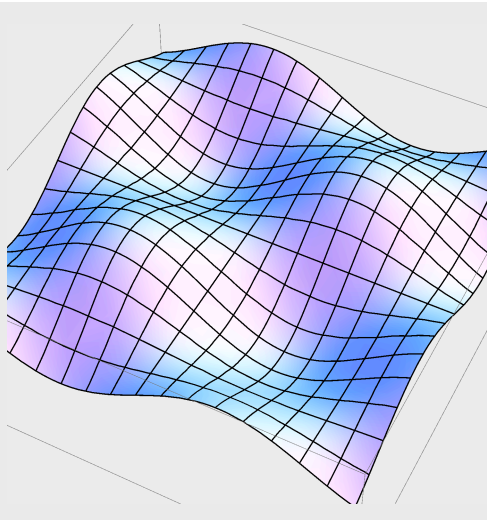

By eye, we can see roughly where the minima sit--there are two minima where z(u,v) = -1, i.e. at {Pi/4, -Pi/4} and {-Pi/4, Pi/4}. But imagine that z is a surface embedded in a much higher dimensional space. How then can we find the minimum when we can't just look at a plot and guess? Gradient descent is a method that will work in the 2D case, but can also be applied to much higher dimension like finding the optic flow (u's and v's) that minimze the above error function e.

■ Gradient field of the cost function

We've seen the gradient of a function used in two earlier contexts. First, we briefly noted that the gradient of an image produces a vector field where the vectors point in the direction of maximum intensity change (i.e. tend to be perpendicular to edges). Second we noted that the gradient of a depth function provided a surface normal representation. Now we are going to use the gradient from vector calculus in a third way. Because the gradient points in the direction of maximum change, if we take the gradient of our cost function e= z(x,y)

```
gradZ[u_,v_] := Evaluate[{D[z[u,v],u],D[z[u,v],v]}];
```
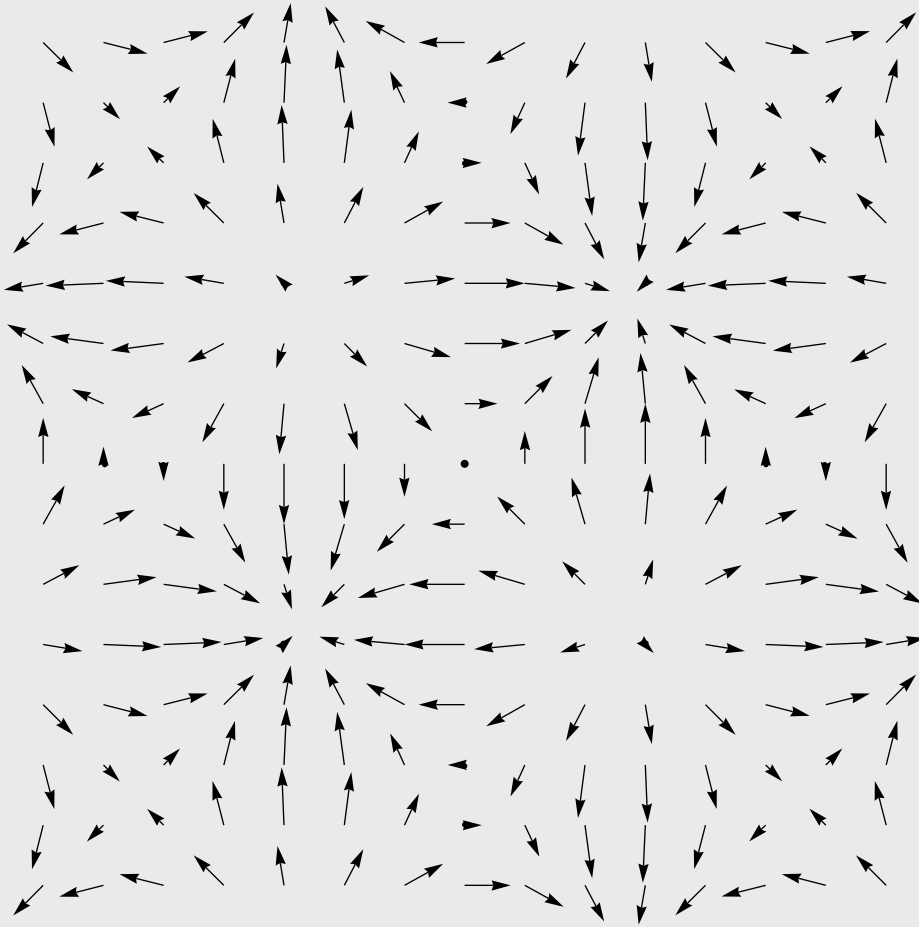
we get,

```
gradZ[u,v]
```

