Proc. R. Soc. Lond. B **240**, 251–278 (1990)
Printed in Great Britain

Neural network model of visual cortex for determining surface curvature from images of shaded surfaces

By S. R. Lehky† and T. J. Sejnowski‡

Department of Biophysics, Johns Hopkins University, Baltimore, Maryland 21218, *U.S.A.*

(*Communicated by F. H. C. Crick, F.R.S. – Received* 11 *January* 1989 – *Revised* 26 *October* 1989)

The visual system can extract information about shape from the pattern of light and dark surface shading on an object. Very little is known about how this is accomplished. We have used a learning algorithm to construct a neural network model that computes the principal curvatures and orientation of elliptic paraboloids independently of the illumination direction. Our chief finding is that receptive fields developed by units of such model network are surprisingly similar to some found in the visual cortex. It appears that neurons that can make use of the continuous gradations of shading have receptive fields similar to those previously interpreted as dealing with contours (i.e. 'bar' detectors or 'edge' detectors). This study illustrates the difficulty of deducing neuronal function within a network solely from receptive fields. It is also important to consider the pattern of connections a neuron makes with subsequent stages, which we call the 'projective field'.

Introduction

Artists commonly convey a sense of volume and depth to a surface through chiaroscuro, the use of continuous gradations of light and dark. The effectiveness of this technique indicates the importance of shading as one cue for shape, among others, such as contours, texture, and stereopsis. Although information about surface shape is presumably encoded in patterns of neural activity, essentially nothing is known about the process, for in contrast with the extensive experimental literature dealing with contours, little work has been done on shading. As far as we know, there have been no neurophysiological studies, and psychophysical data are sparse (Todd & Mingolla 1983; Mingolla & Todd 1986; Ramachandran 1988*a*, *b*; Bulthoff & Mallot 1988). The major interest on the topic has come from computer vision (Brady 1979; Horn 1986; Ikeuchi & Horn 1981; Pentland 1984; see also the more psychophysically oriented theorizing of Koenderink & Van Doorn, 1980).

To investigate how the visual system may be structured to make use of shading

† Present address: Room 1N-107, Building 9, Laboratory of Neuropsychology, National Institute of Mental Health, Bethesda, MD 20892, U.S.A.

‡ Present address: Computational Neurobiology Laboratory, The Salk Institute, P.O. Box 85800, San Diego, CA 92138, U.S.A.

information, we have employed a learning algorithm to construct a neural network model capable of determining surface curvatures from images of simple geometrical surface. A preliminary description of this work has appeared in Lehky & Sejnowski (1988). In the following sections, details of the network architecture and learning algorithm are given, followed by results, and ending with a consideration of the biological significance of this type of network modelling and its relation to machine-vision approaches for extracting image parameters. Although the psychophysical evidence indicates important interactions between shading and other cues for shape, particularly bounding contours (Ramachandran 1988*a*), this modelling considers shading in isolation. No claim is made that this network is a general solution to problem of shape from shading, but rather that it offers some insights into possible neural organizations for handling the problem.

Goals and design considerations for the network

Our desire was to create a neural network model that would extract principal curvatures and their orientations from images of shaded surfaces. Curvature is defined as the rate of change in the direction of the surface-normal vector as a function of arc length along a surface. Its value depends upon the direction one travels along the surface. Principal curvatures refer to the maximum and minimum curvatures for all possible trajectories through a point on a surface; in general, the principal curvatures will be different at each surface point. This network will determine principal curvatures at only a single point: the centre of the surface in question. By a theorem of differential geometry, the two principal curvatures are always oriented orthogonally (Lord & Wilson 1984).

Curvature was selected because it is a relatively robust indicator of shape. Its magnitude is independent or rotations or translations of the surface, which is not true for surface normals. Furthermore, this shape parameter provides information about qualitative properties of a surface even when its values are not precisely known: just knowing the signs of the principal curvatures can be informative. For example, if both principal curvatures are positive, the surface is convex; if both are negative, it is concave; and if they have opposite signs, the surface is saddle-shaped. It should be kept in mind that extracting surface curvature is a different task than determining the curvature of a one-dimensional edge (Dobbins *et al.* 1987).

A difficulty with using shading is that the pattern of reflected light depends not only upon surface shape, but also upon illumination direction. Somehow our visual system is able to separate these two factors. Accordingly, a goal we set for the model was to determine curvature independently of illumination direction. A further goal was to be able to do this without regard for the position of the surface within the input field of the network. One other confounding factor, surface reflectance, is not considered in this model. All surfaces here have uniform reflectance. Overall, surfaces were defined by the following seven parameters, of which the network was to determine the last three: (i) illumination direction (two parameters); (ii) surface position (two parameters); (iii) principal curvature

magnitudes (two parameters); (iv) principal curvature orientations (one parameter). Because principal curvature orientations, also known as principal directions, are always orthogonal, they are described by just one parameter.

Inputs to the network were images of elliptic paraboloids, which are parabolic in depth and elliptical in cross section. We selected this class of surface because it has no occluding edges, and we were interested in finding what information could be extracted purely from shading. (In saying 'purely from shading' we refer to the restricted range of input images, and not the nature of assumptions built into the network, which will be discussed below.) The surface normal at the centre of the paraboloid was always pointed straight at the observer (i.e. it was perpendicular to the frontoparallel plane), so there was no rotation of the surface in depth. Again, this was done to avoid occluding edges. The surface was illuminated by diffuse light, by which we mean that although illumination came predominantly from a particular direction, there were other components arising from light reflected and scattered about by the surrounding environment. Use of diffuse illumination avoided hard shadow edges, again motivated by our desire to study network responses purely to shading. The surface had matt reflectance properties, without any specular highlights. Details of the illumination and reflectance model are given in Appendix 2.

The input paraboloids can be thought of as approximating small patches of a smooth surface within a complex image. This network therefore models processing in a small portion of the visual field, perhaps the area serviced by a single cortical column. The network would have to be replicated at many locations to cover the visual field, and also replicated at different spatial scales. We envisage all these local networks as converging upon higher-level networks that integrate local curvatures into more general shaped descriptions, although such higher-level networks will not be considered here. Modularization of large networks in this way may be necessary because the time necessary to train a network increases more than linearly as a function of network size (Hinton 1989).

A problem faced by the network was that it is impossible to distinguish a positive curvature from a negative one without knowing illumination direction. For example, the appearance of a convex surface with illumination coming at a tilt of 30° is physically indistinguishable from a concave surface illuminated at a tilt of −30°. (See Gregory (1970) for a striking illustration of curvature sign ambiguity in the picture of a face-mask.) The problem is not an idiosyncrasy of this model, but is intrinsic to the physics of image formation.

To resolve this situation, assumptions about the world had to be built into the network. Specifically, two assumptions were made, one for each of the two principal curvatures. The first placed restrictions on possible illumination directions. The network always interpreted an image as if illumination came from above (light tilt between 0° and 180°). This was sufficient to fix the sign of one of the principal curvatures. The well-known 'crater illusion' suggests that biological visual systems do make this assumption. A picture can be seen as depicting either a crater or a mound, depending on which way it is turned, and the interpretation is always consistent with the implicit assumption that illumination comes from above (Ramachandran 1988*a*).

The second assumption was that both principal curvatures had the same sign. That is, images were always interpreted as either concave or convex, but never saddle-shaped. This assumption is more troublesome than the first, as there does not seem to be any supporting biological evidence for it. It could possibly be eliminated if the network were required to come up with a self-consistent global description over a larger, more complicated surface rather than the simple surface patches used here.

Construction of the network

This section describes aspects of the network that were built into it, and not developed through use of the learning algorithm. This includes defining connectivity within the network, as well as the response properties of the input and output units, but not the hidden units.

