Chapter 16

Edge Detection and Image Segmentation

Hierarchies of feature detectors. First Paradigm Computer Vision. Introduction to computer vision: histograms, differential operators in edge detection, composition of smoothing and differential operators (regularized edge detectors) followed by histogram peak selection. Hypothesis testing. Taylor expansions. What is an edge? Can they be defined directly from an image?

16.1 Introduction

In the last lecture we guessed that the structure suggested a hierarchy in which the more abstract, "higher-level" features are supported by a foundation built on solid, "low-level" features. This solution seems so natural, and the terms so easily connected to the physiology, that we can't resist seeing if it actually works. This is, of course, one advantage of the building computational models: they can be tested via simulation. Of course, simulation involves making some simplifying assumptions, but we'll try to be explicit about these.

We begin with the classical computer vision problem of image segmentation, or partitioning the image into a patchwork of sub-images, each of which defines an area of significance. Such local image patches would correspond, in one instance, to the "template" for an object, in which case we would have our first example of separating a FIGURE from BACKGROUND. This, and other issues of perceptual organization, will recur thoughout the rest of these lectures.

Further consideration of what comprises a significant local feature naturally motivates edge detection. This, of course, raises the general question—what is an edge?—a question that is far more subtle than it appeared at first.

Since the goal of segmentation is partitioning the image into pieces corresponding to interesting parts of the visual world, perhaps the problems of working with a single image can be solved by considering more than one. We therefore also take a first look at stereoscoping processing, or processing the image information from two eyes.

Edges, regions, and related structures are image features. During the development we'll have to be clear about the differences between a feature and a feature detector; what the computation of detection is; and what a feature is.

16.2 Segmentation by Simple Statistics

The earliest approaches to image segmentation derived from a biomedical problem not unlike the one faced by *Limulus* in finding a mate. A cartoon of a biological cell imaged through a microscope consists of a (dark) nucleus surrounded by (gray) cytoplasm on a (white) background (Fig. 16.1); the task was to find the cell so it could be analyzed. In effect, this is a problem in which the cell is the "foreground" and the remainder of the image is the "background." The task is to segment the foreground cell from the background.

A count of the number of pixels at each gray level in this cartoon situation demonstrates one of the earliest approaches to segmentation: *Histogram Peak Selection*

1. Count the number of pixels in the image that have each grey level;

2. Plot these as a *histogram*, or number of pixels vs intensity value. Keep the map so that histogram points can be related back to the image. For this example there are three peaks: one for the background, one for the cytoplasm, and one for the nucleus.

3. Place a *threshold* between the peaks corresponding to the cytoplasm and the background; using the map relating points in the histogram to pixels in the image, create a new *characteristic image* in which foreground is mapped to black and background to white.

If the histogram is normalized to sum to 1, it provides an estimate of the probability that a pixel will be gray level (i). This amounts to essentially "throwing a die" with, say, 256 sides, one for each possible grey level.

If the die this thrown independently for each pixel to generate an image, then it will simply be noise; it will resemble the "snow" on an old fashioned TV tuned to a station with no channel.

However, as in the cell example, there is a bias in the die. The values are not uniformly distributed in probability over all intensities; rather, they are biased differently for different parts of the image. (The nucleus is heavily biased toward black; the cytoplasm toward dark gray; and the background toward light gray.)

Histogram threshold analysis is a very simple classifier: those pixels above the threshold are the (foreground) object of interest, while those below are background. This is a purely local decision analysis, analogous to the barnacle's use of a turret to select food. "Noise" on the individual pixel values, whether it is from waves in the

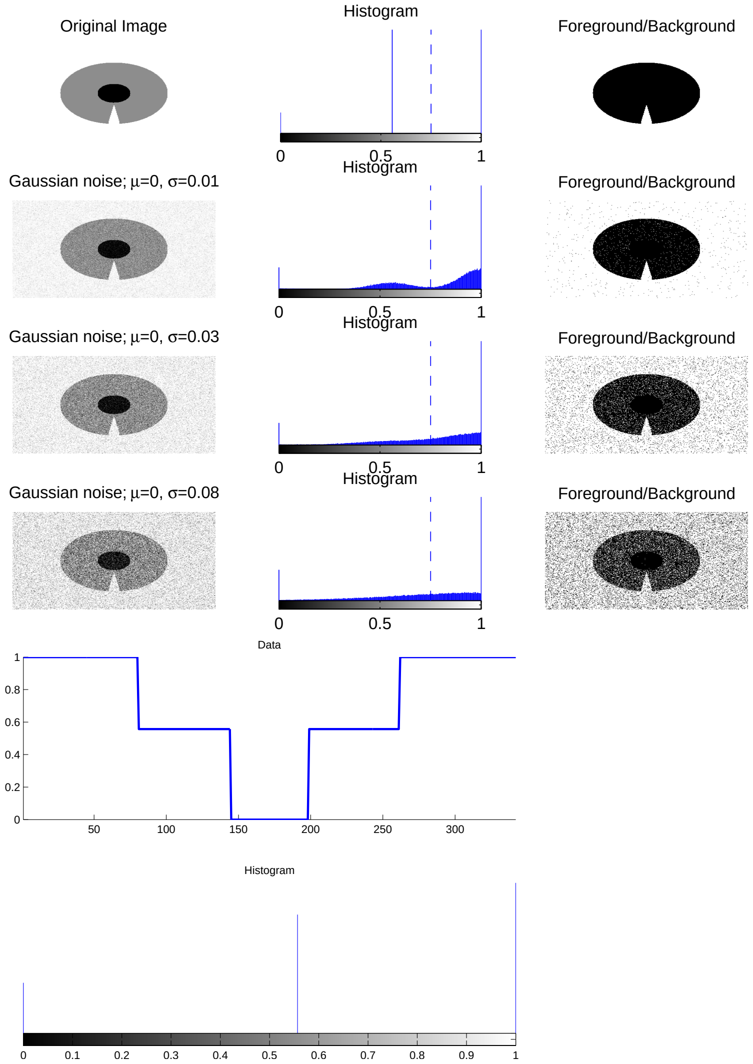


Figure 16.1: Illustration of the histogram, or first-order statistic, for image segmentation. (TOP) Illustration of the cartoon of a cell with a narrow slit. The histogram (middle) indicates clear peaks, so that a threshold can be placed between the cell and the background. The addition of noise creates problems, however. Note how the clear peaks are gradually absorbed into the noise, making the threshold selection problem difficult (if not impossible). The examples shown use the same threshold for each noise level. (Noise: Gaussian i.i.d.; standard deviation shown. (BOTTOM) Scan line through the center of the cell image with its histogram. Image intensities are shown on a normalized scale, running from black (0) to white (1).

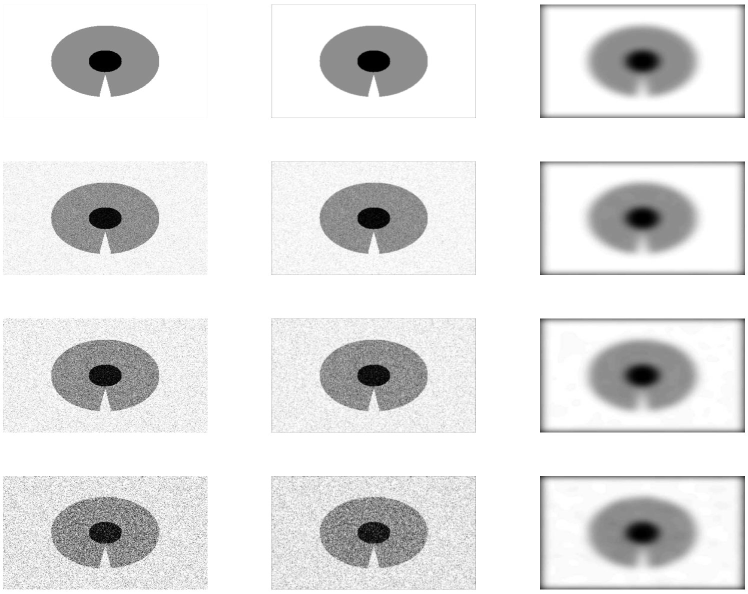


Figure 16.2: Blurring the image has the effect of cleaning up some of the noise, but at the price of altering the structure. The precise location of the boundaries is no longer clear; nor is the shape of the angular indentation.

water, receptor biophysics, or occluding fish, preclude such syntactic rules. We can now appreciate why *Limulus* did not attempt to classify single ommatidia.

16.2.1 Thresholding as Hypothesis Testing

Suppose our image were an extremely simple one: there is a dark object on a light background. The pixel intensities are drawn randomly from two Gaussian distributions (Fig. 16.4:

$$p(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp^{-(x-\mu)^2/2\sigma^2} = p(x;\mu,\sigma). \tag{16.1}$$

Let us assume that the area of the object in the image is the same as the area of the background; then the overall intensity probability is density is $p = \frac{1}{2}p(.;\mu_o,\sigma_o) + \frac{1}{2}p(.;\mu_b,\sigma_b)$.

Basically, we set up the hypothesis: those pixels below the threshold (in intensity) are object and those above are background, and we set the threshold t to minimize the probabilities of misclassification. For this balanced problem in which $\sigma_o = \sigma_b$ we calculate that the optimal threshold is exactly between the means: $t = 1/2(\mu_o + \mu_b)$.

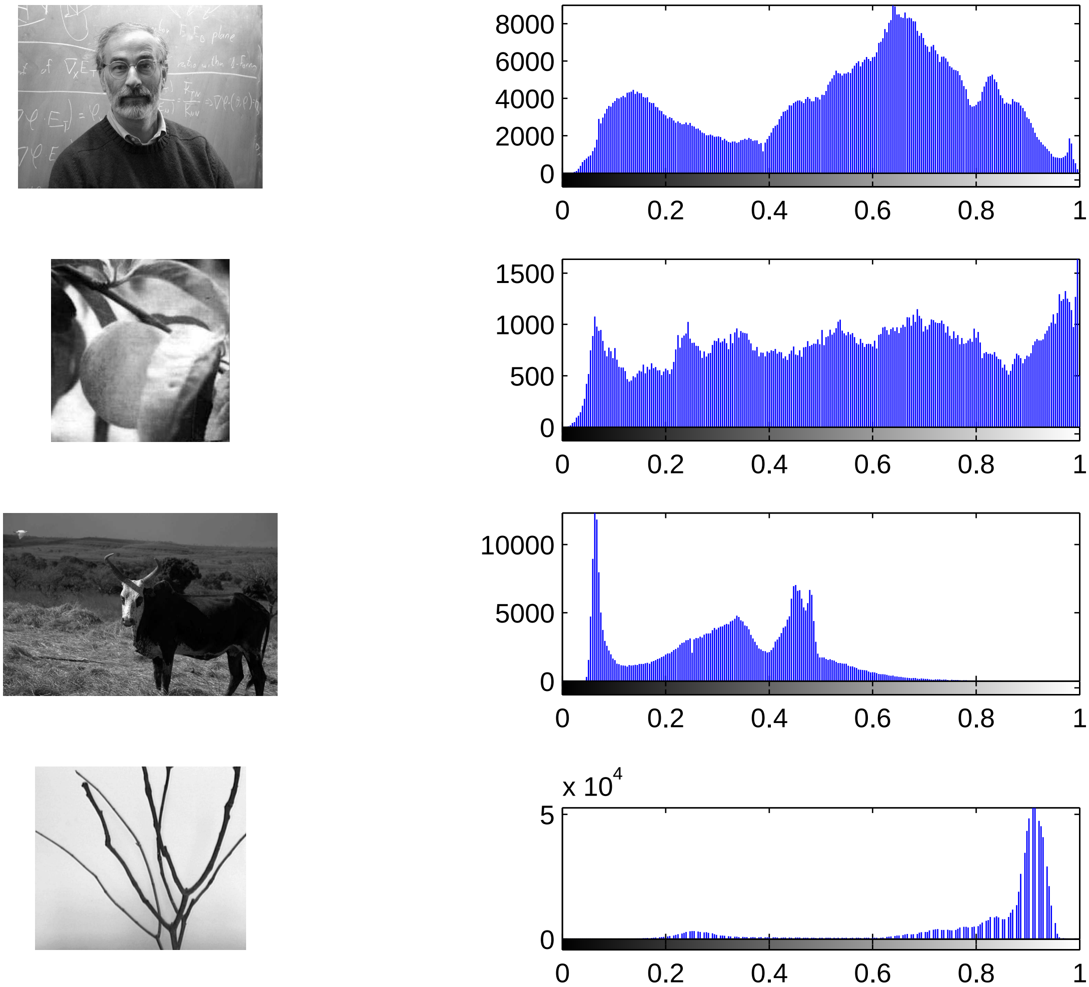


Figure 16.3: Illustration of histograms for several natural images. Where would you place the threshold?

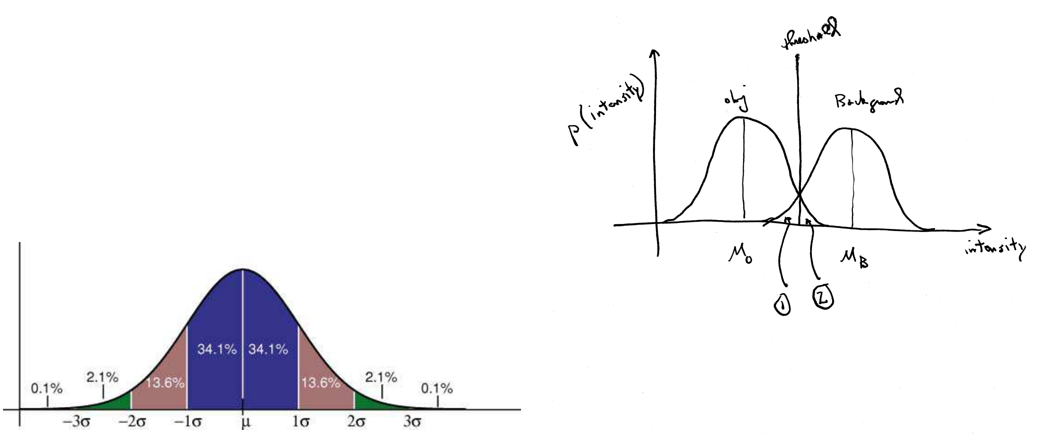


Figure 16.4: Gaussian distribution (left) centered at mean=μ with different standard deviations colored. Note that about 68% of the values of a Gaussian random variable are within ± one standard deviation. (right) Simple model for image intensity distributions in which object and background are balanced Gaussian distributions at different means. A threshold to separate them can be set that minimizes those background pixels marked as object (the area denoted 1) and those object pixels marked as background (2).

16.2.2 Image Statistics

Throwing a die in a random, independent manner would amount to filling "image space" uniformly - but actual images have a lot more structure than the simple example above. A hint of this is given by the different shape of histrograms for natural images, but this is just a hint. To exploit it we need to consider how different pairs, triples, etc., of pixel values co-vary with one another. But how should we decide which pairs? Should they be adjacent, with a single intermediate pixel, or with 37 intermediate pixels? And in which direction?