$\{2 \cos(2\,u) \sin(2\,v),\ 2 \cos(2\,v) \sin(2\,u)\}$

Or visualizing it using **PlotVectorField[ ]**,

```
gvf = VectorFieldPlot[gradZ[u,v],{u,-2,2}, {v,-2,2},DefaultColor->Hue[0.99]
```
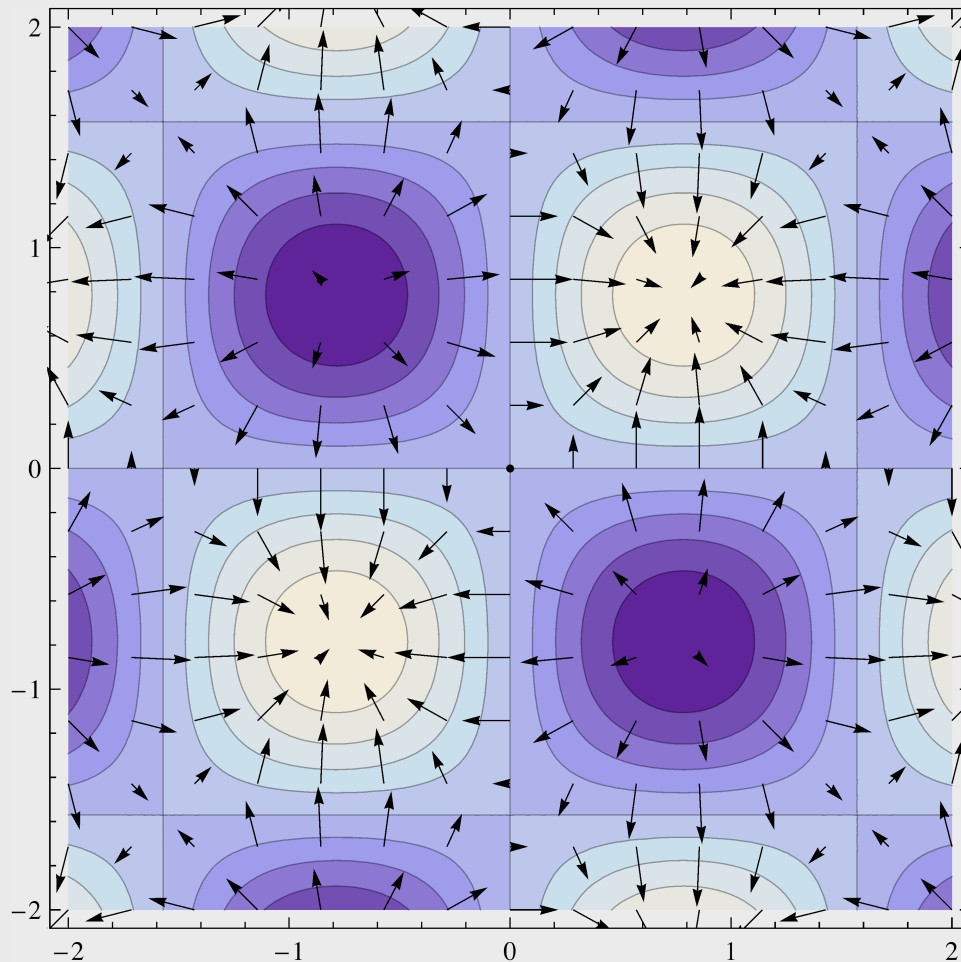

we see a set of vectors each of which always point towards the direction of local steepest ascent on the surface.