Network structure

The network had three layers (figure 1): an input layer, an output layer, and an intermediate, 'hidden' layer. Each unit connected with every unit in the subsequent layer, and could assume a continuous range of activities between 0 and 1. Properties of input and output units were pre-defined for the network, based on the operations we wanted the network to perform, as well as being constrained by biological plausibility. The learning algorithm was then applied to organize the initially random connection strengths between input units and hidden units

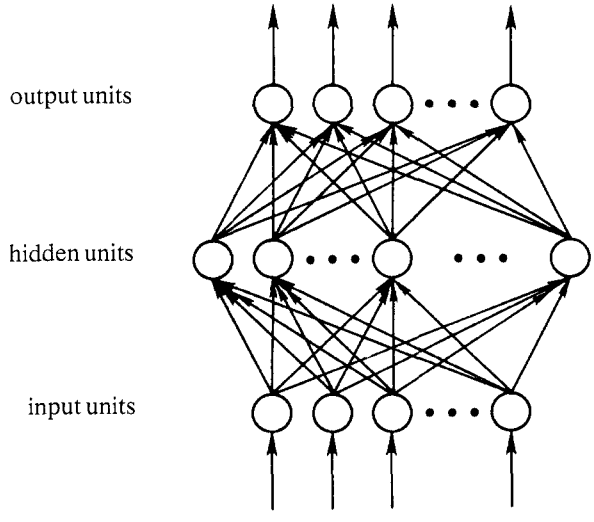

Figure 1. Schematic diagram of the network showing three layers: an input layer (122 units), a middle 'hidden' layer (27 units), and an output layer (24 units). Each unit projected to every unit in the next layer. There were no lateral connections within a layer, and no feedback connections. Activities of units in the input layer are determined by the environment, and activities of other units are determined by linearly summing inputs from the previous layer, weighed by a connection strength, and passing the result through a sigmoid nonlinearity.

(hidden unit receptive fields), as well as connection strengths between hidden units and the output units (hidden unit projective fields).

Input units had circular, centre–surround receptive fields, similar to those of the retina and lateral geniculate nucleus. They were defined mathematically as the Laplacian of a two-dimensional gaussian curve (see Appendix 3). In accord with the biology, there were two classes on input unit: on-centre (shown in figure 2*a*), having excitatory centres and inhibitory surrounds, and off-centre, which had the opposite centre–surround polarity. Although historically the on- and off-centre terminology is based on certain temporal features of biological cells, its use here does not imply any such temporal properties for the model units.

Input units were organized into two hexagonal arrays, one for on-centre and the other for off-centre units. A hexagonal array was chosen rather than a square one because it has a greater degree of rotational symmetry, which we felt would be more conducive to extracting curvature orientations. Biologically, there is too much scatter in retinal ganglion cells to easily classify the sampling lattice (Wassle *et al.* 1981). The two input arrays were superimposed, so that each point on the image was sampled by both types of unit. Each array had 61 units for a total of 122 units in the input layer. This is the minimum that we felt would give a sufficient image sampling density, although the point was not examined systematically. (The specific number 61 happens to come out evenly on a hexagonal grid.) Splitting input units into these two sets prevented them from having negative activities, which would have been unbiological although compatible with the learning algorithm.

Output units had two-dimensional tuning curves that were functions of both the magnitude and the orientation of the principal curvatures (figure 2*c*). The equation defining the output responses was

$$R(M, O) = A(M)\,B(O), \tag{1}$$

where $A(M)$ was a log-normal function of curvature magnitude, M (i.e. gaussian shaped on a logarithmic axis), and $B(O)$ was a gaussian function of curvature orientation O, as detailed in Appendix 3. This type of multidimensional response is typical of cortical cells tuned for parameters such as colour, contour orientation, motion sensitivity, and disparity, although neurons responding to aspects of surface curvature have not been demonstrated. Each output unit in the network had its peak response at a different point in the parameter space (i.e. a different pair of magnitude and orientation values). It should be made clear that these output unit properties were defined as such, and did not arise as a result of training.

A difficulty with units having non-monotonic tuning curves is that multiple abscissa values will produce the same activity. For the two-dimensional case considered here, there are infinitely many combinations of curvature magnitude and curvature orientation that give identical responses.

A way to resolve this degeneracy is to have parameter values represented in a distributed fashion, by the pattern of activity in population of broadly tuned units having different, but overlapping, tuning curves. This network used a distributed representation for the output parameters, as will be described below; the importance of this concept is stressed here. The most familiar example of distributed

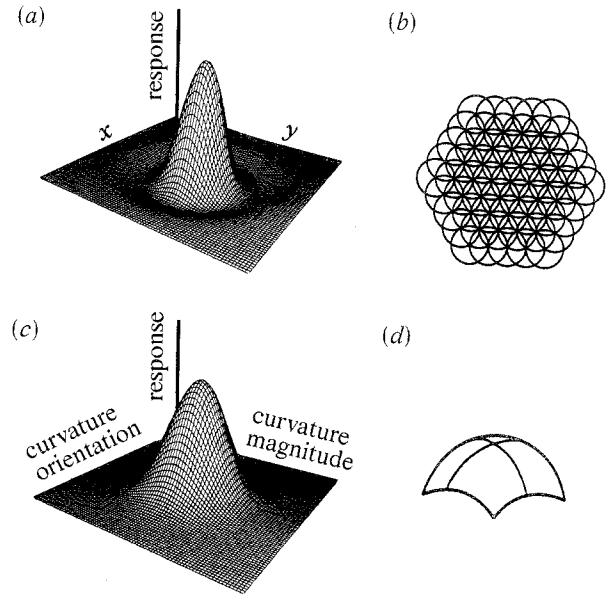

FIGURE 2. (*a*) Receptive field of an input unit. This is the Laplacian of a two-dimensional gaussian function, which provides a circular centre–surround organization as found in the retina and lateral geniculate nucleus. The figure shows an on-centre unit, having an excitatory centre and inhibitory surround. The network also include off-centre units, with excitatory and inhibitory lobes reversed. (*b*) Input units were organized into hexagonal arrays. Circles represent receptive field centres, showing a high degree of overlap. The input image was sampled by on-centre and off-centre arrays of 61 units (*c*) Output units had two-dimensional tuning curves in a parameter space defined by orientation and magnitude of the principal curvatures. Details of the output tuning curve functions are given in Appendix 3. Each output unit had its tuning curve sensitive to different range of curvature magnitudes and orientations. (*d*) Schematic surface showing two principal curvatures (maximum and minimum curvatures) at the centre of the surface.

representation is found in colour vision. Responses of any one of the broadly tuned colour channels is ambiguous, but the joint pattern of activity of all classes of channel allows one to specify colours precisely. Note the economy of this form of encoding: it is possible to form fine discriminations with a small number of coarsely tuned units rather than a large number of finely tuned ones, as was originally pointed out by Helmholtz (1909), and more recently by Hinton *et al.* (1986). Distributed representations have also been used to code motor outputs (Georgopolous *et al* 1986; Lee *et al.* 1988).

Before discussing the output representation in detail, it would be helpful to review the overall organization of the network. This is illustrated in the state diagrams of figure 3, showing the responses (or states) of all units when the network was presented with typical input images.