16.2.3 Regions vs Boundaries

Image segmentation suggests a partition of the image into a set of regions which do not overlap and which cover the image completely. This formulation is area-based: in effect, it would be ideal if the image regions corresponded exactly to objects of interest, and could be obtained simply by linking together nearby pixels with identical (or similar) intensities. While this might work for the cell image, it is unclear how to do such linking for natural images.

The complementary approach is to focus on boundaries, or places of change in intensity. This is suggested by our experience with primate physiology: explore nearby pixel values to see if they are the same or different; i.e., if a bright/dark boundary is present. That is, exploit the way nearby intensity values change to find boundaries and develop descriptions of regions. One hypothesis: intensities will change rapidly across boundaries between objects, or between object and background, but more slowly within an object.

16.3 Edges and Intensity Changes

In the cartoon cell example, different regions of the image are defined by identical intensity values: borders between image regions then correspond to the boundaries between nucleus and cytoplasm and between the cell and the background. In effect we seek an outline of the cell and the nucleus.

Exploring the 1-D slice through the cartoon cell (Fig. 16.1, bottom), we immediately observe that nearby intensity differences have either value 0, if within a constant region, or ±0.5 at a boundary. The sign indicates the direction of transition.

To make this a little more formal, suppose the image has been sampled from a one-dimensional function of intensity, $I = I(x)$. If the spacing between samples is Δx, then the increment in intensity is given by:

$$\Delta I = I(x + \Delta x) - I(x).$$

We immediately recognize this as being related to the definition of derivative in Chapter **??**. Normalizing by the increment Δx, we write the (finite) value of the limit

$$\lim_{\Delta x \to 0} \frac{\Delta I}{\Delta x} \to I'(x).$$

But this only holds, of course, if the intensity function $I(x)$ is differentiable. Before this limit is achieved, however, we can write the related equality:

$$\frac{\Delta I}{\Delta x} = I'(x) + \epsilon \tag{16.2}$$

where the error $\epsilon \to 0$ as $\Delta x \to 0$. The intensity increment is therefore a function of two quantities:

$$\Delta I = I'(x)\Delta x + \epsilon \Delta x.$$

The first term is called the *differential* of the intensity (at the increment Δx) and it dominates the second term, a higher-order infinitesimal, because the second term goes to 0 more rapidly than the first (provided $I' \neq 0$), since it consists of a product of decreasing terms.

Notice that the first term is linear and for now we shall focus entirely upon it.

It is possible to think of this differencing operation as an operator D:

$$D = [+1, -1]. \tag{16.3}$$

Applying it to the image, $D(I(x)) = D * I(x)$ can then be achieved as a convolution, high values of which isolate edge points on the perfect cell image; see Fig. **??**. For example, histogramming the output values makes a threshold selection process clear. This is an example of an ideal step edge (see Box).

Ideal Step Edges

The 1-D cross-section through the cell cartoon can easily be represented as a sequence of step functions, the derivative of which is a sequence of δ-functions (recall Chap. **??**). These δ-functions then localize the edges exactly. For this reason step changes in intensity are often the first model of edges that is employed. See Fig. 16.5, (left).

Once an ideal step passes through a lens, e.g. with a Gaussian point spread function, it is no longer discontintinuous; it now has a sigmoidal shape (Fig. 16.5, right). Its first and second derivatives are as shown; all higher-order derivatives exist as well. (The reason for this is a point we shall come back to.)

A sequence of such sigmoidal functions can be taken to define the limiting process for the (generalized) step function.

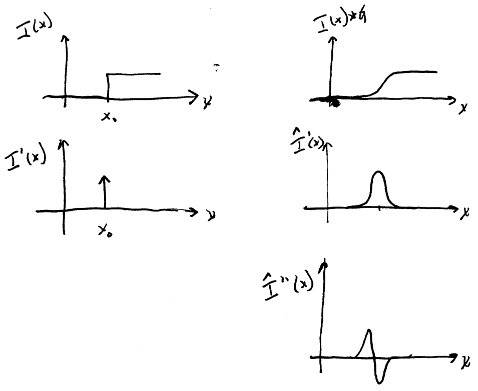


Figure 16.5: Step change in intensity and its derivatives. (TOP, LEFT) Ideal case in which the derivative of a step is a δ-function. (RIGHT) Regularized case in which the step function is first convolved with a Gaussian blur filter; this converts the step into a sigmoid-shaped function and makes it smooth. The first derivative is then a "blurred impulse" and the second derivative crosses through 0 at precisely the location of the original step discontinuity.

Since the value of the step function is undefined at the point of change (the location of the edge), the blur introduced by a real imaging system guarantees that this is not a problem. For the discrete sampling, moreover, it suggests an evaluation of points on one side as compared against points on the other.

But how should we accomplish this numerically? The limiting process can go forward (from a point) and it can go backwards; for a smooth function both of these should be equivalent. But now structure is fixed at a finite scale; if forward and backwards limits are taken they can be averaged. It seems intuitive that the errors in the first calculation will depend on the step size Δx, but what about the last one? Will the averaging decrease the error? Each of these is a plausible scheme for computing a numerical approximation to the derivative:

- FORWARD DIFFERENCE $\frac{dI(x)}{dx} \rightsquigarrow \frac{I(x+\Delta x)-I(x)}{\Delta x} + O(\Delta x)$
- BACKWARD DIFFERENCE $\frac{dI(x)}{dx} \rightsquigarrow \frac{I(x-\Delta x)-I(x)}{\Delta x} + O(\Delta x)$
- CENTRAL DIFFERENCE $\frac{dI(x)}{dx} \rightsquigarrow \frac{I(x+\Delta x)-I(x-\Delta x)}{2\Delta x} + O([\Delta x]^2)$

The notation $O(\Delta x)$ means the error between the difference and the derivative "is of order Δx"; i.e., there is a constant c such that the $|\text{error}| \leq c|\Delta x|$. While this symbolizes our intuition above, how might we check whether these error bounds make sense? Exactly what does the "leads to" symbol "$\rightsquigarrow$" mean here?

16.3.1 The Taylor Expansion

Suppose the intensity value sampled at position x_0 is $I(x_0)$; to simplify notation we shall assume the coordinates are centered so $x_0 = 0$. We are interested in what can be said about the value at a nearby point. If we allow discontinuities, then it could jump anywhere, so let's limit the discussion to functions that are in some sense smooth. (Part of our goal here is to develop intuition–or a more formal meaning–for the term "smooth.")

If the only information given is the single value, then we could postulate that the function is simply a constant–at value $I(0)$–in the neighborhood. ("The best predictor for tomorrow's weather is today's.") However, if we know the function is changing; e.g., if we know the derivative of the function at that point, we might predict a linear relationship $I(x) = I(0)+c_1x$, where $c_1 = dI(x_0)/dx = I'(0)$, the first derivative of the intensity function evaluated at x_0. Thinking of the constant term as "zeroth-order" we are tempted to write (for x close to 0):

$$I(x) = c_0x^0 + c_1x + c_2x^2 + c_3x^3 + \ldots + c_nx^n \tag{16.4}$$

a POLYNOMIAL in x. Because the highest power is n, we say this is an n^{th}-order polynomial. Such functions are among the simplest in mathematics, because only addition, subtraction, and multiplication are needed to calculate them, and among the most convenient, because they are continuous and differentiable to arbitrary order. (In fact, the derivative of a polynomial is another polynomial!)

We confirm $c_0 = I(0)$ by substituting $x = 0$, and we confirm $c_1 = I'(0)$ by differentiating the polynomial 16.4 and evaluating at $x = 0$. In general we have:

$$c_n = \frac{1}{n!}I^{(n)}$$

where $I^{(n)} = d^nI/dx^n$.

Thus, in the neighborhood of an arbitrary point x_i, we have

- · LINEAR POLYNOMIAL $I_1(x) = I(x_i) + I'(x_i)(x - x_i)$
- · QUADRATIC POLYNOMIAL $I_2(x)I(x_i) + I'(x_i)(x - x_i) + \frac{I''(x_i)}{2}(x - x_i)^2$

and so on. Writing this out we get the TAYLOR FORMULA:

$$I(x) = I(x_i)+I'(x_i)(x-x_i)+\frac{I''(x_i)}{2!}(x-x_i)^2+\frac{I'''(x_i)}{3!}(x-x_i)^3+\ldots+\frac{I^(n)(x_i)}{n!}(x-x_i)^n+\ldots \tag{16.5}$$

Notice that the Taylor expansion suggests a very different view of reconstruction from the sampling theorem. In the sampling view we are given a distribution of values of the function at different points; the goal is to reconstruct the function between those points. In the Taylor view we are given the value of a function and

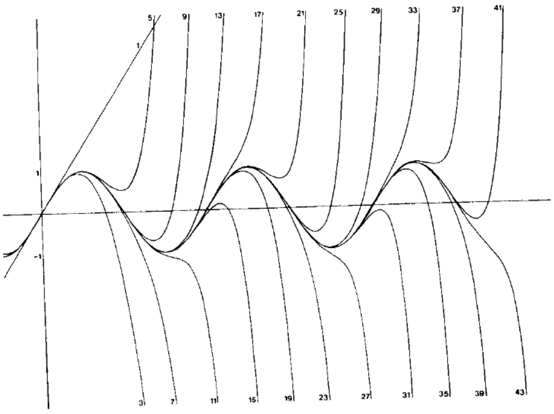


Figure 16.6: Term by term convergence of the Taylor series expansion for sin(x). Note how the series approximates the sin well over an increasingly larger neighborhood as more terms are taken. While this example is highly visual, please remember that Taylor doesn't always converge (this is the definition of an analytic function) and, when it does, it may not converge to the function we want.

several of its derivatives at a single point; the more we know about the derivatives the further we can extend our estimate of the function from that point. This latter form of thinking will become very important in what follows.

To understand how the Taylor series converges consider the expansion for $\sin(x)$:

$$\sin(x) = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + ... \tag{16.6}$$

which, when plotted out term by term illustrates how well the function is approximated near the origin, but not further out (Fig. 16.6. The key point that will be important for us later is that the domain over which the approximation is "good" grows, in a sense, with the number of terms taken in the approximation. But it is important to remember that this is just an approximation and, even if it does converge, it man not converge to the function of interst.

Analytic Functions

In the above discussion there were several implicit assumptions. First, we assumed that an expansion of the form 16.4 was possible, and we assumed that the intensity distribution $I(x)$ was differentiable, as were its derivatives $I'(x), I''(x)$ and so on. Finally we assumed that we could continue this process infinitely often and that, in the end, the full series of terms all converged to a quantity that was meaningful.

In mathematics, functions that admit a convergent Taylor expansion are called ANALYTIC FUNCTIONS.

A particularly important point about the Taylor series is that for physical applications it is typically truncated after n terms:

$$I_n(x) = I(x_i)+I'(x_i)(x-x_i)+\frac{I''(x_i)}{2!}(x-x_i)^2+\frac{I'''(x_i)}{3!}(x-x_i)^3+\ldots+\frac{I^{(}n)(x_i)}{n!}(x-x_i)^n \tag{16.7}$$

which introduces the REMAINDER:

$$I(x) - I_n(x) = R_{n+1}(x) = \frac{I^{(n+1)}(\xi)}{(n+1)!}(x - x_i)^{(n+1)}$$

The good news is that we can write out this remainder but the bad news is the derivative is no longer evaluated at the point x_i but rather at some point $x_i \leq \xi \leq x$. Nevertheless, it is still useful; see Fig.**.

16.3.2 Error Analysis

We now return to considering the forward and backward difference approximations of the derivative. To do this we use Taylor twice:

$$I(x + \Delta x) = I(x) + I'(x)(\Delta x) + \frac{I''(x)}{2!}(\Delta x)^2 + \frac{I'''(x)}{3!}(\Delta x)^3 + \ldots$$

and

$$I(x - \Delta x) = I(x) - I'(x)(\Delta x) + \frac{I''(x)}{2!}(\Delta x)^2 - \frac{I'''(x)}{3!}(\Delta x)^3 + \ldots .$$

Rearranging terms and dropping 2^{nd} and higher-order terms in these two equations yields the forward and backwards difference approximations, respectively. That is,

$$I'(x) = I(x) - I(x - \Delta x) + \text{error} \tag{16.8}$$

with the error terms given by:

$$E = \pm\Delta x \frac{I''(\xi)}{2!}(\Delta x)^2 = O(\Delta x).$$

Noticing the sign difference in the second-order terms above, we can add the forward and backward difference approximations to get the central difference, at which time the second-order terms cancel and we get the third-order remainder term.

Subtracting the forward and backward Taylor expansions, we get the second derivative (Laplacian) approximation:

$$I''(x) = \frac{I(x + \Delta x) - 2I(x) + I(x - \Delta x)}{(\Delta x)^2}$$

with remainder $O((\Delta x)^2)$.

16.3.3 Regularized Derivatives

The central difference form, then, has a lower error bound, which leads us to the "computational template":

$$D(x) = \text{constant}[+1, x, -1]. \tag{16.9}$$

This should be read as assigning the difference between the previous and the next positions as the value for the current (x)-position. (For now we shall ignore the constant.)

Unfortunately (but not surprisingly) adding noise (e.g., in the form of a noise image $\eta(x)$) to the image clouds the issue. $D(I(x) + \eta(x)) = D(I(x)) + D(\eta(x))$, where the latter term can dominate the former. This confirms our earlier observations that taking differences can amplify noise.

One antidote to noise is smoothing, and we have already seen that a smoothing operator $S(x)$ can be defined by the box filter:

$$S(x) = \text{constant}[1, 1, x, 1, 1]. \tag{16.10}$$

which replaces the value at x with the running average of the previous two and the following two values. (Again, the constant is for normalization). Edge enhancement (preceeding detection) can then be formulated as $D(S(I(x)))$, where the smoothing operator first averages the noise away and then the result is differenced.

Most importantly, since the smoothing and the differentiation operators are linear, we have:

$$D(S(I(x))) = (D(S))(I(x))$$

In other words, we can combine the smoothing and the differentiation operators into a single operator (a REGULARIZED DERIVATIVE):

$$D_\sigma(x) = \text{constant}[1, 1, x, -1, -1] \tag{16.11}$$

where σ is a parameter of the smoothing operator.

The box filter is a simple approximation and it hides a fundamental trade-off that has entered the analysis. To articulate it, we switch to the continuous domain and consider the Gaussian filter, which is more appropriate when the noise is drawn from a (Gaussian) distribution centered around a mean value of 0. Optimal noise cancellation is given by the Gaussian function:

$$g_\sigma(x) = \frac{1}{2\pi\sigma^2} \exp^{-\frac{1}{2}\frac{x^2}{\sigma^2}} \tag{16.12}$$

where the parameter σ denotes the spread of the Gaussian.