Note: you can use the add-in function which will compute the gradient for you:

```
GradientFieldPlot[z[u,v],{u, -2, 2}, {v, -2, 2}]
```

■ **Let's make a contour plot of the cost function and put the vector field on top:**

```
gc = ContourPlot[z[u, v], {u, -2, 2}, {v, -2, 2}];
Show[gc, gvf]
```


So each vector says which direction to take a step if you are climbing. And its length tells you how high a step you'll have to take. If you want to go *down* the landscape, just go in the direction given by the *negative* of the gradient, -gradZ[u,v]. The negative just turns the arrows around to put in the opposite direction. *Gradient descent* is what we will calculate below.

■ Gradient descent

The idea behind gradient descent is this. Suppose you are plopped down randomly somewhere on a smooth cost function landscape. If you begin to take steps in the direction given by the negative gradient, you will go down...and this is better than wondering around aimlessly. If there is a lowest spot, then you can check that you are there, because the gradient there is zero. Gradient descent isn't perfect. If your steps are too big, you could step right over the minimum. You could end up in different minima depending on your starting position. Further, the minimum you end up in could be a local rather than global minimum. Sometimes this is OK, but it isn't the best for optic flow estimation where we normally want to find the position on the landscape (i.e. the optic flow) that minimizes the overall error or cost the best--i.e. the global minimum of the cost function.

Let **r = {u,v}** be a location on the cost function landscape. We are going to start off at **r0 = {u0,v0}**, then calculate the u and v directions of the step to take, **{Δu,Δv} = gradZ[r[[1]],r[[2]]]**, scale this step size by **dt (= α)**, and then set the update equation:

newr = r - dt gradZ[r]

Note that this is discretized form of the equation:

$$\frac{\partial \vec{r}}{\partial t} = -\nabla e$$

where - ∇e = -gradZ.

■ Mathematica demonstration of gradient descent

Let's define the above equation in mathematica:

```
u0 = Random[Real,5]-1; v0 = Random[Real,5]-1;
dt = 0.1;
newr[r_] := r - dt*gradZ[r[[1]],r[[2]]];
```

To do the iteration, we use the built-in function NestList[]. NestList[f, expr, n] gives a list of the results of applying f to expr 0 through n times:

```
result = NestList[newr,{u0,v0},200];

gresults = ListPlot[result,
    PlotJoined->False,AxesOrigin->{0,0},
    PlotRange->{{-2,2},{-2,2}},FrameLabel->{"u","v"},
    Frame->True, AspectRatio->1,RotateLabel->False,PlotLabel->"State Space"
    Ticks->None, PlotStyle->{RGBColor[1,0,0], PointSize[0.015]}];
    Show[gc, gresults]
```
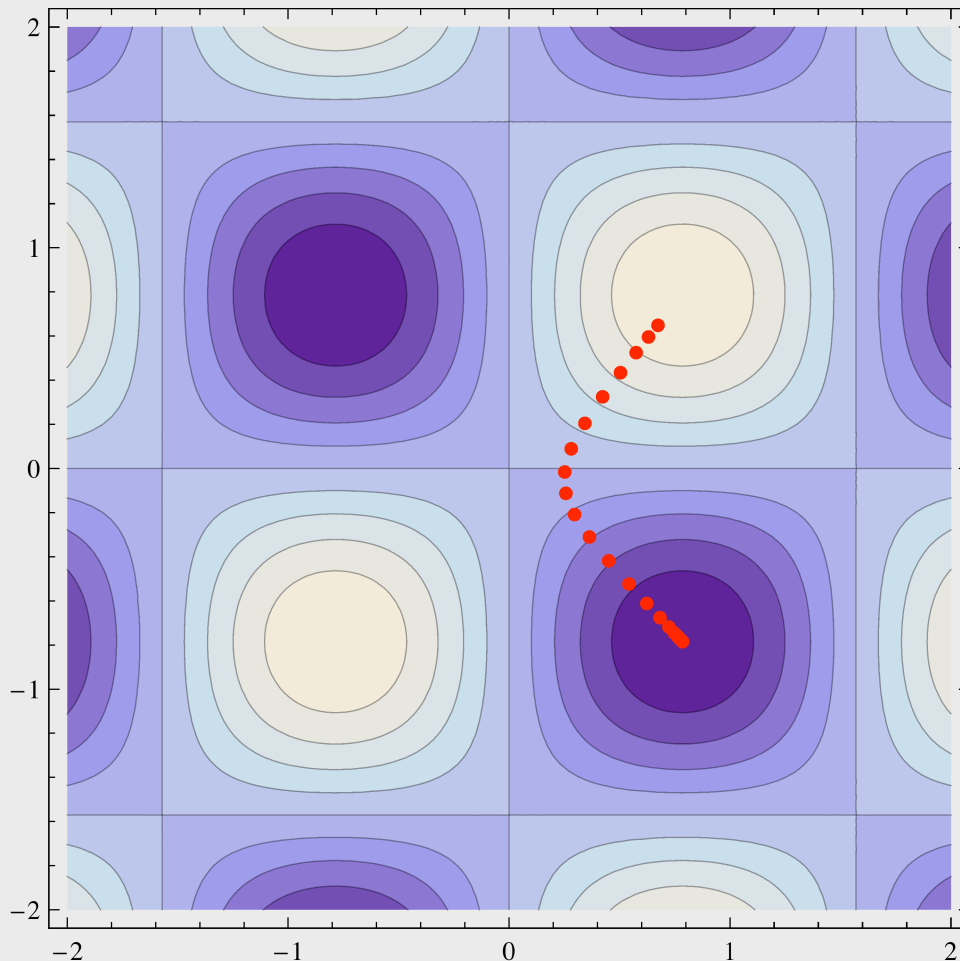