Magnitude and orientation of the principal curvatures were represented in a distributed fashion in the 4×6 array of output units as follows. The columns correspond to units tuned to six different, but overlapping, orientation bands, with peaks at 0°, 30°, 60°, 90°, 120° and 150° (details in Appendix 3). It is the pattern

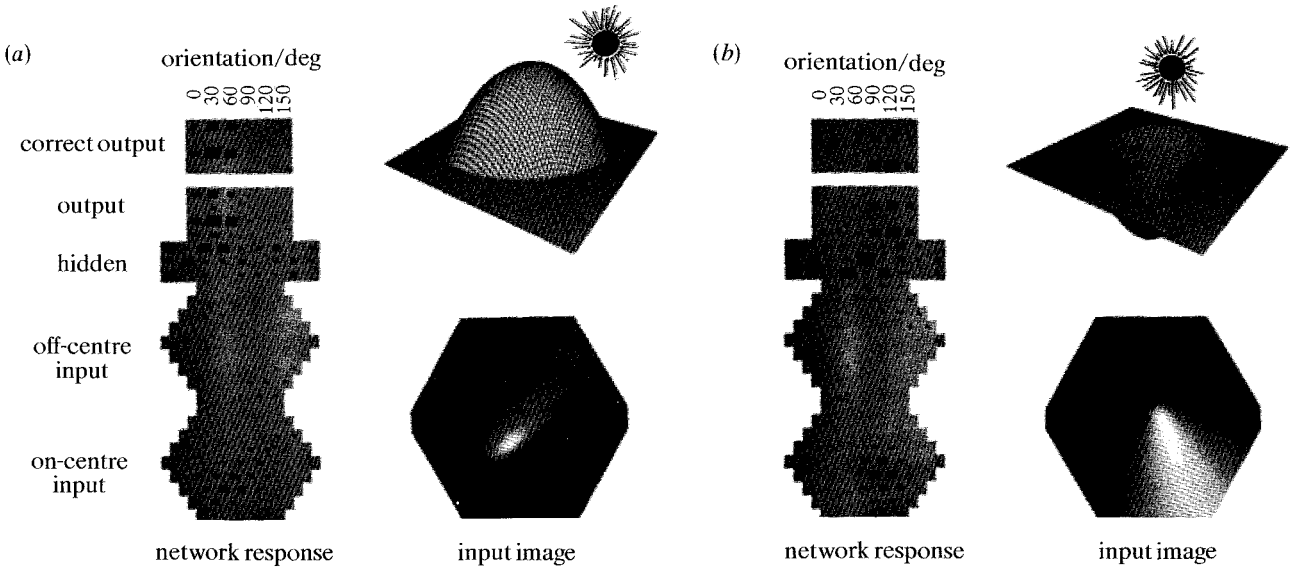

FIGURE 3. Responses of the network to two typical input images, one convex (*a*) and the other concave. (*b*) Double hexagons in the hourglass-shaped figure show responses of 61 on-centre and 61 off-centre input units, calculated by convolving their receptive fields with the image. The area of a black square is proportional to a unit's activity. Converging synaptic inputs from the input layer produced activity in the 27 hidden units, arranged in a 3×9 array above the hexagons. The hidden units in turn projected to the output layer of 24 output units, shown in a 4×6 array at the top. This output should be compared with the 4×6 array at the very top (separated from the rest), showing the correct response for the image. The 4×6 array of output units is arranged as follows. The six columns correspond to different peaks in orientation tuning (0°, 30°, 60°, 90°, 120°, and 150°). Rows correspond to different curvature magnitudes. The top two rows code for positive and negative values of the smaller principal curvature (C_s); the bottom two rows code the same for the larger principal curvature (C_L). (*a*) Response for image of surface with smaller principal curvature equal to 4.20 deg^{-1} and larger principal curvature equal to 10.80 deg^{-1}, with the long axis of the surface rotated 37.8°. The centre of the surface was shifted from the centre of the network input field by 0.27° at an angle of 245.5°. Illumination tilt was 83.40° and slant was 13.70°. (*b*) Response for image of surface with smaller principal curvature equal to -4.10 deg^{-1} and larger principal curvature equal to -8.70 deg^{-1}, with the long axis of the surface rotated 111.4°. The centre of the surface was shifted from the centre of the network input field by 0.10° at an angle of 59.0°. Illumination tilt was 110.70° and slant was 47.90°.

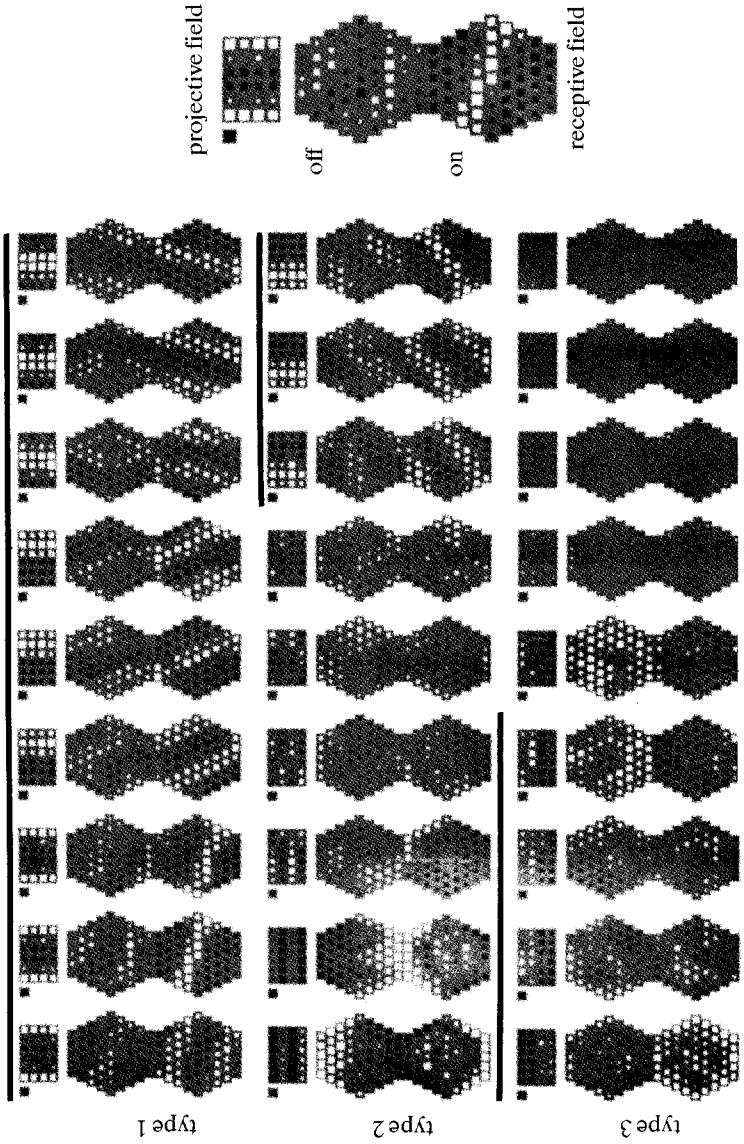

FIGURE 4. Diagram showing connection strengths in the network. Each of the 27 hidden units is represented by one hourglass-shaped icon, showing an input receptive field from on- and off-centre units (double hexagons), and an output projective field (4×6 array at the top). Organization of the 4×6 array is the same as described in figure 3. Excitatory weights are white, inhibitory ones are black, and the area of a square indicates the connection strength. Isolated square at the upper left of each icon indicates the unit's bias (equivalent to a negative threshold). The units are shown grouped together by type based on the organization of their receptive and projective fields (black lines). This grouping was made by hand and was not a property of the model.

of activity in all six that indicates orientation, and not that of any one unit. The curvature orientation indicated by these units was that of the smaller of the two principal curvatures. This implicitly defines the orientation of the other principal curvature, because they are orthogonal. The four rows represent tunings to different curvature magnitudes. The four rows are split into two pairs. The top pair codes the value of the smaller of the two principal curvatures (lying along the long axis of the elliptic paraboloid); the bottom pair codes the larger principal curvature (along short axis of the elliptic paraboloid). Within each pair of rows, the units in the upper row (i.e. rows 1 and 3) responded if the curvature was positive (convex surface) with a peak tuning of $+8$ deg^{-1}. The units in the bottom row in each pair (i.e. rows 2 and 4) responded if the curvature was negative (concave surface) with a peak tuning at -8 deg^{-1}. Splitting the representation of positive and negative parameter values into two sets of output units prevented them from taking on negative activity levels.