The derivative of the Gaussian filter is then given by:

$$Dg_\sigma(x) = \frac{-x}{2\pi\sigma^4} \exp^{-\frac{1}{2}\frac{x^2}{\sigma^2}} \tag{16.13}$$

Notice immediately that the spatial support of the smoothing operator and the derivative of the smoothing operator are large, in contrast to the differential operator which is very local. In fact, if the noise at each position is identical and independent, with standard deviation σ, then the noise in the averaged result will be decrease in proportion to $\sigma/\sqrt{n/2}$, where n is the number of pixels we include in the measurement. Clearly, as more pixels are included, n increases, and the noise decreases. However, the more pixels are included in the spatial support of the operator, the more likely it is that it will cross from one structure in the image to another, thereby averaging "edges" rather than noise.

16.3.4 Two-Dimensional Numerical Approximations

To extend the differentiation operations to two-dimensional images, the first derivative becomes the gradient and we have to respect the tradeoff between regularization (smoothing) and differentiation.

The first popular choice is the SOBEL EDGE OPERATOR, which approximates (crudely) the Gaussian weighting:

$$S_x = \begin{array}{|c|c|c|}\hline -1 & 0 & 1 \\ \hline -2 & 0 & 2 \\ \hline -1 & 0 & 1 \\ \hline \end{array}$$

and

$$S_y = \begin{array}{|c|c|c|}\hline -1 & -2 & -1 \\ \hline 0 & 0 & 0 \\ \hline 1 & 2 & 1 \\ \hline \end{array}$$

The magnitude of the gradient at the center point is then given by:

$$||S||_2 = \sqrt{S_x^2 + S_y^2}$$

or for efficency sometimes $S_1 = |S_x| + |S_y|$ is used. The angle of the gradient is

$$S(\theta) = \arctan(\frac{S_y}{S_x}).$$

The Canny operator uses a Gaussian smoothing filter plus some (non-linear) "hysteresis" that keeps the edge going in the same direction provided there are additional gradient responses at that location. (We shall discuss such non-local operations in a subsequent lecture.)

Two standard numerical approximations for the Laplacian are:

$$\nabla = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial x^2} = \begin{array}{|c|c|c|} 1 & 0 & 1 \\ \hline 0 & \text{-4} & 0 \\ \hline 1 & 0 & 1 \\ \hline \end{array} \tag{16.14}$$

or the more accurate:

$$\begin{array}{|c|c|c|} 1 & 4 & 1 \\ \hline 4 & \text{-20} & 4 \\ \hline 1 & 4 & 1 \\ \hline \end{array}$$

The laplacian of the Gaussian can be computed directly and sampled to the desired accuracy. This operator is sometimes called a "Mexican hat" operator, because of its shape. If is nicely approximated by a DIFFERENCE OF GAUSSIAN operation.

Of course, since the Laplacian is a second derivative, to localize the edge it is necessary to find the zero-crossing and not just the local maxima; see Fig. 16.5.

16.3.5 Examples

In the following series of figures we show the results of applying different "edge" detectors to different images. It is important for you to examine these figures and determine, for yourself, whether you think that they are working well enough to (i) find edges in images; (ii) to support the hierarchy of visual feature extraction; and (iii) whether you would want to start a company based on this level of performance.

16.4 Scale and Blur

Different edges have different profiles from different physical causes. Edges in focus are concisely represented in the image, meaning that they have a very local support; edges out of focus are blurred, which increases their local support and moves their

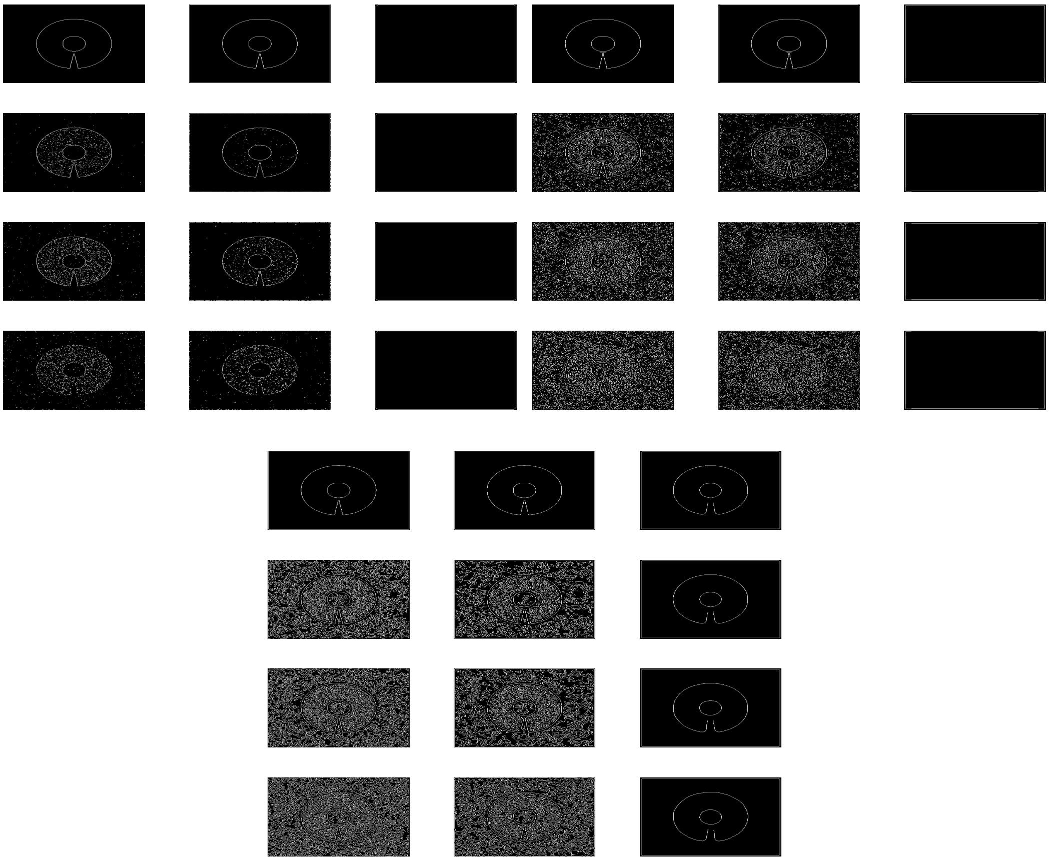


Figure 16.7: TOP, LEFT) The Sobel edge detector on the noisy "cell" image. Note how increasing smoothing (scale) for the edge detector cleans more noise but rounds edges. (TOP, RIGHT The Laplacian of Gaussian edge detector on the noisy "cell" image. Zero crossings are shown. (BOTTOM) The Canny edge detector on the noisy "cell" image. Note how increasing smoothing (scale) for the edge detector cleans more noise but rounds edges.

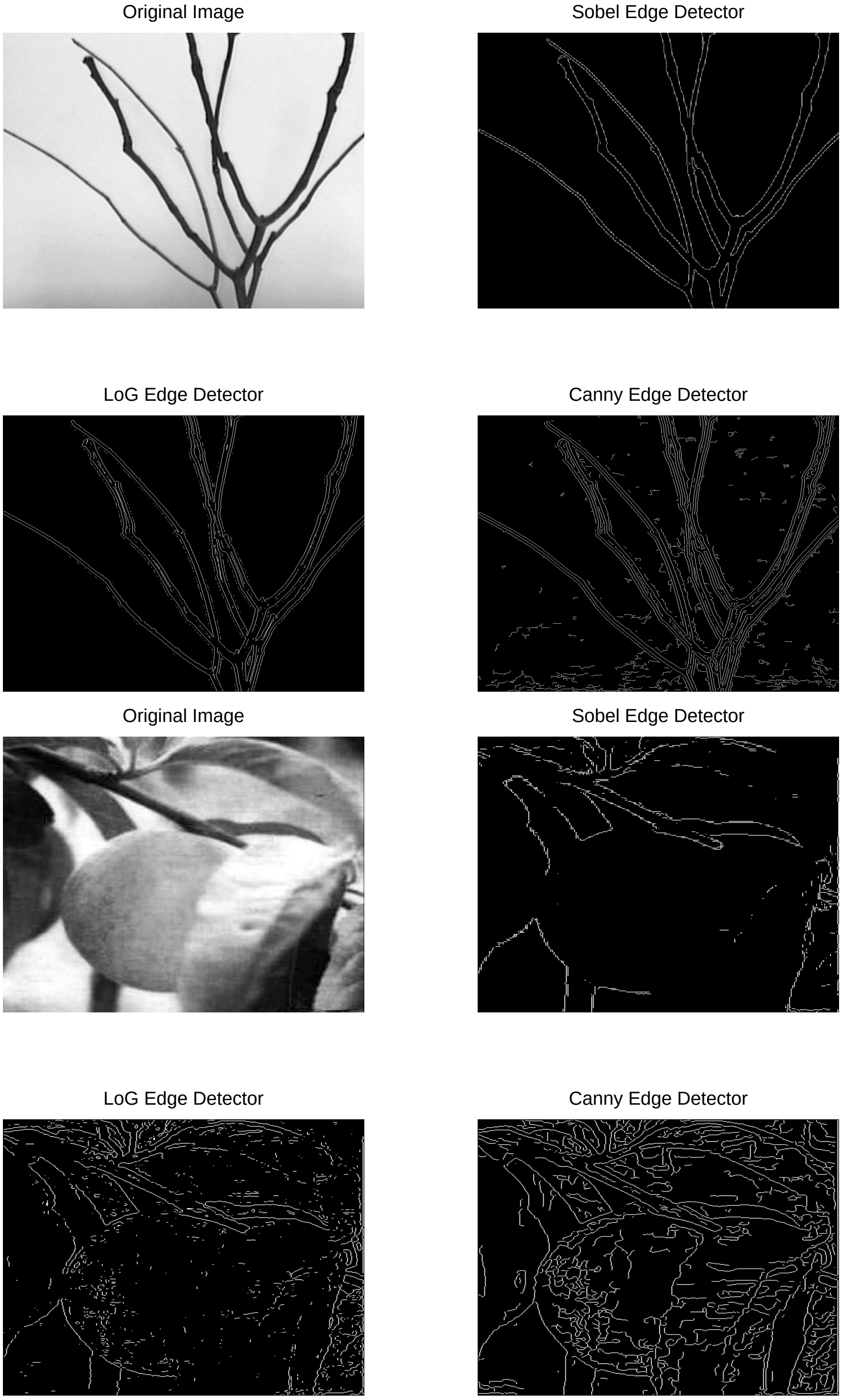


Figure 16.8: Examples of different edge detectors on the twigs image (left) and peach image (right).

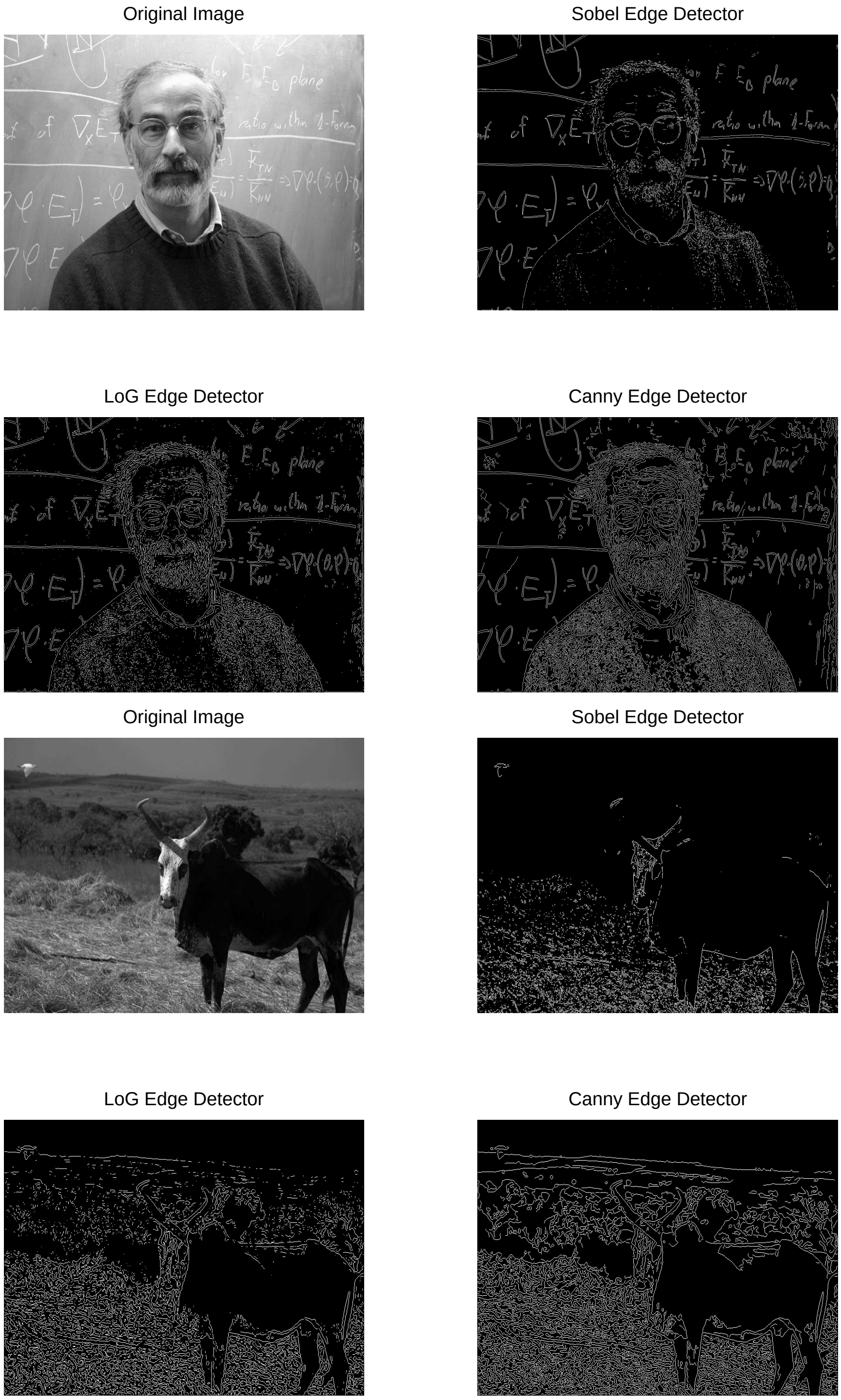


Figure 16.9: Examples of different edge detectors on the Steve image and zebu image.