What happens if you start at **r0 = {0,0}**?

For any kind of iterative solution method, we should ask whether the method converges, and if so, whether it converges to the smallest error. A bumpy cost function surface could easily lead our descent into a local minimum--and usually we are interested in the global minimum. It can be shown that the above energy function is typically convex with a single minimum, and thus the above method should provide a unique estimate of the optic flow. This algorithm works fairly well, especially if motion flow at a contour boundary is available to additionally constrain the interior values. One problem, is that without this external constraint, the estimates smooth over discontinuities. A second problem is that a good solution may require many iterations--and is therefore too slow. A lot of research has gone into devising efficient optimization methods, and in actual practice, you would probably want to find some expertise in numerical methods.

Gradient descent solution for optic flow

Let's see how to apply gradient descent to the optic flow problem. One reason that derivatives are chosen for the smoothness constraint is that there exist standard mathematical tools to solve these minimization problems. (See p. 284 in Horn's Robot Vision book for a discussion of the calculus of variations--some mathematics that has been used in many areas of mathematical physics ranging from classical mechanics (least action), to optics (least time), to quantum mechanics.)

Suppose we discretize the flow field into two n x n pixel maps (uik, vik), and estimate the error or cost function at each pair of pixel sites. Now compute the gradient of the error function:

$$\nabla \mathbf{e} = (\frac{\partial e}{\partial u_{11}}, \ \frac{\partial e}{\partial u_{12}}, \ \dots \frac{\partial e}{\partial u_{ke}}, \ \dots \frac{\partial e}{\partial v_{11}}, \ \frac{\partial e}{\partial v_{12}}, \ \dots \frac{\partial e}{\partial v_{ke}})$$

We make an initial guess (perhaps at random) for the initial flow field:

$$u_{ik}^{0}, v_{ik}^{0}$$

Then iterate according to the following rule:

$$u_{ik}^{n+1} = u_{ik}^{n} - \alpha \frac{\partial e}{\partial u_{ik}}\Big|_{u_{ik} = u_{ik}^{n}}$$

$$v_{ik}^{n+1} = v_{ik}^{n} - \alpha \frac{\partial e}{\partial v_{ik}}\Big|_{v_{ik} = v_{ik}^{n}}$$

over all points, (i,k). That is, at each iteration, we move the values of the flow field, (uik, vik), in the opposite direction of the gradient of the error function. That is, we always take steps down the error function surface. That way, we eventually, we come to a minimum. a has to be chosen small enough that we don't hop over minima, but big enough to not take forever to find the minimum.

Human motion perception: area-based vs. contour-based models

Are the representation, constraints, and algorithm that we've assumed above a good model of human motion perception?