Curvature orientation was precisely defined by the joint activity of the population of units tuned to the overlapping orientation bands. However, this was not the case for curvature magnitude, whose representation in this network remains degenerate. The reason is that there is no overlap between the tuning curves for different magnitudes ($+8$ deg^{-1} and -8 deg^{-1}). Because of the non-monotonicity of the tuning curves, a response can correspond to either of two magnitudes, located equidistantly above and below the tuning curve peak. This could be corrected by expand the network to include output units tuned to additional, overlapping ranges of curvature magnitude, equivalent to including units sensitive to different spatial scales.

Applying the learning algorithm

We used the 'back-propagation' learning algorithm to organize the network so as to provide a transform between the retinotopic space of the inputs and the curvature parameter space of the outputs. (Details of the algorithm are given in Appendix 1.) In brief, the algorithm incrementally changed connection strengths while the network was presented with many images of elliptic paraboloids. For each presentation, responses of the input units were propagated up through the network to the output units. Responses of the output units were then compared with the correct output for that image. Based on the difference between actual and correct outputs, connection strengths throughout the network were slightly modified to reduce the error, starting with connections to the output units and then moving back down through the network (hence the name back-propagation). Each unit also had a bias, effectively equivalent to a negative threshold, whose strength was continuously modified throughout the learning process in accord with the algorithm. After thousands if image presentations, the initially random connection strengths organized themselves to provide the correct input–output transfer function.

The training set was 2000 synthetic images of elliptic paraboloid surfaces, described in Appendix 2. This was empirically determined to be a corpus of sufficient size that the network could not memorize each image, but rather would have to extract parameters in a more general manner. Each image had different

values for the seven parameters that defined them, each independently chosen from a uniform random distribution, except for curvature magnitude, which was uniformly distributed on a logarithmic scale. Curvature magnitudes ranged from 2.0 to 32.0 deg^{-1} and also -2.0 deg^{-1} to -32.0 deg^{-1}. Curvature orientation ran between 0.0 and 180.0°. Illumination tilt was also between 0.0 and 180.0°, and illumination slant was between 0.0 and 60°. The centre of the paraboloid surface lay anywhere within the central third of the image.

Results

The performance of the network reached a plateau correlation of close to 0.88 between actual network outputs and the correct outputs after about 40000 presentation. To determine this figure, the correlation coefficient between the actual activities of the 24 output units and the correct activities was calculated for each of the 2000 input images, and then the median value of these 2000 correlation coefficients was found.

The pattern of connection strengths that developed is shown in figure 4, called the weights diagram. Each of the 27 hourglass-shaped figures in figure 4 indicates all the connection strengths associated with a single hidden unit. Within each hourglass icon, two sets of connections are shown. First, there are the connections from all 122 input units to that hidden unit (hidden-unit receptive field). Secondly, there are the connections from the hidden unit to all 24 output units (hidden unit projective field), depicted by the 4×6 array at the top of each icon.

Receptive-field properties

The weights diagram (figure 4) shows that the hidden units formed a variety of receptive-field patterns. As seen in the double hexagon portion of each icon, many hidden units had receptive fields that were orientation-specific. These oriented fields generally had several excitatory and inhibitory lobes, and different units could have the same orientation but have the lobes shifted in phase. This arrangement is similar to that observed in simple cells of cat and monkey visual cortex. which are often fitted with Gabor functions or other similar functions (DeValois *et al.* 1979; Kulikowski & Bishop 1981; Andrews & Pollen 1979). (The earlier studies of Hubel & Wiesel (1962, 1965) focused on the central two or three lobes. which are most prominent.) In addition to units that were orientation-selective. a number of units had receptive fields that were more or less circularly symmetric. An interesting observation is that some hidden units failed to develop significant connection strengths (four units at the end of the third row of figure 4). It seems that only a limited number of hidden units are needed of achieve the task of the network. The extra hidden units undergo 'cell death' (the consequence of a small weight decay term in the learning algorithm (Appendix 1), apparently unable to serve a useful role. Perhaps the number of hidden units required by the network is a measure of the complexity of the task.

Projective fields

Three types of projective fields can be distinguished in figure 4. Type 1 has a vertical organization to the 4×6 array at the top of each icon; type 2 has a

horizontal organization with alternate rows similar; and type 3 also has a horizontal organization, but with adjacent rows similar. Type 1 hidden units appear to provide information about orientation of the principal curvatures, type 2 about their signs (convexity or concavity of the surface) and type 3 about the relative magnitudes of the principal curvatures. Incidentally, during construction of the network the three types of hidden unit always developed in a particular temporal sequence. Convexity–concavity units (type 2) appeared first, followed by curvature orientation unit (type 1) and finally, curvature magnitude units (type 3).

Figure 4 groups together units with similar projective fields. This grouping was done manually for purposes of display, and is not a product of the model. In reality, units developed in random order. Because this network is globally connected, the geometrical position of a unit's 'cell body' is of no significance; in other words, the network has no topography.

Variability among networks

The weights diagram in figure 4 is representative of many learning runs, each started from a different set of random weights. Similar receptive fields always developed, although there were variations in their detailed structure from run to run. In addition, the same three classes of projective field were always found. However, on some runs the classes were more sharply distinguishable, and on others they tended to blend into each other. Also, inhibitory and excitatory weights throughout the network could be reversed, so that the weights from different runs could form a negative image of each other. Finally, there was variation in the number of units that failed to develop strong connections. Despite this variability, the final performance of the network remained quite uniform, with median correlations between actual and correct outputs remaining in the narrow range 0.87–0.90.

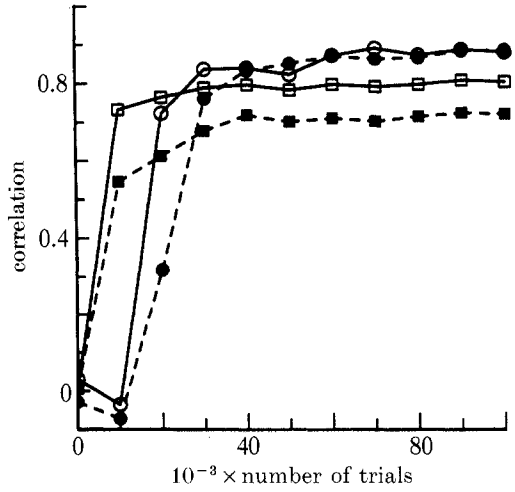

FIGURE 5. Network learning curves, showing correlation between actual and correct responses of the output units as a function of the number of learning trials. Curves for networks with different numbers of hidden units are shown: open squares, 0; filled squares, 3; open circles, 27; filled circles, 40.

Dependence on number of hidden units

The timecourse of network development is shown in the learning curves of figure 5, which plot median correlation between actual and correct outputs as a function of image presentations. It can be seen that including more than about 12 hidden units failed to improve network performance as measured by the correlation coefficient, although additional hidden units did develop connections until about 20 such units were in place, at which point the previously mentioned 'developmental failure' occurred. Although performance failed to improve after about 40000 trials, connection strengths continued to evolve until about 150000–200000 trials. We also tried two other measures of network performance: the root mean square (r.m.s.) error and the vector dot product between actual and correct outputs. Neither seemed to add much information over that provided by correlation (r.m.s. error was 0.10 in the mature network).