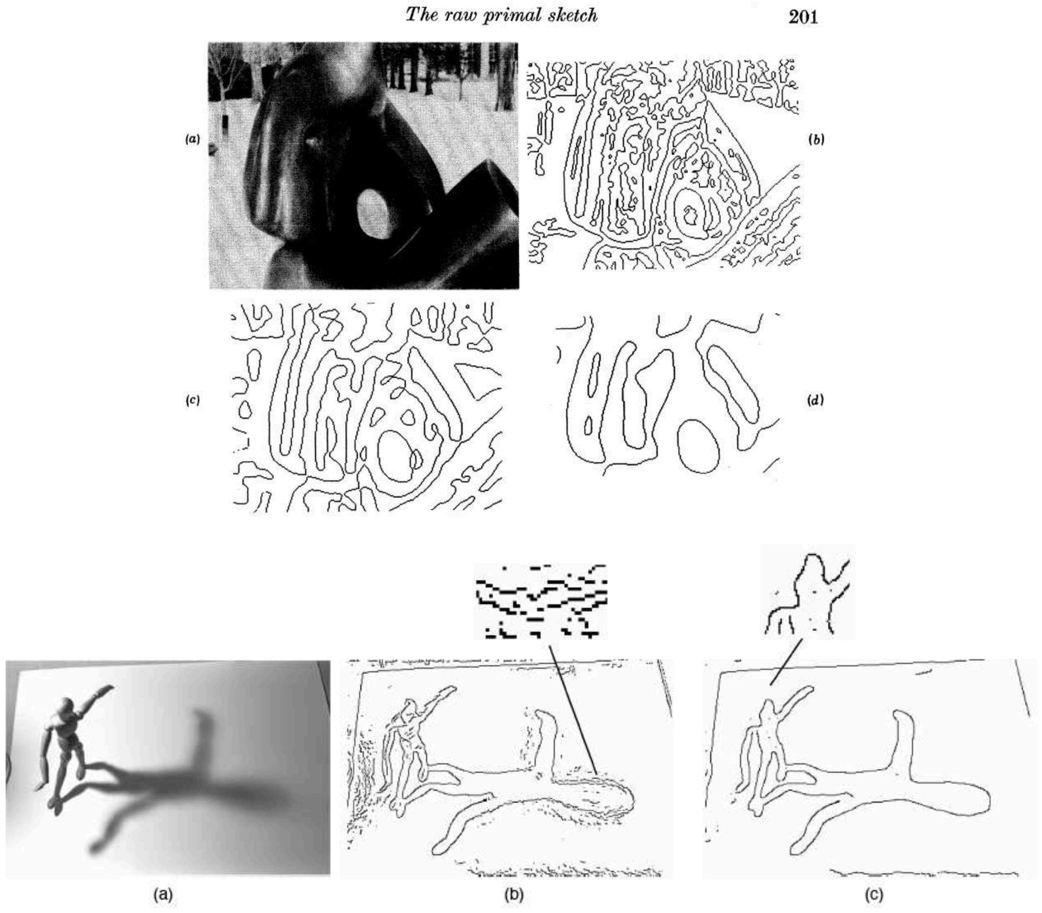


Figure 16.10: (top) Marr-Hildreth zero crossings of the Laplacian of the Gaussian operator at different scales (σ = 6, 12, and 24 pixels in the original). Notice how the different putative edge locations vary with the scale of the operator. Finer scales indicate more edges; coarser scales indicate fewer, blurrier edges. Some of these latter "edges" derive from shading and focus effects. (bottom) An image from Elder showing how scale varies with type of edge. The Canny edge detector is used, first at a fine scale and then at a coarse scale (chosen to match the blurry shadow edge. Which scale do you prefer? Why? Perhaps not a single scale should be used?

frequency representation lower. And shadows resemble blurred edges, because of optics in occluding light sources; See Fig. 16.10.

A second issue is noise. Edges in noisy images will be more difficult to detect, because of the averaging implicit in noise removal (recall adaptation to low-light situations); again, larger receptive fields will imply lower spatial frequencies, more averaging, and less positional accuracy.

There is clearly a relationship between the type of an edge (or, more basically, its physical cause, a point to which we shall return in later lectures); it's representation in spatial frequency; and the size of operators for detecting it.

16.4.1 Scale Space

How long is the coast of Britain? This is Mandlebrot's famous question that led, in applied mathematical sciences, to the notion of fractals, or curves whose dimension is not an integer = 1. For curves such as the coast of Britain or the boundary of a snowflake, the answer depends on the length of the "ruler" used to make the measurement.

While we shall have more to say about definitions of dimension in later lectures, for now we note that this is not unrelated to the idea of edge detection at different scales (ruler sizes). There are two choices.

- *Choose all scales.* Witkin introduces the notion of SCALE SPACE: blur an image by Gaussians of increasing σ; this correspondes to viewing an image at all values of blur, or equivalently, at all scales. Note how information is mixed by the Gaussian at larger scales.

  The implication for edge detection emerges when looking at the zero crossings of $\nabla^2 G$ as a function of σ. The extrema of the first derivative are given by a zero crossing of the second derivative; following these yields curves that are open at the fine scale but closed at the coarse scales. See Fig. 16.11.

- *Choose certain scales* Marr's choice was dictated by an observation from human psychophysics. Recall the perception of spatial frequency gratings (Fig. 8.22); the data for just-detectable contrasts could be fit by postulating that there were 4 spatial frequency "channels" (H. Wilson); it was thought that each "channel" could be a separate projection from retina to V1. Marr speculated that edge detection was completed separately in each "channel" and then those edges that existed in multiple channels survived to perception. Looking at Fig. 16.10(top) it's hard to understand this conclusion.

16.4.2 Minimum Reliable Scale

Computer vision idea: competing issues. We seek the smallest scale possible for localization, but the larger scale for noise reduction. Elder-Zucker idea: use a statis-

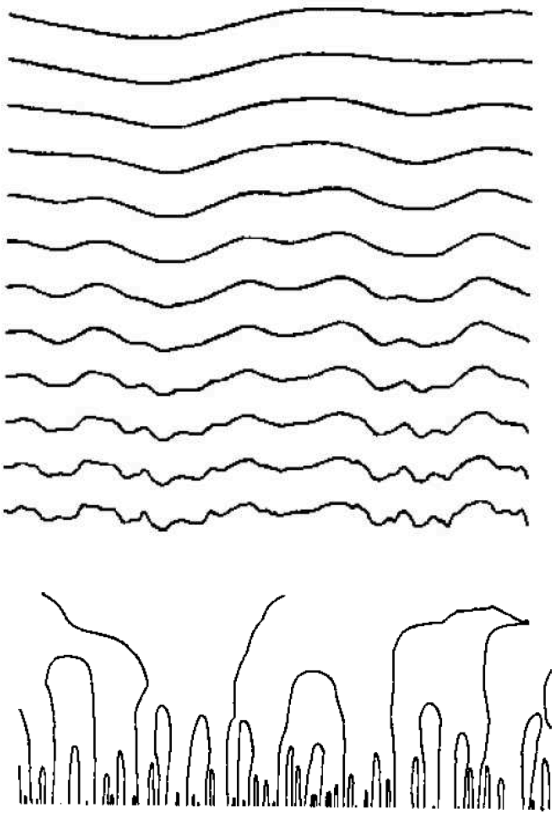


Figure 16.11: Witkin's scale space. (top) In 1-D, a function is convolved against a Gaussian of increasing σ (from bottom to top); notice how smooth it gets. (bottom) Contours of $\nabla^2 G(x;\sigma)$. Notice how they're closed at coarse scales and end at zero's at the finest scale. This permits tracking edges "across scale" to localize them.

tical model for noisy step edges to define a test for the minimum reliable scale. See Fig. 16.13.

16.5 Discussion

There are basic issues around edge detection and whether it does in fact comprise the first level of cortical visual analysis. Here are some of the problems.

The first level of difference is what the columnar architecture implies. Notice that, as with the Sobel operator for example, it was sufficient to evaluate it at a pair of (orthogonal) orientations; the other, intermediate orientations could be interpolated from these. Why are the intermediate orientations made explicit in the columnar functional architecure?

16.5.1 Hierarchy Revisited

There is more to the question of whether simple/complex/hypercomplex cells necessarily form a hierarchy than that the base of this hierarchy–simple cells detect edge boundaries–is apparently false. A more modern review of the neuroscience has shown:

1. Hypercomplex cells with both simple and complex central receptive fields have been found.

2. The distinction between simple and complex cells is often one of degree rather than absolute; many experimentalists believe there is a continuum between these extremes.

3. The underlying circuitry to build simple and complex cell receptive fields may not be ordered as in the hierarchy; rather, similar circuits can provide both directly from layer IVc input.

4. At a further step back, there is the problem that V1 projects to V2 and to the LGN; what is the role of these circuits? Key to the idea of hierarchy is that they form a tree; in this case it is a general graph with loops. Why are the loops present?

But in this lecture the notion of "hierarchy" arises in a different sense as well (scale spaces as discussed here).

16.5.2 Computation vs Communication

Viewing the differential operators as encoders was a position adopted earlier in discussing the retina. The goal for communication was reconstruction of the input information. This was also the position taken in "reading" the neural code for optical flow in H1 neurons of the fly.

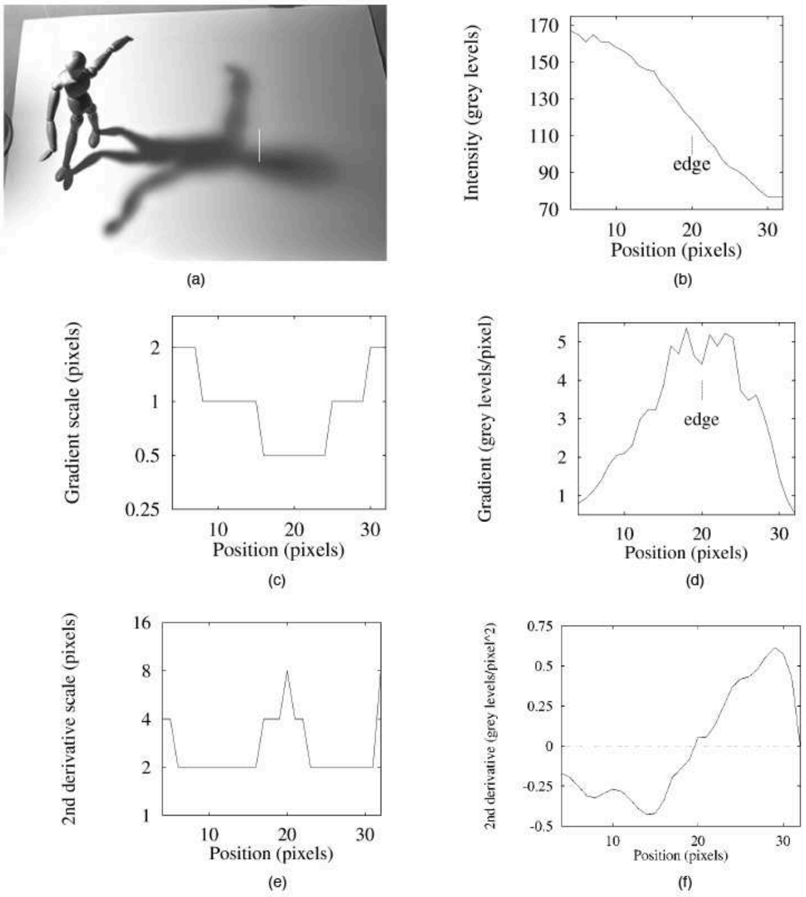


Figure 16.12: Computing edges at the minimum reliable scale. (a) Original image. White line illustrates the set of positions to be considered in the following. (b) Intensities along the white line (defines the x scale). (c) Minimum reliable scale for gradient estimation. (d) Gradient at each position; note there are multiple maxima. which should be chosen? (scale: gray levels/pixel). (e) Second derivative scale. (f) Second derivative (gray levels/pixel^2). Figure from Elder-Z.

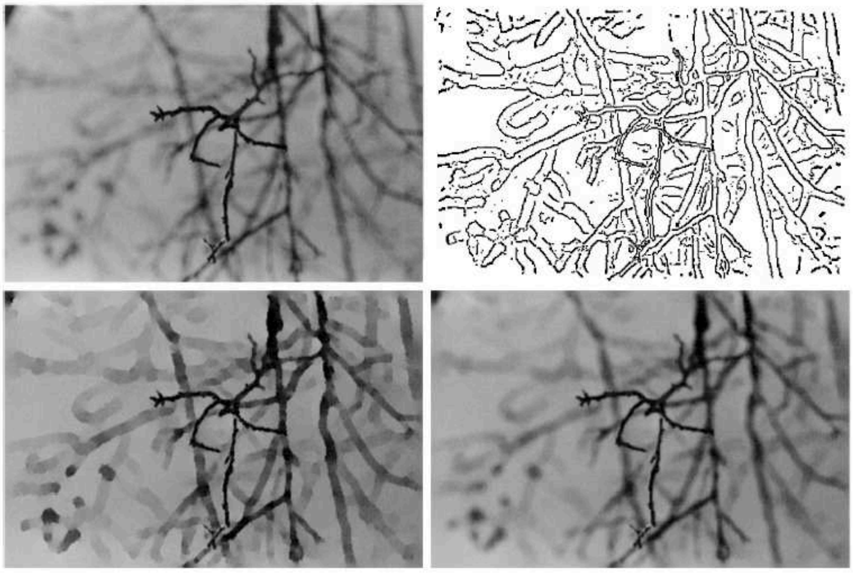


Figure 16.13: (top left) Original image. (top right) Edges found from minimum reliable scale algorithm (see Fig. 16.13). (bottom left) Reconstruction from diffusig the intensities on either side of an edge; notice how unrealistic it looks. (bottom right) Blurring the edges according to minimum reliable scale. Figure from Elder

However edge detection seems different. We take not all values everywhere but only the "strongest" ones. Can this be viewed as a code?

Evidence is that we are capable of "reconstructing" an image from edges; http://static.ne

Of course, but the question is: how well can the image be reconstructed? (Answer: not perfectly, but also not too badly). Key result: knowing edge locations is not enough – also need to know the scale of the edge at each of those points.

But why does the primate want to do image reconstruction? This is not the way to solve tasks such as finding lizards in trees. How can such problems be formulated? This will be an increasingly important change in our perspective. Rather than communication, we shall be thinking computation, and in particular inferences based on image information.

16.6 Conclusions

Perhaps we were naive to think that edge detection was a simple process. After all, not everyone can even draw the "right" edges as well as artists can; compare the Canny result on Paolina (Fig. 16.14 to Fig. 16.15.