The answer seems to be "no". The representation of the input is probably wrong. Human observers sometimes give more weight to contour movement than to intensity flow. Human perception of the sequence illustrated below differs from "area-based" models of optic flow such as the above Horn and Schunck algorithm. The two curves below would give a maximum correlation at zero--hence zero predicted velocity. Human observers see the contour move from left to right-- because the contours are stronger features than the gray-levels. However we will see in Adelson's missing fundamental illusion (next lecture) that the story is not as simple as a mere "tracking of edges" --and we will return to spatial frequency channels to account for the human visual system's motion measurements.
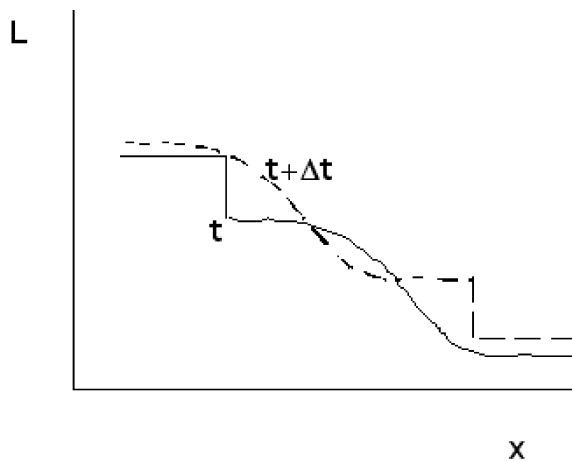

Hildreth's contour model

Dense on a contour, sparse elsewhere.

Some sort of smoothness constraint seems to be important, but it could operate on the contours, rather than intensities. Further, the smoothness constraint has to be generalized (see Yuille, A., & Grzywacz, N., 1988 for a discussion of smoothness constraints). In a superbly written Ph.D. thesis, Ellen Hildreth suggested a contour-based scheme that is consistent with a number of observations from human perception. Her error term was:

$$e = \sum_{over\ contour} (\mathbf{v} \cdot \hat{\mathbf{n}} - |\mathbf{v_n}|)^2 + \lambda \left|\frac{\partial \mathbf{v}}{\partial s}\right|^2$$

Where $\mathbf{v}$ (s) is the velocity at position s of the contour. We want to estimate $\mathbf{v}$(s), but the only data available is the magnitude of the component of $\mathbf{v}$(s) perpendicular to the contour, call it v_n(s). So if possible, we want to choose $\mathbf{v}$ so that v•n = v_n(s), as shown in the figure:
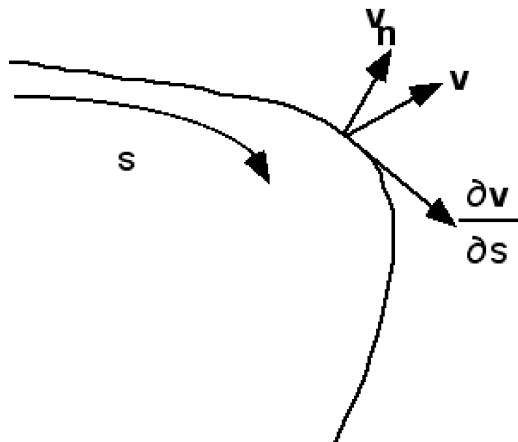
To make the problem well-posed, smoothness is imposed by requiring that a measure of the magnitude of the rate of change of the contour velocity along the contour, e.g. $|\partial v/\partial s|^2$ be small.

Although not strictly necessary from a mathematical point of view, both the area based method of Horn and Schunck, and Hildreth's contour method specified some kind of iterative method for solution. How appropriate is the gradient descent algorithm from a biological point of view?

Gradient descent as well as other methods can be implemented in biologically plausible neural network algorithms--both linear, and non-linear (e.g. Cohen, M. A., & Grossberg, S. (1983). Absolute Stability of Global Pattern Formation and Parallel Memory Storage by Competitive Neural Networks. IEEE Transactions SMC-13, , 815-826.; Hopfield, J. J. (1984). Neurons with graded response have collective computational properties like those of two-state neurons. Proceedings National Academy of Science, USA, 81, 3088-3092.). However, it seems unlikely that biological systems would use a method that could require many iterations--neural elements just aren't fast enough. By the time the flow is computed, we would have been either devoured by the jaguar, or more likely in our society, hit by one on the freeway.