One aspect of the learning curves that may seem odd is that network performance *decreased* going from zero to three hidden units. The explanation for this may lie in the total number of 'synapses'. A network having zero hidden units, with input and output units directly connected, has a far greater number of connections than a network with three hidden units (figure 6). Having a very small number of intermediate hidden units causes a bottleneck within the network. Upon examining figure 6, however, we see that even though a network with zero hidden units (a two-layer network) does relatively well, it does not do as well as a three-layer network with an equivalent number of connections. The small difference between two- and three-layer networks may have occurred because the simple images used here did not extend very far beyond the capabilities of a two-layer network, and perhaps a greater advantage for three-layer networks might be apparent when dealing with more complex inputs.

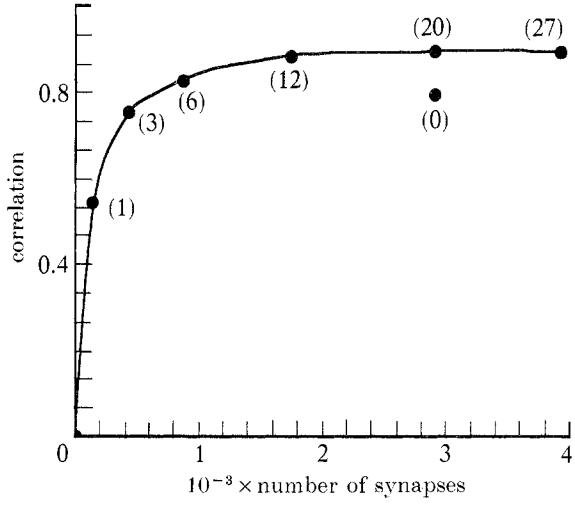

FIGURE 6. Asymptotic performance of the network as a function of the number of synapses. The number of synapses was varied solely by changing the number of hidden units (indicated by numbers in parenthesis). The figure shows that additional synapses improve network performance up to a limit, and that a two-layer network (0 hidden units) performs less well than a three-layer network having the same number of synapses.

Pattern of errors

We tried to identify which aspects of the input gave the network the most trouble. This was done by plotting the 2000 correlation coefficients for all input images as a function of each of the seven parameters defining an image. Although these plots were highly scattered, it appeared that performance was poorest for parameter values at the edges of the ranges over which the network was trained. The most severe problems were due to illumination arising from certain directions. In particular, if illumination came from close to horizontal (light tilt near 0.0° or 180.0°) the network would sometimes reverse curvature signs, interpreting concave surfaces as convex and *vice versa*. Another difficult illumination direction was straight onto the surface (illumination slant close to 0.0°).

Generalization

The ability of the network to produce correct outputs when presented with inputs not in the training set was tested in the following manner. The corpus of 2000 images was randomly divided into two equal sets. The network was trained on one set, and then had its performance tested on the other set. The results are given in figure 7. We also tested the network's ability to extract curvature independently of illumination direction by presenting it with 100 new images with illumination along the long axis of the paraboloid and the same 100 images with illumination orthogonal to the long axis. In both cases median correlation was the same, at 0.88. These results show that the network generalizes well for novel inputs drawn from the same class it was trained upon.

Generalization was poorer when the test set differed more substantially from the training set. When the network was tested with images produced by using unidirectional rather than partly diffuse illumination, so that there were sharp shadow

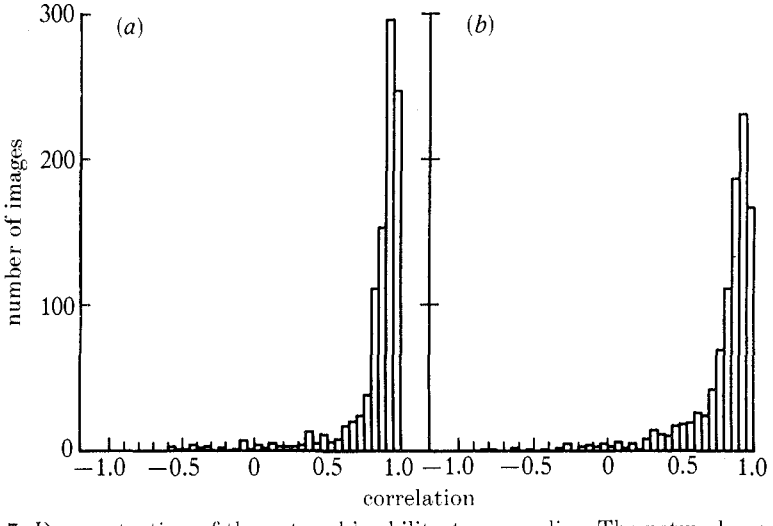

FIGURE 7. Demonstration of the network's ability to generalize. The network was trained on a set of 1000 images, and then tested on another set of 1000 images. (*a*) Distribution of correlation coefficients between actual and correct outputs for the training set (median = 0.91). (*b*) Distribution of correlation coefficients for the test set (median = 0.88).

edges, correlation decreased to 0.78. When tested with surfaces having specular highlights, correlation was 0.70, and when albedo was cut in half, correlation was 0.59. When the network was tested with ellipsoid surfaces (which had occluding edges) after being trained on paraboloids (which did not), it was 0.58. Rotating the paraboloids in depth randomly over the range $\pm 15°$ reduced correlation to 0.79. Asymmetric images produced by both stretching the positive x-axis and contracting the negative x-axis by a factor of $\sqrt{3}$ had a correlation of 0.81. Randomly changing pixel values by ± 0.20 had no effect on correlation, which remained at 0.88; increasing the pixel noise to ± 0.60 produced a correlation of 0.82. The network was relatively immune to such high-frequency noise because it was not matched to the spatial frequency response of the centre–surround input units, whose centres were 64 pixels wide. Adding noise matched to the input filters, by placing a texture of polka-dots of random reflectance on the surface with the same diameter as the receptive-field centres, badly disrupted network performance, producing a correlation of 0.37. In general, despite the loss of accuracy, these results show that some transfer to other types of surface occurred even without training on those surfaces.

Features or filters?

An interesting question is whether the hidden units in this network act as feature detectors or as parameter filters. By a feature detector we mean a unit that responds strongly only when presented with an appropriate and specific stimulus and poorly to all other inputs, in essence an all-or-nothing response. By a parameter filter we mean a unit that responds with a continuous range of activities when presented with various stimuli, which may encode the value of some parameter. To investigate the matter, we looked at the statistical distribution of responses of individual hidden units when presented with the 2000 images. By plotting a histogram of the unit's response levels we hoped to classify the unit. By our criteria, a unit having a unimodal response histogram with a peak at some intermediate level of activity was classified as a parameter filter, and a unit having a bimodal histogram with activities concentrated at either very high or very low levels was a feature detector.

We found examples of both kinds of response. Of the three hidden unit classes, the orientation units (type 1) and the magnitude units (type 3) had unimodal distributions (figure 8*a*), and we classified them as parameter filters. In contrast, the curvature sign units (type 2) invariably had bimodal distributions (figure 8*b*) tending to be fully on or fully off. We interpret type 2 as feature detectors, which discriminate between convexity and concavity of surfaces.

