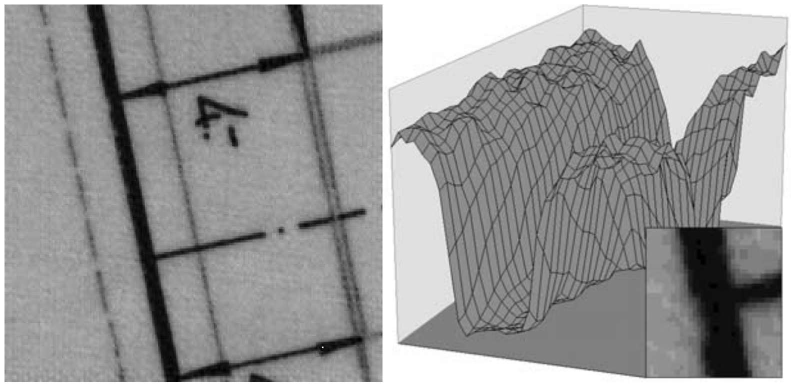
Local image structure

An image assigns has an intensity at each point. If at the point (x_0, y_0) the intensity has a sudden change along a line, there may be an *edge* of an object passing through that point. That is important to detect, for image understanding. This figure ilustrates this: it shows an image



of a drawing and a detail of it, both as an intensity image (in the lower right) and as a plot of the image function. (To see this properly, use at least '`magstep = 3`' in `ghostview`, or print this postscript file).

In order to develop quantitative methods for computer vision, we need to analyze such local structures in an image.

Mathematically, the image can be seen as a function $f : R \times R \to R$ assigning to each 'point' (x, y) the real value $f(x, y)$. If we assume that this function is sufficiently smooth, for instance at least twice differentiable, then we can develop the function as a (2-dimensional) Taylor series around (x_0, y_0)) The first 3 terms of that series then describe the image well, up to second order.

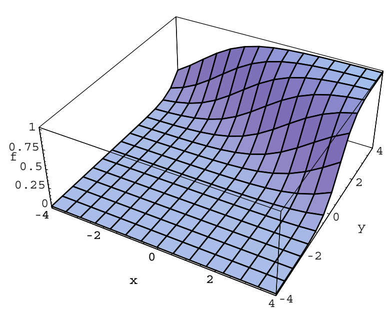
You may have had Taylor series only in 1 dimension (see also `taylor.ps`):

$$f(x_0 + \epsilon) = f(x_0) + f'(x_0)\epsilon + f''(x_0)\frac{\epsilon^2}{2} + O(\epsilon^3). \tag{1}$$

In two dimensions, you have to take partial derivatives of f in the x-direction and y-direction. We denote these by f_x and f_y, and their derivatives by f_{xx}, f_{xy}, f_{yy}, etcetera. Then the Taylor series is, to second order:

$$\begin{aligned} f(x_0 + \epsilon, y_0 + \delta) \quad &\approx \quad f(x_0, y_0) + f_x(x_0, y_0)\epsilon + f_y(x_0, y_0)\delta \\ &\quad + \frac{1}{2}(f_{xx}(x_0, y_0)\epsilon^2 + 2f_{xy}(x_0, y_0)\epsilon\delta + f_{yy}(x_0, y_0)\delta^2) \end{aligned} \tag{2}$$

(1) The function $f(x, y) = \frac{1}{1+e^{-(x+2y-5)}}$ is fairly typical of what we would like to call an edge:



(2) Compute f_x and f_y. As often happens when you compute with exponential functions, you can express these in f (see also `sigma.ps`). Do that, it makes the computation of the second derivatives a lot easier. (**Answer**: $f_x = f(1-f)$, $f_y = 2f(1-f)$.)

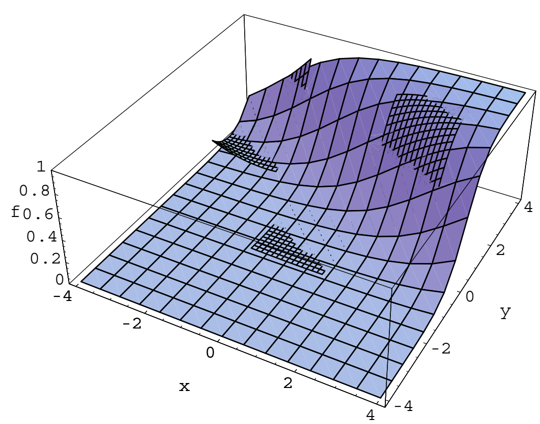
(3) Now compute the second derivatives f_{xx}, f_{xy}, f_{yx}, f_{yy}. (**Answer**: $f_{xx} = f(1-f)(1-2f)$ and $f_{xy} = f_{yx} = 2f(1-f)(1-2f)$ and $f_{yy} = 4f(1-f)(1-2f)$. Note that $f_{xy} = f_{yx}$.)

(4) Give the second order Taylor approximation to f at a point (x_0, y_0).