For certain minimization problems, one can show that a feedforward one-shot solution exists. For example, this is straightforward for error functions that are quadratic. where a linear matrix operation can solve the problem in a single shot.

Next time

We have studied some of the computational problems of motion measurement. In the next lecture, we will look at how a Bayesian formulation with the right priors can be used to model a wide variety of human motion results. And we will look at neural systems solutions to the problem of motion measurement.

Neither the area-based nor the contour-based algorithms we've seen can account for the range of human motion phenomena or psychophysical data that we now have. We will look at a recent Bayesian model that uses the area-based gradient constraint, combines it with a particular set of prior assumptions to compute a somewhat dense optic flow field that accounts for a wide range of human motion data.

Appendices

Searching for spatial correlations in two frames

```
argmax2[x_] := Flatten[Position[x, Max[x]]];
(*coords of max in matrix*)
```

```
size = 32; x0 = 4; y0 = 4; pw = 24; xoffset = 3;
A1 = Table[Random[], {size}, {size}]; A2 = A1;
(*A2=Table[Random[],{size},{size}];*)
A2[[Range[y0, y0 + pw], Range[x0, x0 + pw]]] =
  A1[[Range[y0, y0 + pw], Range[x0 - xoffset, x0 + pw - xoffset]]];

ga1 = ArrayPlot[A1, Mesh → False];
ga2 = ArrayPlot[A2, Mesh → False];
```

```
argmax2[ListCorrelate[A1, A1, {size / 2, size / 2}]]
```

{16, 16}

```
argmax2[ListCorrelate[A1, A2, {size / 2, size / 2}]]
```

{16, 19}

How does correlation work if the patch is small?

Try having dynamic noise in the background, by uncommenting (*A2=Table[Random[],{size},{size}];*).

References

Adelson, E. H., & Bergen, J. R. (1985). Spatiotemporal Energy Models for the Perception of Motion. Journal of the Optical Society of America, 2(2), 284-299.

Barlow, H. B., & Levick, R. W. (1965). The Mechanism of Directional Selectivity in the Rabbit's Retina. Journal of Physiology, 173, 477-504.

Barlow, H. (1996). Intraneuronal information processing, directional selectivity and memory for spatio-temporal sequences. Network: Computation in NeuralSystems, 7, 251-259.

Braddick, O. J. (1974). A short-range process in apparent motion. Vision Research, 14, 519-527.

Borst, A. (2007). Correlation versus gradient type motion detectors: the pros and cons. *Philos Trans R Soc Lond B Biol Sci, 362*(1479), 369-374.

Hager GD, Belhumeur PN (1996) Real-Time Tracking of Image Regions with Changes in Geometry and Illumination. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, p 403—410.

Hassenstein, B., & Reichardt, W. (1956). Functional Structure of a Mechanism of Perception of Optical Movement. Proceedings First International Congress on Cybernetics, 797-801.

He, S., & Masland, R. H. (1997). Retinal direction selectivity after targeted laser ablation of starburst amacrine cells. Nature, 389(6649), 378-82.

Hildreth, E. C. (1984). The Measurement of Visual Motion. Cambridge, MA: MIT Press.

Isard M, Blake A (1998) Condensation -- conditional density propagation for visual tracking. International Journal of Computer Vision 29:5--28.

Koch, C., Torre, V., & Poggio, T. (1986). Computations in the Vertebrate Retina: Motion Discrimination Gain Enhancement and Differentiation. Trends in Neuroscience, 9(5), 204-211.

Movshon JA, Adelson EH, Gizzi MS, Newsome WT (1986) The Analysis of Moving Visual Patterns. In: Pattern Recognition Mechanisms (Chagas C, Gattas R, Gross CG, eds), pp 117-151. Rome, Italy: Vatican Press.

Van Santen, J. P. H., & Sperling, G. (1985). Elaborated Reichardt Detectors. Journal of the Optical Society of America, 2(7), 300-321.

Yuille, A., & Grzywacz, N. (1988). A computational theory for the perception of coherent visual motion. Nature, 333, 71-74.