The distinctiveness of the type 2 units from both the type 1 and type 3 units is also seen in 'ablation' experiments, in which individual hidden units were destroyed by setting their connection strengths to zero and then observing the resultant degradation in network performance. Destroying a single type 1 or type 3 hidden unit decreased the median correlation coefficient by an average of 0.03. There were a few cases in which ablating one of these units improved network performance very slightly. Ablating a single type 2 unit had much greater effect, decreasing the median correlation coefficient by an average of 0.16. The greater damage caused by removing a type 2 unit may happen because those units are

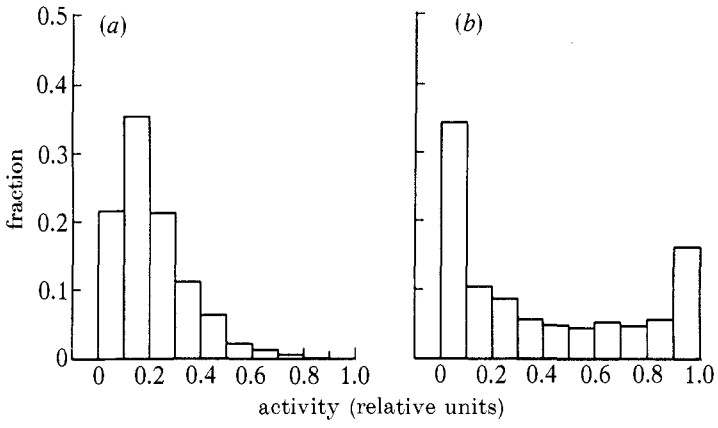

FIGURE 8. Distribution responses of individual hidden units when the network was presented with a set of 2000 images. (*a*) Unimodal distribution typical of a type 1 unit, selective for curvature orientation. Type 3 units, selective for the relative magnitudes of the principal curvatures, also had unimodal distributions. We interpret units with unimodal distributions as parameter filters. (*b*) Bimodal distribution typical of a type 2 unit, selective for principal curvature sign (convexity or concavity). Units having this bimodal response are interpreted as feature detectors.

involved in checking every image presented to the network for concavity or convexity, whereas the determination of orientation, for example, is split among a number of units tuned to different ranges of orientation, and only a fraction of the images fall within the jurisdiction of any one unit.

Simulated neurophysiology

The term 'receptive field' has so far referred to the pattern of excitatory and inhibitory connections arriving from the preceding layer of units. However, the true receptive field is something different: it is not a pattern of synaptic contacts but rather a mapping of a unit's response as a function of stimulus position. This mapping depends not only upon the immediate synaptic inputs to a unit, but also on the filtering properties of units in the preceding layer or layers. To explore true receptive fields of units we conducted simulated neurophysiology, presenting them with 'bars of light'. The bars were varied in position, orientation, width, and length; tuning curves for these parameters were determined for various units. Bar stimuli were chosen because there is an extensive experimental literature based on them, and so responses of model neurons could be compared with those of real ones. Because units in the network responded more vigorously to bar stimuli than to any of the paraboloid images (even though the network had been trained on paraboloids), it was necessary to reduce the 'luminance' of the bars so as not to saturate responses.

Responses of hidden units to bars were easily predictable from the connections they received from the input units. It was possible to form a good estimate of the optimal bar stimulus just by examining the patterns in the double hexagons of each icon in figure 4. The ease in understanding hidden unit responses is not surprising, because the only intermediary between them and the stimulus was the array of centre–surround receptive fields of the input units. Figure 9*a* is an

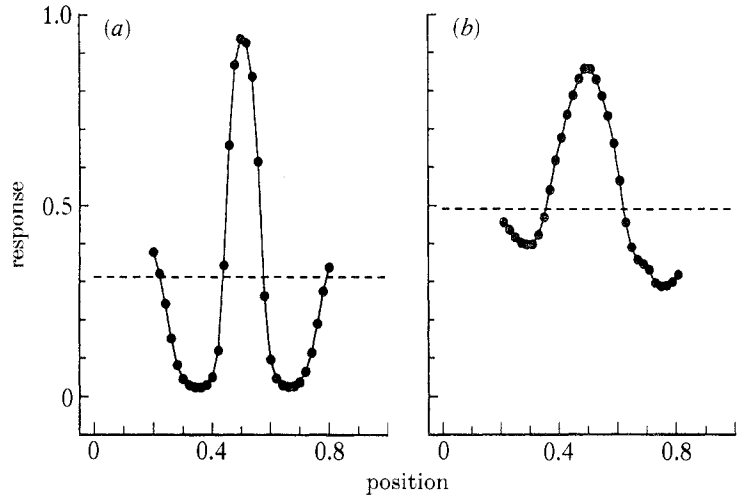

FIGURE 9. Examples of hidden unit responses as a bar stimulus was swept across the input field of the network, for (*a*) a hidden unit and (*b*) an output unit. Dotted lines indicate 'spontaneous' activities when the input was a uniform black field.

example showing the response of one of the oriented hidden units as an optimal bar was swept across its receptive field in a direction perpendicular to the optimal orientation.

The situation was quite different for output units (figure 9*b*). Finding an optimal stimulus took extensive trial and error. Again, this may not be surprising: the response for each output unit is determined by convergent inputs from all 27 hidden units. Despite the complex organization of receptive fields for output units, it was in all cases possible to obtain smooth tuning curves for the various bar parameters. Tunings were generally broader than for hidden units. A feature of some output units was the presence of strong 'end-stopped inhibition', to use the neurophysiological terminology (Gilbert & Wiesel 1979). By this we mean that responses dropped sharply when bar length was extended beyond a certain point. In addition, as the bar was swept across the network in a direction perpendicular to the optimal orientation, the output units showed some degree of spatial delocalization in their responses compared with hidden units, responding over a broader spatial range by a factor of about 1.5–2.0.

Responses of units in the hidden and output layers to bars were reminiscent of some units in visual cortex. Hidden units appeared to have properties somewhat like simple cells, with their oriented, centre–surround organization. Output units were more like some types of complex cell, having a more delocalized response and end-stopped inhibition (although in cortex there are also end-stopped simple cells). What brings the analogy to mind most strongly, however, is again the point made above: hidden units are easy to map out, but output unit responses seem to have subunits that make their responses difficult to understand. In drawing this analogy it should be kept in mind that the relationship between hidden units and output units in the model was strictly hierarchical, because responses of output units were entirely synthesized from the preceding hidden units without any lateral or

feedback connections. The extent to which the visual cortex follows such a hierarchical organization remains controversial (Gilbert 1983; Stone 1983).

Alternative network architectures

We have tested the sensitivity of the receptive and projective fields formed by the network to various changes in its architecture. Testing a limited number of alternatives, we found that the patterns of connections were stubbornly resistant to structural perturbations.

One change was to make all connections from hidden units to output units excitatory, instead of allowing both excitatory inhibitory ones. The rationale for this was that output of real neurons are all of one type. In particular, projections from the lateral geniculate nucleus to the cortex are believed to be all excitatory (Toyoma *et al.* 1977). On training a network with this new architecture, it formed projective and receptive fields that were essentially the same as those in figure 4, apart from the obvious difference that, because there were no inhibitory inputs to hidden units, the black squares in the receptive fields were all blanked out. Performance, as measured by median correlation coefficient, decreased only marginally (by 0.02). Allowing only inhibitory connections between hidden and output units had the same effect. Another configuration we tried was to have only on-centre input units, which could form both excitatory and inhibitory connections. Again the network formed essentially the same pattern of connection as before, and decrease in performance was marginal. A third configuration combined the two above: that is, there were only on-centre inputs and these were allowed to form only excitatory connections. In this case the receptive fields that formed were poorly defined, although the performance of the network remained relatively high, with a median correlation of 0.84.

The ability to eliminate entire classes of connection without affecting performance suggests a high degree of synaptic redundancy within each hidden unit. The possibility remained, however, that these impoverished networks did well, not because there was redundancy within hidden units, but because there was a redundancy of hidden units themselves, which allowed them to compensate for deficits among each other. We examined this by using a network having only three hidden units instead of 27, again constraining connections between hidden units and output units to have positive weights only. In contrast to the network with 27 hidden units, there was now a significant decrease in network performance, with median correlation dropping from 0.76 to 0.65. This result suggests that an excess of hidden units can indeed make up for limitations in each one.