Answer: in shorthand, with $f_0 = f(x_0, y_0)$:

$$f(x_0+\epsilon, y_0+\delta) = f_0 + f_0(1-f_0)(\epsilon+2\delta) + f_0(1-f_0)(1-2f_0)\frac{(\epsilon+2\delta)^2}{2} \tag{3}$$

(5) The following plot superimposes the Taylor approximations at several places (namely $(0,0), (2,2), (-3,3)$) onto the function. Locally, the approximations are close enough that they are indistinguishable from the surface; but when you move too far, they can get pretty bad.



(6) The occurrence of $(\epsilon + 2\delta)$ suggests that we will get simpler formulas when we change coordinates. We prefer an orthonormal basis, so we use as a coordinate transformation:

$$\begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} \frac{1}{\sqrt{5}} & \frac{2}{\sqrt{5}} \\ \frac{-2}{\sqrt{5}} & \frac{1}{\sqrt{5}} \end{pmatrix} \begin{pmatrix} \epsilon \\ \delta \end{pmatrix} \tag{4}$$

(7) This is a rotation (see `rotation2d.ps`); over which angle? (**Answer**: atan(2).)

(8) Give the Taylor expansion of f in these new coordinates.

Answer:

$$f(x_0 + \frac{u-2v}{\sqrt{5}}, y_0 + \frac{2u+v}{\sqrt{5}}) = f_0 + \sqrt{5}f_0(1-f_0)u + \frac{5}{2}f_0(1-f_0)(1-2f_0)u^2. \tag{5}$$

(9) The new coordinates are *gauge coordinates*, in this case they have been chosen in the direction of the *gradient* (direction of maximum derivative). (It is coincidence of this example that it also makes the second derivative terms so nice, this does not usually happen for curved edges.)

(10) We can use the Taylor series to determine where the function is *locally flat*. This means that the second derivative terms must locally be 0, independent of u and v. At which function values does this happen? At which points (i.e. (x, y) values) does this happen?

Answer: When $f_0(1-f_0)(1-2f_0) = 0$, so when $f_0 = 0$ or $f_0 = 1$ or $f_0 = \frac{1}{2}$. The first two possibilities happen only for infinite x or y (and indeed, the function looks flat there); so the last one is the most interesting. From $f(x, y) = \frac{1}{2}$, we get: $x + y - 5 = 0$, so it happens at the line $y = 5 - x$. This is just at the edge! This is because the edge is already straight in the direction along it (the u direction), and that at the *inflection points* is also must be flat in the u-direction, since the second derivative changes sign.

(11) We thus found the location of the edge. This example had some simplifying properties – for one thing, the edge was straight. In general, to find an edge you have to look for a second derivative that is zero independent of u (in the gradient gauge), at the points where the first order terms are not too small (or it will hardly be an edge at all).

(12) If you want to try your hand at a curved edge, try: $f(x, y) = \frac{1}{1+e^{-(x^2+2y-5)}}$, and follow the same procedure.

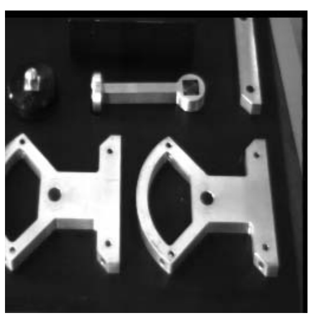
Answer: Some intermediate answers:

$$f(x_0 + \epsilon, y_0 + \delta) = f_0 + 2f_0(1-f_0)(x\epsilon + \delta) + 2f_0(1-f_0)\left((1-2f_0)(x\epsilon+\delta)^2 + \frac{1}{2}\epsilon^2\right) \quad (6)$$

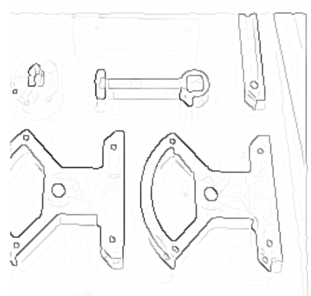
The gradient gauge becomes dependent on position:

$$\begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} \frac{x}{\sqrt{1+x^2}} & \frac{1}{\sqrt{1+x^2}} \\ \frac{-1}{\sqrt{1+x^2}} & \frac{x}{\sqrt{1+x^2}} \end{pmatrix} \begin{pmatrix} \epsilon \\ \delta \end{pmatrix} \quad (7)$$

(13) This may all be very well for functions, because you can do the differentiation. But how does it work in images, and especially discrete images in which x and y change stepwise, form pixel to pixel? In that case you need to *estimate* (or *measure*) the derivatives. Again, your course in computer vision will teach you how to that properly (for instance by filtering images with the derivatives of Gaussians). A result of this kind of edge detection is:



original image



edges

This is one of the best edge detectors currently available. It is called the Canny edge detector; and the Gaussian derivatives are by Koenderink.