The conclusion from these examples is that there's a lot more to edge detection than this simple model would imply - edges have many different causes in the physical

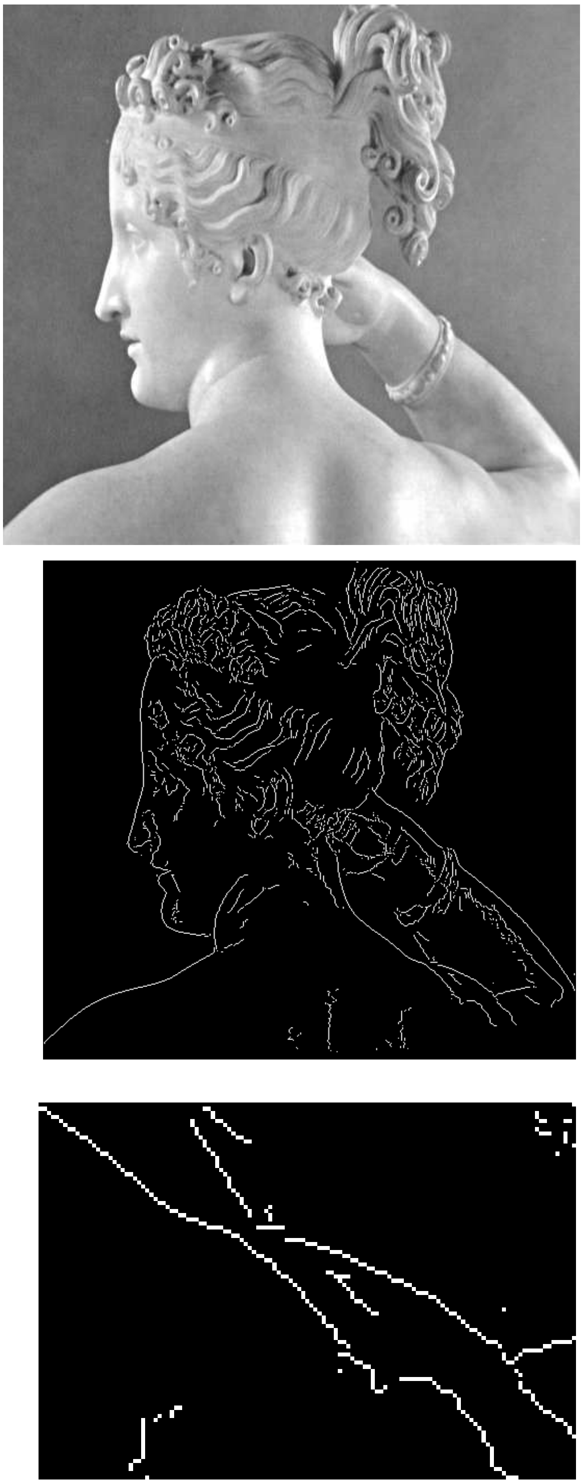


Figure 16.14: The Canny edge detector on an image of the Canova statue of Paolina. What's good about it? What's not? Is the zoom on the arm structure a sensible edge map?

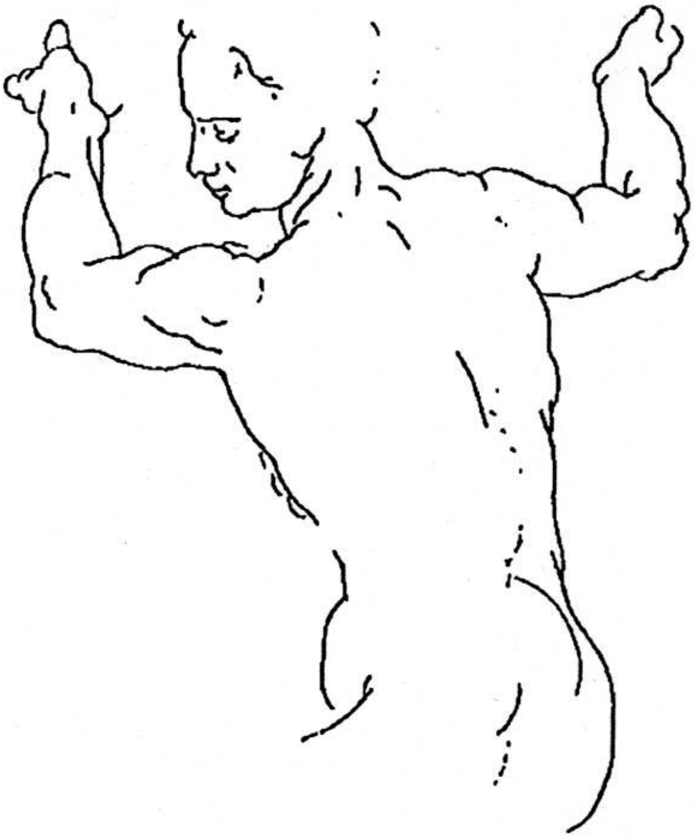


Figure 16.15: Drawings provide an illustration of edge maps as line drawings. Of all the curves that could be drawn on paper, why did Michaelangelo choose these? How close are they to the edge maps we've been computing? What is missing? What did we miss?

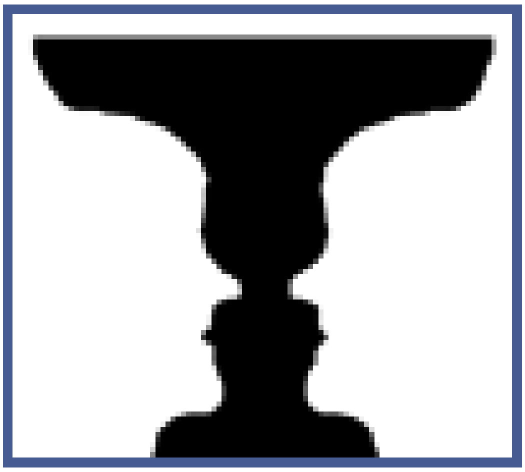


Figure 16.16: Even if we could do edge detection perfectly, there is much much more to figure-ground organization. In this binary image the segmentation is trivial – but what is the figure? Are you sure? This is Rubin's vase, developed around 1915. We shall return to these perceptual organization issues in later lectures.

world (boundaries of objects; shadows; surface markings). Maybe an edge isn't an edge; maybe this is too coarse a grouping and it should be broken up into several different classes of image features.

At a subtle mathematical level, we should also not miss the fact that, in this lecture, yet another expansion idea for functions was introduced. The Taylor series is different from the other expansions we have already encountered – it will turn out to play a very fundamental role in what happens as we move forward in these lectures.

Furthermore, there's more to an edge than the edge of an object; see Fig. 16.16. While in general this will relate to perceptual organization, it does raise the issue of what the intermediate levels might be for feature hierarchies? What might general intermediate-level desscriptors for our visual world be? Perhaps we can answer this by the following reasoning: If so-called edge detectors (namely, cells with simple receptive fields) are not detecting edges, then what are they doing? What might cells with complex receptive fields be measuring? Perhaps our mistake was in trying to guess what an edge is without taking a more careful look at how images are formed, so we can get a clear idea from the physical world what is going on.

Chapter 17

Hierarchies in the Small: The Microstructure of Lines and Edges

Structure within a receptive field; continuity of curves; the limiting operation revisited; logical/linear operators; composing receptive fields; shunting inhibition; the importance of non-linearities as structural guarantees

17.1 Introduction

Our analysis of edge detection revealed the difficulties in taking the pure feature-detection viewpoint: evaluate a linear operator against the image, threshold (or otherwise select maximal values) and mark these as the image coordinates of the scene structure. While this might have a chance of working for insects, the complicated nature of the VS and HS neurons indicated that there is a lot to the *assembly* of elementary units (the local motion detectors) into a global pattern (the "roll" optical flow template) suitable for action. What are the analagous assembly processes for simple cells? What are the elementary processes? And how might they be combined?

In this lecture we start with a more detailed review of the neurons involved in representing orientation information, and build up some ideas for putting them together. The basic question is that, if it were as straightforward a matter to just "add up" the afferent projections from simpler stages to make more complex ones, why is it necessary to have the detailed, layered organization in visual cortex? If linear systems are too simple, what aspect of them can be maintained for more complex systems? This is especially relevant, because as we have seen, many of the cells in visual cortex exhibit receptive fields that resemble linear operators, at least over a significant range. And finally, we ask how well can one do at local "edge" detection?

17.2 Gabor Models for Simple Cells

When we studied Fourier transforms, we noted that different functions had different "spreads" in the spatial and the spatial-frequency domains. (Remember how the spacing between the delta's in the comb function was inverse between the space side and the frequency side; informally, how T = 1/f). In communication systems we worry about different properties of systems in these domains: in space (or time) we worry about how closely spaced pulses (think impulses of light on an optical fibre) can be to one another, while in frequency think how these pulses are spread out by the filtering (blurring) property of the fibre). It is natural when worrying about both of these perspectives to consider them together, i.e., to attempt to minimize the "product" of these "spreads." While telecommunication systems were developing and Shannon's concentration was on information and the sampling theory, Dennis Gabor in England derived a solution to minimizing the "uncertainty" function from the product of localization in space and in frequency (analagous to Heisenberg).

His solution to this minimization is a family of Gabor functions that are sinusoids tempered by a Gaussian envelope:

$$gabor(x, y; \lambda, \theta, \psi, \sigma, \gamma)_{\text{even}} = \exp\left(-\frac{\hat{x}^2 + \gamma^2\hat{y}^2}{2\sigma^2}\right)\cos\left(2\pi\frac{\hat{x}}{\lambda} + \phi\right) \tag{17.1}$$

where $\hat{x} = xcos\theta + ysin\theta$ and $\hat{y} = -xsin\theta + ycos\theta$. Here λ represents wavelength (or inverse of frequence) in the harmonic factor, θ is the orientation of the filter (normal to the cosine stripes), ϕ is the phase (offset of center peak from the center of the Gabor), σ is the s.d. of the Gaussian envelope and γ is the spatial aspect ratio, or how elongated (elliptical) the Gabor support is. An example of such a function is shown in Fig. 17.1. Notice this is a filter with even symmetry.

Gabor's solution also includes the odd-symmetric harmonics:

$$gabor(x, y; \lambda, \theta, \psi, \sigma, \gamma)_{\text{odd}} = \exp\left(-\frac{\hat{x}^2 + \gamma^2\hat{y}^2}{2\sigma^2}\right)\sin\left(2\pi\frac{\hat{x}}{\lambda} + \phi\right) \tag{17.2}$$

Recalling our use of complex numbers in Chap. **??**, these are readily combined into the complex exponential (with the real part given by the cos term and the imaginary part given by the complex term):

$$gabor(x, y; \lambda, \theta, \psi, \sigma, \gamma) = \exp\left(-\frac{\hat{x}^2 + \gamma^2\hat{y}^2}{2\sigma^2}\right)\exp\left(\imath\left(2\pi\frac{\hat{x}}{\lambda} + \phi\right)\right) \tag{17.3}$$

The complex form of the Gabor will be important at the end of the next lecture; for now we shall be concerned with the real versions.

It was observed in the late 1980's that such functions were an excellent fit to simple-cell receptive field profiles; see Fig. 17.2.[1]

[1] J. P. Jones and L. A. Palmer, An evaluation of the two-dimensional Gabor filter model of simple

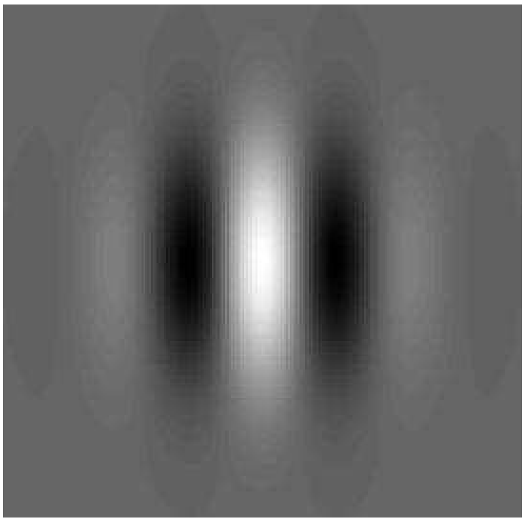


Figure 17.1: Example of a Gabor function. Notice it's orientation, size, and spatial-frequency response (consider the central lobes); notice futher that overall receptive field size is not necessarily in correspondence with spatial frequency selectivity. What might cells with extra side lobes be good for detecting?

17.3 Receptive Fields Redux

Visual cortex is a highly organized arrangement of cells, axons, dendrites and syapses, with enormous potential for visual function (Fig. 17.3. This figure shows a tendency for anatomical organization into vertically-oriented modules running between layers. Apical dendrites mainly from layer V and layer II/III Such modules could arise from development but illustrate the great potential for building complex function around orientation between layers. For a more modern example, see

See Fig. 25.14.

The anatomical complexity leads us to consider what receptive fields look like in a little more detail. In Fig. 17.5 we show two examples.

Our first observation is that receptive fields are functions of time as well as space; and that there is a complementary relationship to the spatial distribution across time. Quite naturally, excitatory zones stimulated by the onset of a stimulus are followed, in time, by inhibitory zones stimulated by stimulus offset. Such relationships provide information about temporal dynamics that shall not concern us in this lecture.

More to the point is the number of subzones across the receptive field. Given that edge detectors needed two subzones (odd symmetry) and line detectors three (even symmetry), what is the role of these extra subzones? Is it just biological variation, or might there be something else to it? The example from Heggelund shows 5 subzones,

receptive fields in cat striate cortex J Neurophysiol 58: 1233-1258, 1987.

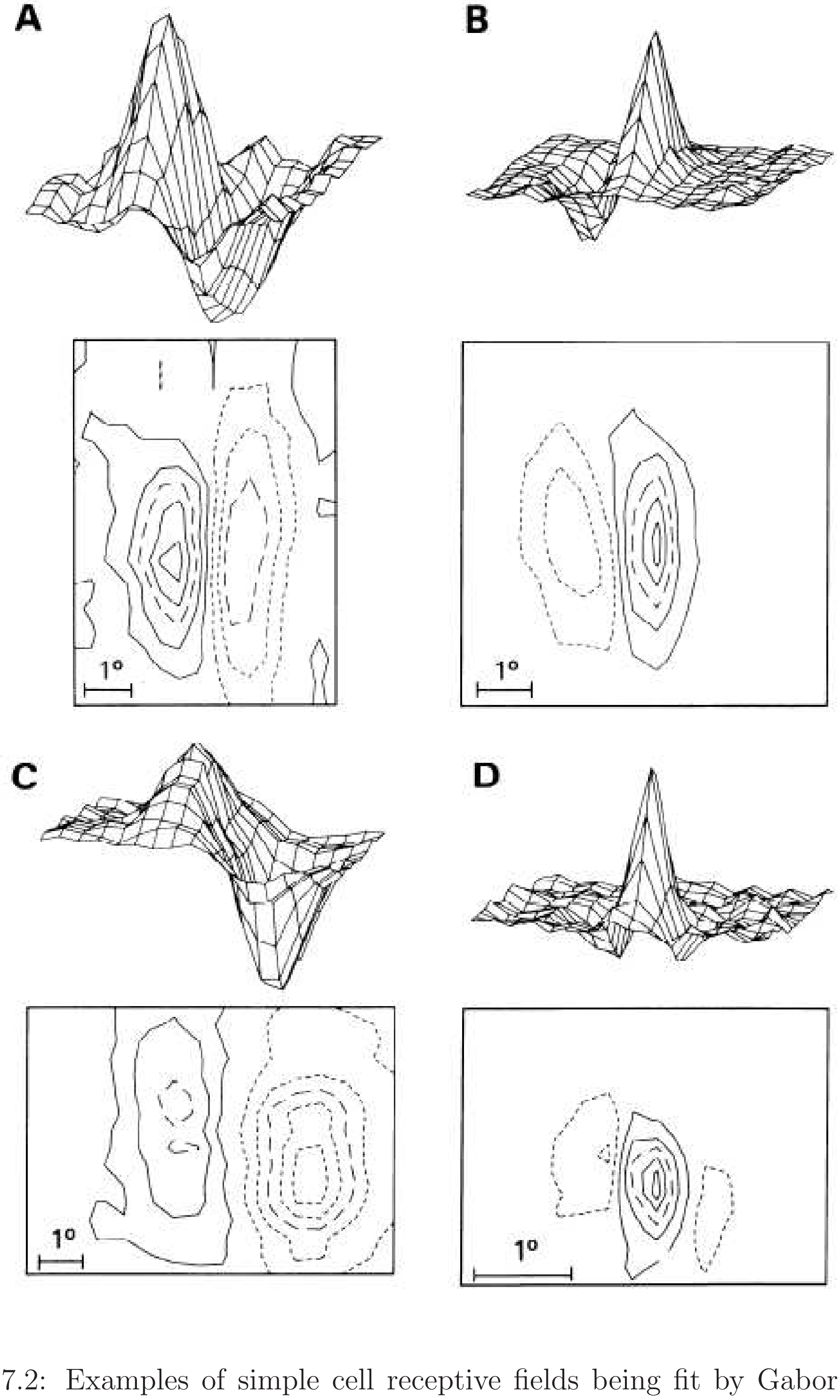


Figure 17.2: Examples of simple cell receptive fields being fit by Gabor functions. While sometimes the fit is excellent, sometimes there is some skew, etc. Figure from Palmer.