Two other network configurations were tried, both with 27 hidden units. First, we reduced the overlap between receptive fields of input units. This did not affect patterns of connections or network performance. Secondly, we increased the number of input units from 122 to 434 without changing receptive-field overlap, almost doubling the diameter of the input field of the network. The resulting projective fields were the same as before, but there was an interesting variation in the hidden units' receptive fields. Their shapes were identical, but their sizes were normalized to fill the larger size of the network input field. For example, for oriented units the widths of the 'stripes' were double in absolute terms, but

remained the same relative to the increased diameter of the network input field. The performance of this network was the same as before.

Overall, the network appeared resilient to perturbations in its architecture. It is not known if including lateral connections within a layer or feedback connections would lead to a fundamentally different pattern of organization. A generalization of the back-propagation algorithm (Pineda 1987) would allow exploration of networks with these more complex architectures.

Discussion

Biological significance

It is not obvious what this network is doing if one simply examines receptive fields of individual units (figure 4). One might be tempted to classify them into such categories as 'edge' detectors or 'bar' detectors. However, having constructed this network, we know that they are engaged in something entirely different: they are extracting information about surface curvatures from the continuous gradations of shading in an image. Yet it seems unlikely that this interpretation would be among the first that spring to mind. Although this model of course does not prove that cortical cells that have receptive fields similar to those found in the network are engaged in the analysis of shaded surfaces, it does demonstrate that detecting bounding contours is not the only possible function of cells with such receptive fields.

When we say that the units in this network are not acting as bar detectors, that does not mean that they do not respond well to bars. In fact we known that some respond more strongly to bars than to any of the shaded objects that were used to create them. What we mean is that from a computational point of view their response to bars is not the relevant aspect of their behaviour in this network. To label them as bar detectors would give a misleading impression of their functional role.

An important lesson from this modelling is that knowledge of a unit's receptive field is not sufficient to deduce its function. Units with similar receptive fields can have different functions because of different projective fields; indeed, it is possible that a single unit may have multiple functions if it projects to several areas. Understanding the function of a neuron within a network appears to require not only knowledge of the pattern of input connections, which forms its receptive field, but also knowledge of the pattern of output connections, which forms its projective field. This raises questions about conventional interpretations of the receptive fields of real neurons, not only in visual pathways but in other sensory systems as well.

Although responses of units in our model were similar in some respects to those in visual cortex, the detailed manner in which these properties actually arise in cortex could be quite different from that in the network. As an example, oriented responses of cells in visual cortex may arise in part from inhibitory interneurons (Sillito 1975), which we did not include. We feel that the network in its present form should not be taken literally as a model of cortical circuitry, which is much more complex, but rather should be thought of in a more abstract sense as a model

that may capture certain essential features of the cortical representation of information about surfaces. The same representations could be constructed differently in different systems, as indeed the orientation tuning of neurons in different species might also have different origins although they serve the same function.

The distributed output representation we chose may seem unintuitive, because one cannot easily recognize what it represents in the same way that one could if parameter values were made explicit in the activity of a single unit. The advantages of a distributed representation were outlined earlier. If such a representation happens not to be well matched to out ability to extract information from it, that should not be regarded as an argument against it; it is an internal code of the brain and not meant for our perusal. We believe that it is a form of the homuncular fallacy to expect that parameter values are made explicit at some point in the brain. In our view, a value is represented by a pattern of activity, with the output pattern of one network acting as the input pattern for the next, until eventually it may reach a motor output, which again is a pattern of activity because movement is the product of an ensemble of muscles, or the pattern may be stored in memory in some form.

The performance criterion we used, the correlation between actual output and the 'correct' output, indicates that we can use the learning algorithm to construct a network that produces a specified input–output transfer function. This begs the question of the appropriateness of the transfer function we selected, and in particular the appropriateness of our output representation, which is speculative (although based on principles that have some experimental support). Furthermore, a particular transfer function can arise through different algorithms, and our performance measure does not indicate how well the network follows a particular algorithm, in particular the one used biologically. Because none of the relevant biological information is available, network performance cannot be compared directly. Nevertheless, there are some tests that could be done. Psychophysically, the model could be evaluated by seeing whether it fails in the same way that humans do when asked to extract shape parameters from shaded surfaces. Physiologically, it would be interesting to use shaded images to test the responses of visual neurons. We expect that a class of end-stopped complex cells will be sensitive to the local surface curvature over a wide range of illumination directions.

Significance of the back-propagation algorithm

Back-propagation was used purely as a formal technique for constructing a network with a particular transfer function between inputs and outputs. This is not a model of developmental neurobiology, and no claims are made about the biological significance of the process by which the network was created. Nor is this intended as a cognitive model of learning, describing, for instance, a process by which a person may become more adept in extracting particular visual features through repeated exposure to them. That being the case, the learning curves in figure 5 are presented as technical indicators of the performance of the algorithm, and the 'number of trials' for network performance to plateau is not in any sense a psychological prediction. The significance attached to this model is not in the

process by which it was created, but rather in the resulting properties of the mature network.

Many features of the network are non-biological. For example, summation of inputs to a unit is linear in the model, whereas in reality the process can be a highly nonlinear, dependent on the biophysics and geometry of dendrites (Koch & Poggio 1987; Rall & Segev 1987). Furthermore, connections arising from a single model unit can be both excitatory and inhibitory, whereas in real cortical neurons all connections are either one or the other. There are no lateral interactions between units of the same layer in the network, nor are there any feedback loops. Those are both important features of real neural networks. Besides problems in physical organization, there are also those involving the procedures of the learning algorithm. It requires a 'teacher', which provides a standard to which outputs can be compared and adaptively corrected. It also requires and implausible two-step sequential cycle in which responses are first propagated up through the connections of the network, and then information about the output errors propagated back down these same connections (although there is a version of back-propagation that segregates the two streams of information into separate sets of units (Parker 1985).

In response, we have already mentioned that the model network appears robust to various changes that would make it more physiological. How much biological detail this class of model can support before becoming unwieldy remains to be seen. Concerning the existence of a 'teacher', there are other algorithms that may be more realistic from a developmental point of view, for example the reinforcement algorithms (such as that Barto *et al.* (1983)) in which the network receives feedback about whether its response is appropriate or not without being told the nature of the error, as well as the entirely unsupervised algorithms (Linsker 1986; Kohonen 1984). On the other hand, those algorithms are not as suitable as the supervised algorithm used here when one wants to construct a network with particular characteristics.

Finally, we wish to emphasize that blind application of a learning algorithm is no substitute for thoughtful consideration of the problem at hand. We believe that careful selection of the input and output representations for the network, based on knowledge of biological visual systems, were essential for creating conditions that allowed the algorithm to generate interesting results. A good strategy for constructing model neural networks may be to incorporate as much knowledge as possible into its initial structure, and then use the algorithm to extend one's reach by filling in gaps and details.

Relation to machine vision models

There are several approaches to extracting shape from shaded surfaces in the machine vision literature (see for example, Ikeuchi & Horn (1981); Pentland (1984) but for the purposes of comparison here they are all models of the same class in that all formulate explicit rules for extracting shape, rules expressed by sets of mathematical equations. On the other hand, the approach used here differs fundamentally. In the network model, rules, or algorithms are never explicitly stated, but rather are implicit within thousands of connections interacting within a nonlinear system.

This raises the question of whether the algorithm implicit within the network can in principle be reduced to a simple set of equations comparable to the ones in machine vision. First of all, there is no guarantee that this can be done, and it is very possible that there is no simpler description of the network than itself. Furthermore, even if a compact algorithmic description exists, there is no principled way of finding it, and the difficulty increases rapidly with the size of the network. These questions remain as research issues within neural network modelling, and at present much of the work in the field, including that reported here, should be considered experimental rather than theoretical.