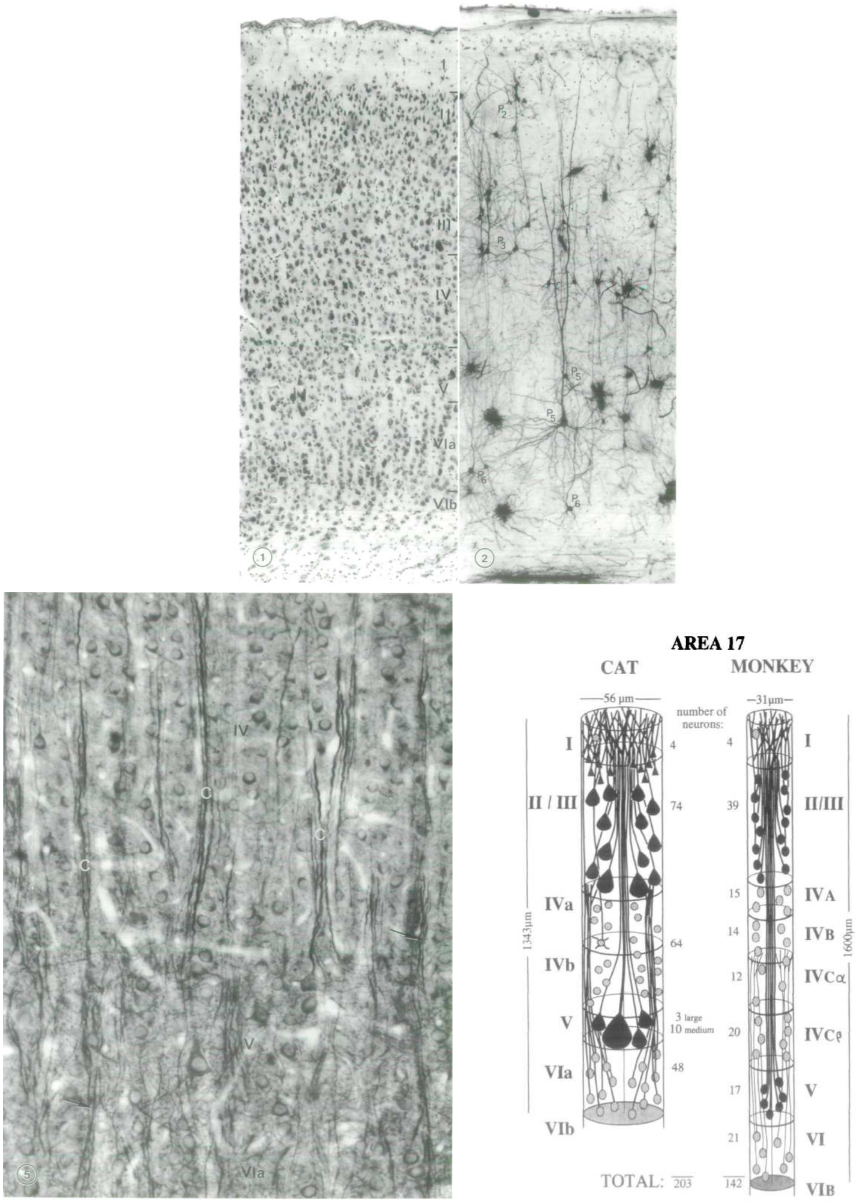


Figure 17.3: Visual cortex consists of a complex arrangement of cells, with rich interactions within and between layers. Here we concentrate on the vertically-oriented modules structured by apical dendrites. (top left) Nissl-stained section of cat primary visual cortex, illustrating cell bodies for pyramidal neurons. (top right) Golgi-Cox preparation for a similar section of cat visual cortex (suprasplenial gyrus). Note the variation in pyramidal cell types and their rich dendritic arborization. (bottom left) MAP2-stained ertical section showing clusters of the apical dendrites from layer V pyramidal neurons. (bottom right) Diagrams of pyramidal cell modules formed around apical dendrites. (Apical dendrites emerge from cells in black.) Given this complexity, what types of networks are being formed? Figs from Peters and Yilmaz

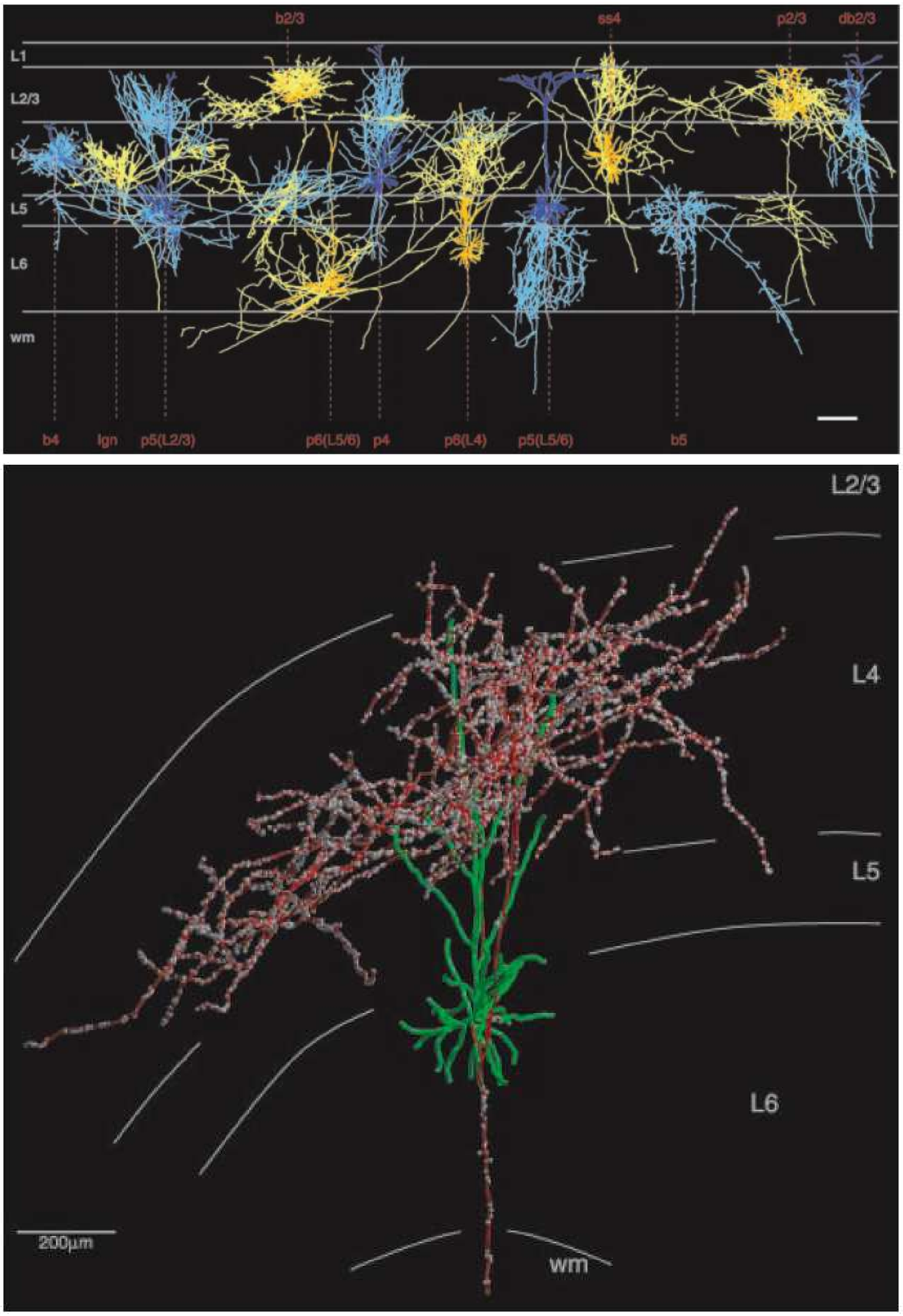


Figure 17.4: (top) Illustration of the different cell types in cat visual cortex. Blue and yellow were used only to aid viewing; the axons for each type are shown in bright blue or bright yellow. Boutons are not shown. Cell types: basket cells (b2/3, b4, b5 in layer 2/3, 4, 5); db: double bouquet; p pyramidal; ss spiny stellate. Coronal view. (BOTTOM) Reconstructed layer 6 pyramidal cell. Axon shown in red, boutons in white, dendrites in green. Cortical layer separations as gray curves. Simple receptive field, monocular, preferred orientation 60 degrees, size: 0.3 x 0.5 degrees. Coronal view. From Binzinger, Douglas, and Martin.

so as we move across the receptive field we see +, -, +, -, +, respectively.

Thus, from a neurobiological perspective, one would expect something more complex than simple linear combinations, as in the original Hubel-Wiesel model (see Sec. **??**).

The good news: linear operators respond properly on what they should see; Fig. 17.6. However, the bad news: linear operators responding when they shouldn't. Further comments about white noise analysis for simple cells; also on spatial freq analysis.

This is structurally a problem, because for a receptive-field operator to respond strongly suggests that there is that type of structure within its receptive field; but this might now be the proper conclusion. Sometimes they respond when there is no structure present! This is the down side of superposition. See Fig. 17.7.

17.4 Logical/Linear Operators

Since there is a sense that higher-level visual representations are constructed on the foundations provided by earlier ones, it is necessary to design feature detectors so that they respond when the appropriate s

Under linear operators many different image configurations map to the same response; how can these be inverted? This is not just a mathematical question: given that a cell is responding, we need to know what it means to properly design or understand the network in which it is participating.

These problems are laid out in Fig. 17.17. A linear operator resembling a simple-cell has the important property that it will respond to properly oriented image lines with the right contrast, but the negative properties that they will respond (to different extents) to improperly oriented and placed lines, as well as to edges. Given that edges bound objects but dark lines are creases or cracks, clearly they are photometrically different.

17.4.1 Design Considerations

Here we turn the tables around. While linear operators were great because they responded when a given structure – for example, an edge passed through the location – as we saw, they also respond to other structures. To make a proper decision we need to understand not only what the signal could come from, but to get as close as possible to what it *did* come from. Desire: an operator that responds if—*and only if*—the proper scene structure is present. This is clearly impossible to achieve, because there may be no image pattern within the image support for that operator (need figure for this). Instead we settle for a more modest goal: to represent a linear operator as a network of smaller operators designed to check local image structure for evidence of these different false-positive situations; if they are detected, then the

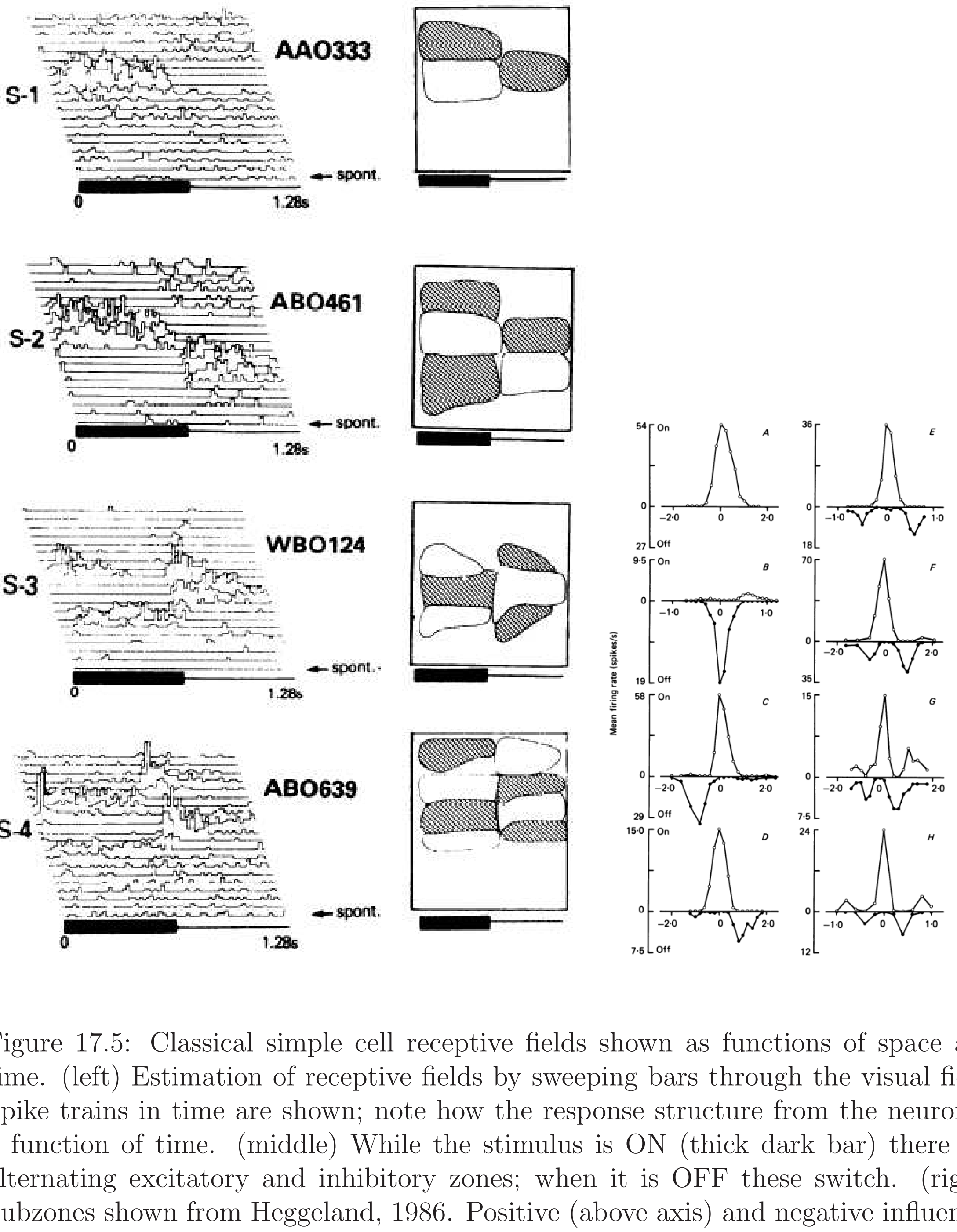


Figure 17.5: Classical simple cell receptive fields shown as functions of space and time. (left) Estimation of receptive fields by sweeping bars through the visual field. Spike trains in time are shown; note how the response structure from the neuron is a function of time. (middle) While the stimulus is ON (thick dark bar) there are alternating excitatory and inhibitory zones; when it is OFF these switch. (right) Subzones shown from Heggeland, 1986. Positive (above axis) and negative influences (below axis) are segregated for display. Note that for some cells (e.g., lower right) there are many subzones.

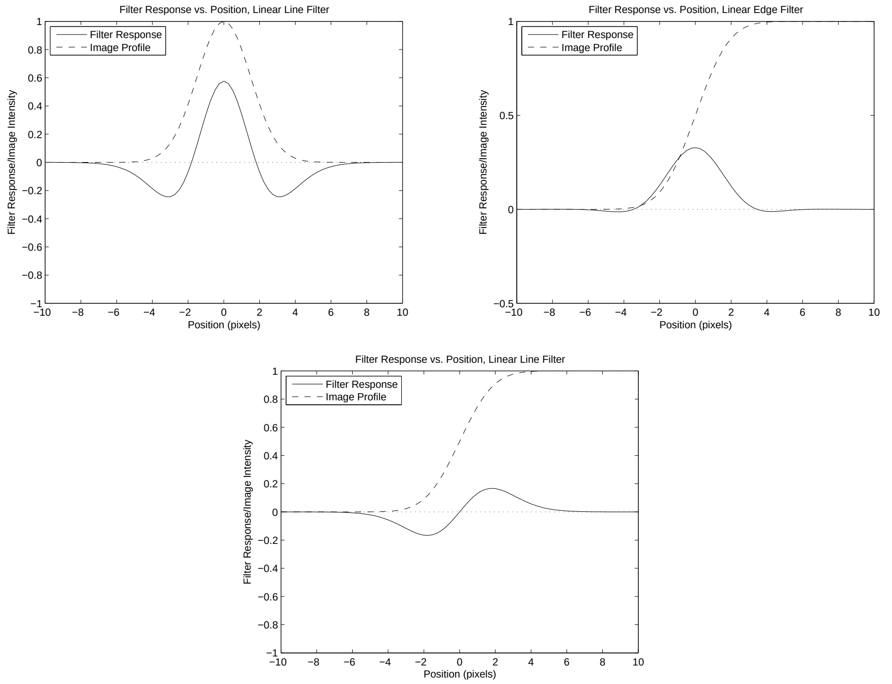


Figure 17.6: top: good news, linear line to line stimulus and linear edge to edge stimulus. bad news: linear line to edge stimulus

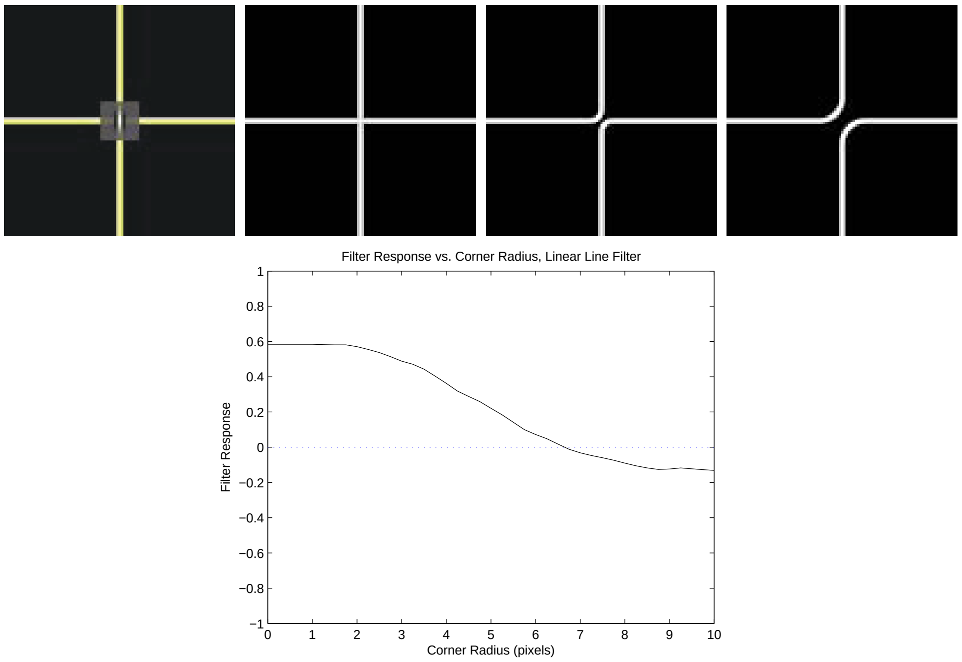


Figure 17.7: Linear operators obey the superposition principle, so they will respond as well to short bars or to interrupted lines within their receptive field. This shows the response to a vertically-oriented filter (about 17 pixels in diameter) at the center position to a pair of crossing lines (left), a pair of interrupted lines connected by an arc of circle closely spaced (diameter = 5 pixels) (middle); and a pair further apart (diameter = 10) (right). Upper left shows the operator to scale. It is essentially an even-symmetric Gabor function with three subzones. Note that there is a strong response at center point even though there is no curve structure passing through it, provided some portion of the contour passes through the central excitatory subzone.

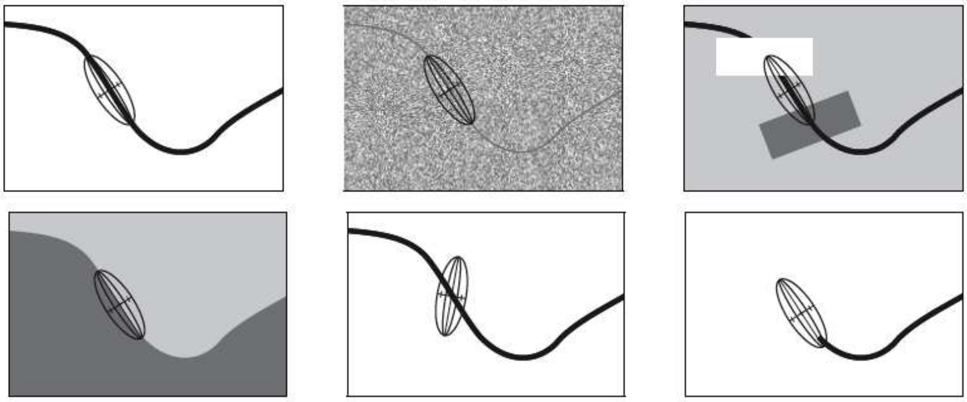


Figure 17.8: (top) Configurations that should give a positive response to a dark line on a light background. They include the ideal case, the noisy case, and the structurally-obscured case. (bottom) Configurations that should not give a positive response to a dark line on a light background. They include edges (which are different from cracks), improperly-oriented lines, and lines which do not pass through the center of the receptive field.

operators response should be vetoed – that is, shut down to 0. Thus there are both image measurements and logical connectives between those measurements.

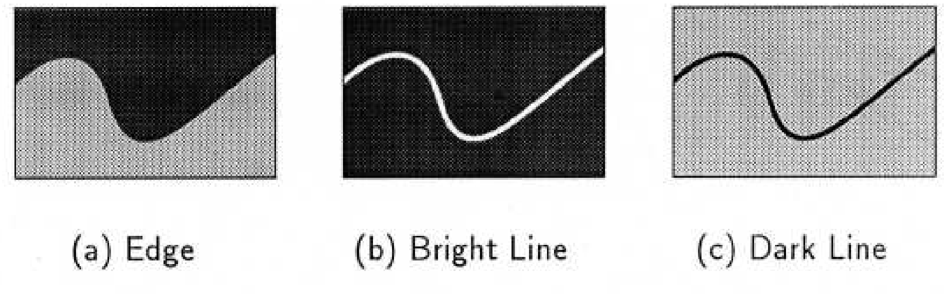
After defining several logical conditions by which the pieces should be fit together, we start with some very small linear operators and combine them into a composite operator provided the conditions are met; if not their response is "vetoed" to zero.

For now we use the terminology (Fig. 17.9):

- *Edges* are the curves which separate lighter and darker areas of an image—the perceived discontinuities in the intensity surface;

- *lines* are those curves which might have been drawn by a pen or pencil; they tend to arise in natural images from surface markings, cracks, or highlights.

An IMAGE CURVE is either of these. Note that linear operators would respond to all of these different configurations, thereby mapping all these structures to a single value. We seek to undo this confusion by exploiting two local, but independent properties that describe image curves: their structure along the tangential and in the normal directions; Fig. 17.10. Tangentially, both lines and edges are projections of space curves; and as such they must be continuous almost everywhere. To capture this type of open set we shall have to look along the open set of point in a neighborhood "before" and "after" They are differentiated by their cross-sectional structure in the image. As we structure them, both are "point" or local properties, so that we can embody their structure in local (non-linear) operators.



(a) Edge (b) Bright Line (c) Dark Line

Figure 17.9: Image curve configurations.

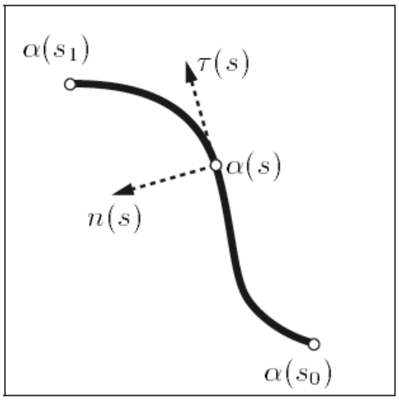


Figure 17.10: The design of image operators should include a component in the tangential direction, that confirm continuity, and a component in the normal direction that confirms photometric conditions.

A related problem will be to localize the ending points of lines, since linear operators blur (extend) them for some fraction of a receptive-field length beyond their physical endpoint. This extension property is further exacerbated by networks of cells supporting one another. (Recall the co-linear extension idea from two lectures back.)

Strategy: decompose the problem into two more managable problems: consider the 2D operator as the product of two 1D operators. Decomposability

Our strategy is as follows. Given the prominence of linear models for simple cell receptive fields, we seek to build them up from more basic pieces, combining them as the evidence permits.

17.4.2 Image Derivatives and Operator Structure

Local extremum in normal direction defines a line

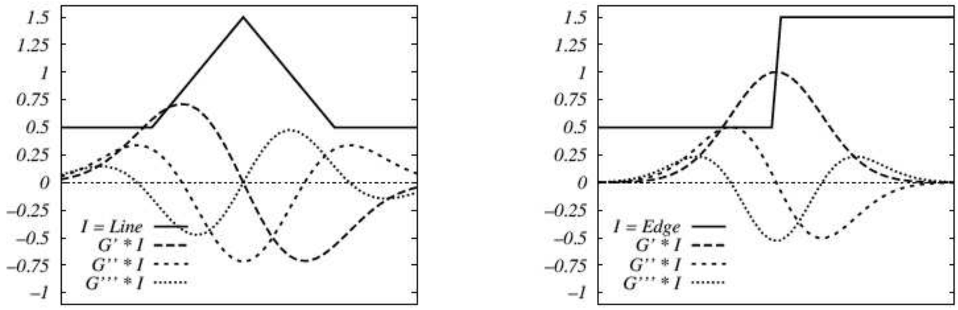


Figure 17.11: Normal (intensity) configurations for a brightline (left) and an edge (right). Various regularized derivatives convolved against the image information are shown. Notation: G is a Gaussian and G' denotes $\frac{dG}{dx}$, G'' denotes $\frac{d^2G}{dx^2}$, etc.

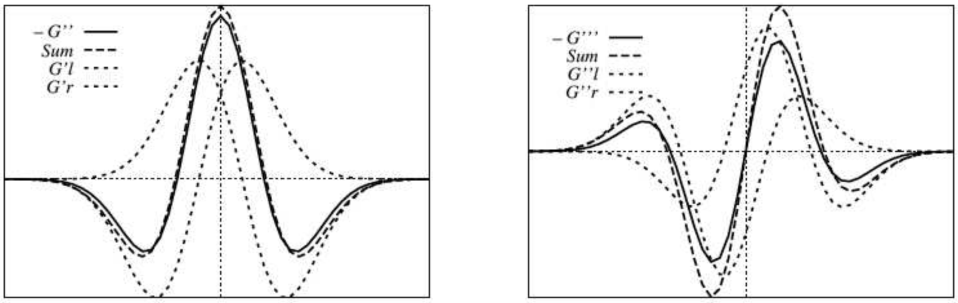


Figure 17.12: Central differences approximate the next higher derivative. (left) Second derivative of a Gaussian is approximated by two displaced first derivatives. (right) Third derivative of a Gaussian is approximated by displaced second derivatives.

17.4.3 Combining Subunits

The Logical/Linear combinators $\wedge_+$ and $\vee_+$ are given by

$$x \wedge_+ y = \begin{cases} x+y, & \text{if } x > 0 \wedge y > 0; \\ y, & \text{if } x > 0 \wedge y \leq 0; \\ x, & \text{if } x \leq 0 \wedge y > 0; \\ x+y, & \text{if } x \leq 0 \wedge y \leq 0. \end{cases} \tag{17.4}$$

and

$$x \vee_+ y = \begin{cases} x+y, & \text{if } x > 0 \wedge y > 0; \\ x, & \text{if } x > 0 \wedge y \leq 0; \\ y, & \text{if } x \leq 0 \wedge y > 0; \\ x+y, & \text{if } x \leq 0 \wedge y \leq 0. \end{cases} \tag{17.5}$$

17.4.4 Results

17.5 Receptive Fields Redux

Now that we're thinking about receptive fields need to consider subzones, what functions they might implement, etc.

types of cells

nature of different circuit types; see Fig. 17.18.

Williams and Shapley have examples that show there are strict phase non-linearities; see paper in bkd.

subzones (Yang Dan)

17.5.1 Shunting Inhibition

17.5.2 Logical Dendritic Trees

dendritic logical trees (Bart Mel)

17.6 Summary

17.7 Notes

iverson paper and thesis

P. Hammond and D. M. Mackay, Influence of luminance gradient reversal on simple cells in feline striate cortex J Physiol April 1983 337 (1) 69-87.

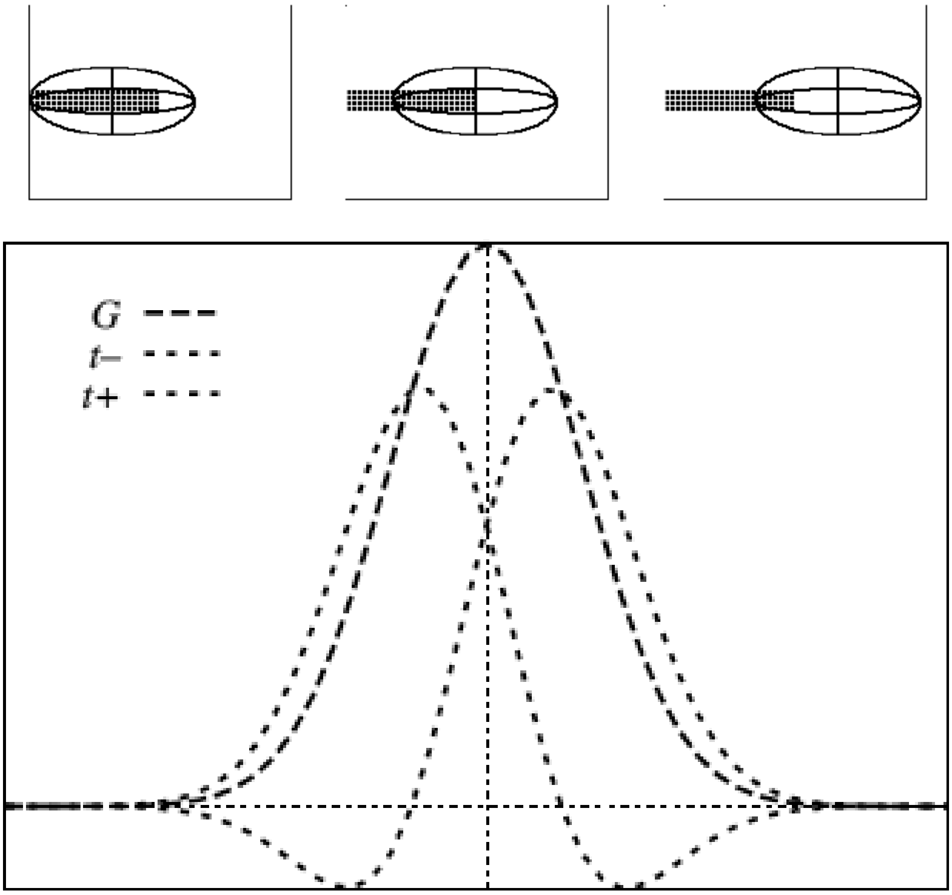


Figure 17.13: Proper placement of an operator near the end of a line. (top) Line extends into both tangential sub-zones. (middle) Line fills one subzone and its endpoint is in the other. (right) Line ends within a subzone. (bottom) End-line stabilizers provide the proper decomposition.

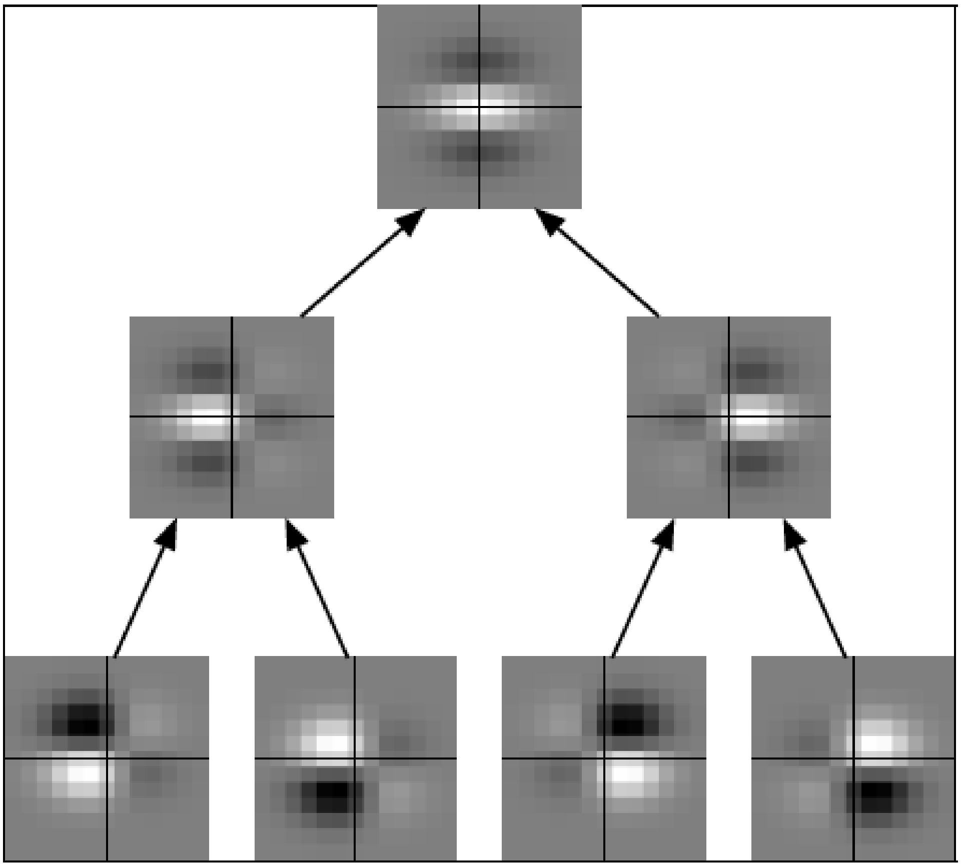


Figure 17.14:

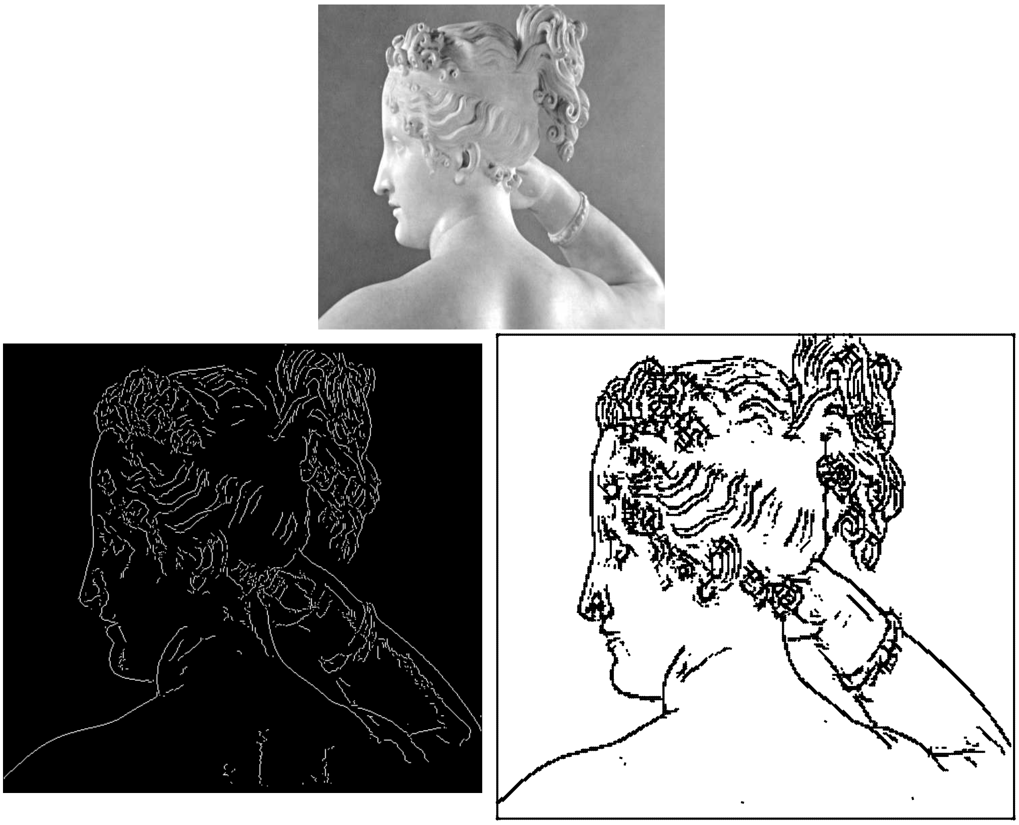


Figure 17.15: Evaluation of the operators on an image of Paolina. (bottom, left) Canny. Notice how the outside boundary is not connected to the inside branch and the multiple traces. (bottom, right) Logical-linear edge (black) and line (gray) operator responses.

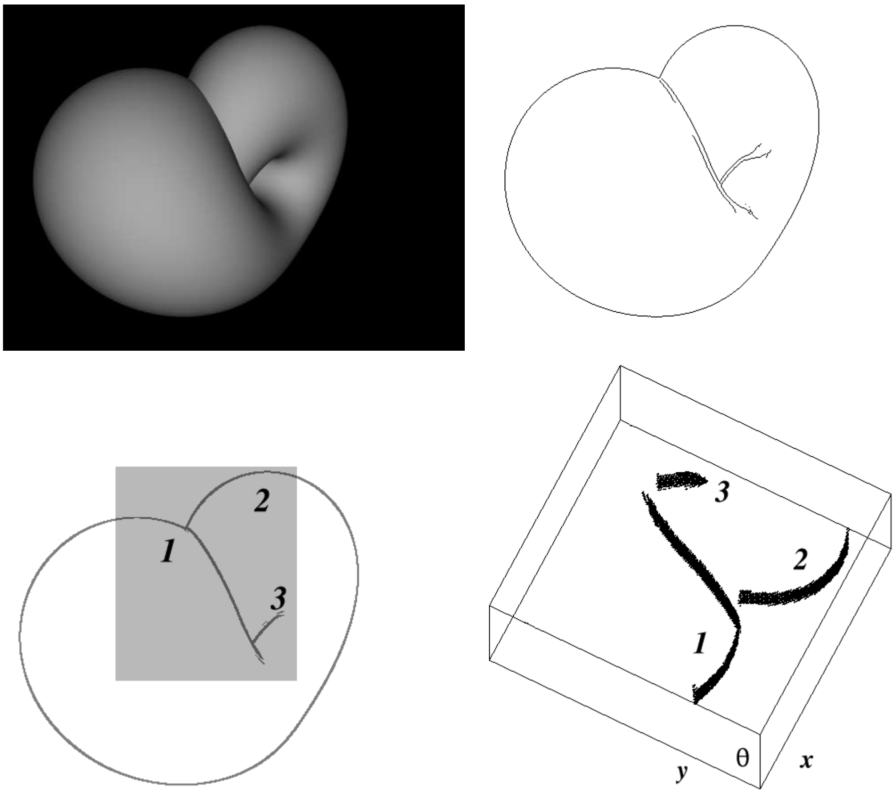


Figure 17.16: Evaluation of the operators on an image of a Klein bottle. (top, right) Canny. Notice how the outside boundary is not connected to the inside branch and the multiple traces. (bottom, left) Logical-linear operator responses. (bottom, right) Logical-linear operator responses lifted into (x, y, θ) coordinates. A perspective on display of the 3D coordinate system is chosen to emphasizie how discontinuities correspond to multiple tangents at the same image coordinate location. Representations of discontinuities such as this are extremely important for understanding how contours are constructed.

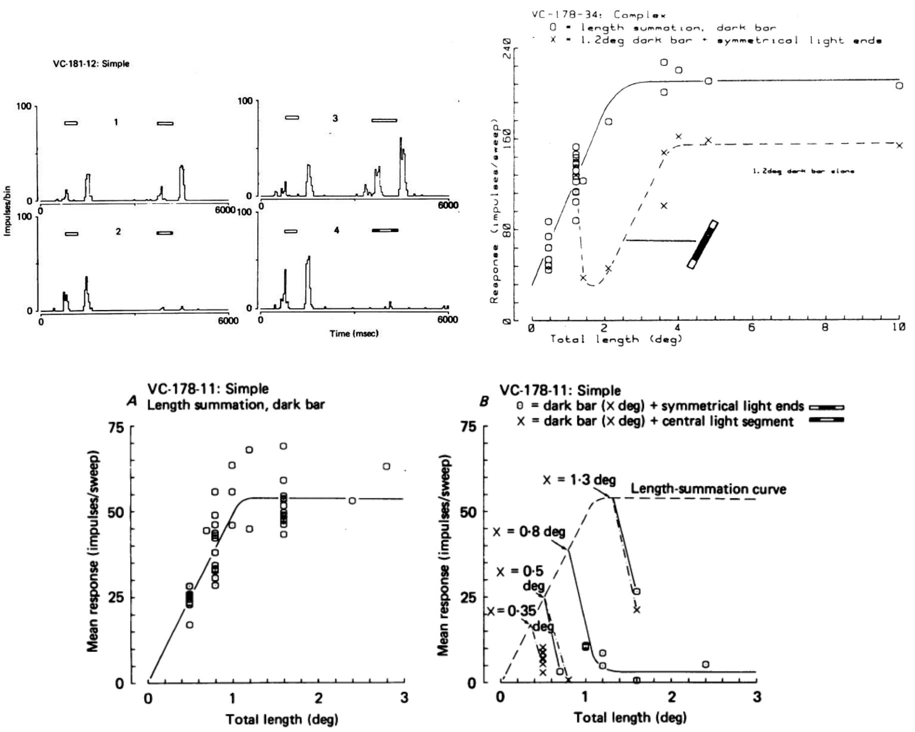


Figure 17.17: Figures from Hammond-MacKay.

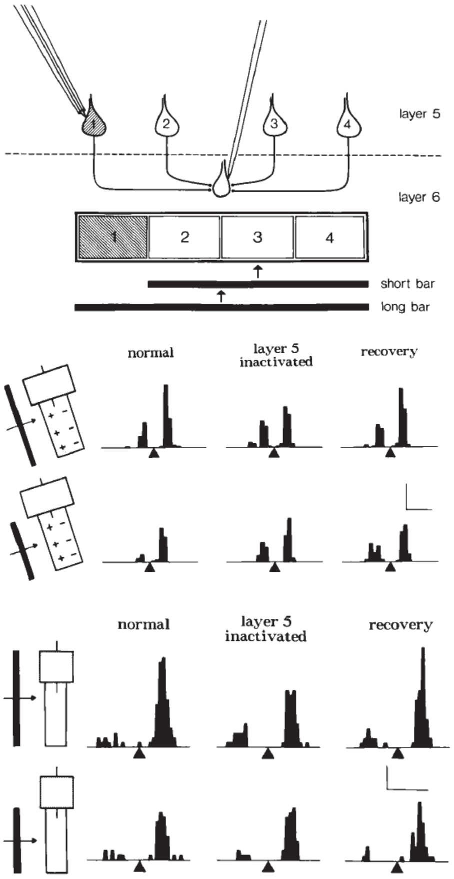


Figure 17.18: Figures from Bolz-gilbert.