Nevertheless, there are some apparent differences in the algorithms implemented by the machine-vision and network models. The machine algorithms proceed by serially examining relations between adjacent pixels, whereas the network integrates information over the entire image in parallel. Because this network is fully connected, its response to any region of the image is simultaneously influenced by all other regions in a nonlinear fashion. We would therefore classify the algorithm implemented by the network as intrinsically global, whereas globality in machine algorithms is achieved by iterating some local analysis. Even if one viewed our entire network as processing a local patch of a more complex image, as was suggested earlier, the more general network to which all the local networks project would still be analysing the image in a global manner.

The distinctiveness of the network at the algorithmic level is a result of the constraints, imposed at the hardware level, of being constructed out of large numbers of highly interconnected analogue units each having very limited capabilities. It is common in research on artificial intelligence and cognitive modelling to focus exclusively on algorithms at the expense of questions of implementation and hardware. Although it is true that a particular algorithm can be implemented in innumerable ways, the converse is not true; a particular hardware configuration cannot necessarily embody a wide variety of algorithms. If one is constrained to solving a problem by using a network architecture similar to that occurring in the brain, the resulting solution may be quite different from that that would have been conceived if one were trying to solve the problem in any manner possible.

No claim is made here that the network has found a general solution to the shape-from-shading problem. The network was constructed to deal only with a particular set of simple surfaces. The aim here was to study the basic capabilities of the network learning algorithm, and to use the algorithm to investigate possible organizations of the visual cortex, rather than to construct a practical system for interpreting images of the real world. The simple surfaces we used could have been solved by conventional algorithms (even local algorithms, although we claim that ours is global). However, we feel that the highly parallel and connected nature of networks may take them particularly suitable for integrating information across complex images to form a globally self-consistent description, although this remains to be demonstrated. That is not to say that networks will be able to recover a mathematically exact description of a surface, but only a sufficient approximation, which may be the best that biological systems can do in any case.

Conclusions

Although neural network modelling is at an early stage of development, it is apparent that new principles are emerging concerning the representation and transformation of information within populations of neurons. For example, Georgopoulos *et al.* (1986) have shown that, in motor cortex, information about the intended direction of arm movement is distributed in populations of neurons broadly tuned to that direction, analogous to the output representation used in this model. Zipser & Andersen (1988) have applied the same learning algorithm used here to construct a transform from retina-based coordinates to head-centred coordinates. They report that hidden unit properties in their model are similar to those of some neurones in parietal cortex. The success of their model and ours depended to a large extent on incorporating knowledge about single unit properties and the style of representation found in cerebral cortex. Learning algorithms provide a new technique for drawing out the implication of these assumptions and exploring some of the principles of distributed processing in sensory and motor systems.

A central task in modelling is to abstract features of the system that are relevant to the behaviour in question. Depending on what one wants to study, different levels of abstraction become appropriate. Incorporating as much realism as possible into model neurons or the organization of the network ought not to be a goal in itself. In doing so one risks a model whose complexity, although perhaps capable of mimicking the data, may not provide any insight into function. For example, detailed models of orientation-selectivity of simple cells in visual cortex provide a useful way of exploring the anatomical basis for this selectivity (Sejnowski *et al.* 1988; Wehmeier *et al.* 1989), but do not explain the possible functions of these properties. The claim for 'simplifying' models, which omit many biophysical features of real neurons, is that the general organization of a richly interconnected network of simplified units captures certain essential computational features of the cortex, while being different in detail (Sejnowski *et al.* 1988).

We have shown that a network model can determine surface curvatures from images of certain shaded surfaces independently of illumination direction. Processing units in the model had receptive fields similar to those found in the visual cortex. As was stressed earlier, it is unlikely that anyone would infer, merely by examining receptive fields, that the network deals with smooth gradations of shaded curves. To understand a network it appears essential to know the organization of the projective fields as well as the receptive field. Yet even with the entire circuit laid out before our eyes it was still difficult to comprehend. This provides a lesson for research aimed at trying to understand real neural networks, in which we are given access to an infinitesimal part of the circuit and in which determination of projective fields is beyond the present technology. In this difficult situation, perhaps the greatest contribution this modelling can provide is to help us realize that a neuron can be doing something entirely different from what initial impressions would suggest.

This work was supported by a Presidential Young Investigator Award to T.J.S. and a grant from the Sloan Foundation to T.J.S. and G. F. Poggio. We thank David Rose for useful comments on the manuscript.

Appendix 1. Back-propagation learning algorithm

Back-propagation is a supervised learning procedure, developed by Rumelhart *et al.* (1986) (and independently by Le Cun (1985), Parker (1985) and Werbos (1987)), which modifies connection strengths throughout a feedforward neural network to reduce the error between the actual and correct outputs.

The input to a unit i, E_i, is the sum of all unit activities s_j in the previous layer:

$$E_i = \sum_j w_{ij} s_j, \tag{1.1}$$

where w_{ij} is the weight from the jth to the ith unit. The output of the ith unit, s_i, is formed by passing E_i through the sigmoid nonlinearity:

$$s_i = P(E_i) = 1/(1+\exp[-E_i]). \tag{1.2}$$

The algorithm minimizes the average squared error between the actual outputs, $s_i^{(n)}$, and correct outputs, s_i^*. (A superscript denotes the layer, with the output layer designated N.) The error arising from a single input pattern is:

$$\text{error} = \sum_{i=1}^{J} (s_i^* - s_i^{(N)})^2, \tag{1.3}$$

where J is the number of output units.

During each learning trial, input trials are exposed to the stimulus and their activities propagated up through subsequent layers by repeated use of equations 1.1 and 1.2 until the output layer is reached.

Given the outputs, the error gradient $\delta_i^{(N)}$ for each output unit is:

$$\delta_i^{(N)} = (s_i^* - s_i^{(N)}) P'(E_i^{(N)}). \tag{1.4}$$

Error gradients for units in earlier layers are found by propagating errors down the network, layer by layer:

$$\delta_i^{(n)} = (\sum_j \delta_j^{(n+1)} w_{ji}^{(n)}) P'(E_i^{(n)}). \tag{1.5}$$

The weight gradient $\Delta w_{ij}^{(n)}$ in equation 1.6 depends on the two factors in square brackets: (i) the activity along the output line for the weight and (ii) the unit's error gradient. The weight change is based on a running average of the weight gradient with an exponentially decaying filter, and the rest of equation 1.6 concerns this smoothing:

$$\Delta w_{ij}^{(n)}(\text{new}) = \alpha \Delta w_{ij}^{(n)}(\text{old}) + (1-\alpha)[\delta_i^{(n+1)} s_j^{(n)}], \tag{1.6}$$

where $\alpha = 0.95$. We averaged equation 1.6 over five input patterns before updating $\Delta w_{ij}^{(n)}$(new). The smoothed $\Delta_{ij}^{(n)}$ are then used to update the weights:

$$w_{ij}^{(n)}(\text{new}) = (1-\beta) w_{ij}^{(n)}(\text{old}) + \epsilon \Delta w_{ij}^{(n)}, \tag{1.7}$$

where the learning rate $\epsilon = 10.0$ and $\beta = 0.0001$ is a weight decay term. The error signal was back-propagated only when the output error was greater than 0.03.

In Rumelhart *et al.* (1986), ϵ rather than $(1-\alpha)$ is used in equation 1.6. Our parameter α smooths the weight gradient independently of ϵ, and our averaging

procedure makes it unnecessary to scale ϵ by the number of presentations per weight update. Finally, in our notation the subscripts i and j are reversed compared with Rumelhart *et al.* (1986).

Appendix 2. Illumination and reflectance model

The illumination and reflectance model is illustrated geometrically in figure 10. The value of θ indicates the angle between the surface-normal vector and the predominant illumination. The light intensity reflected from a surface is indicated by the length of the vector extending from the origin to the circle.

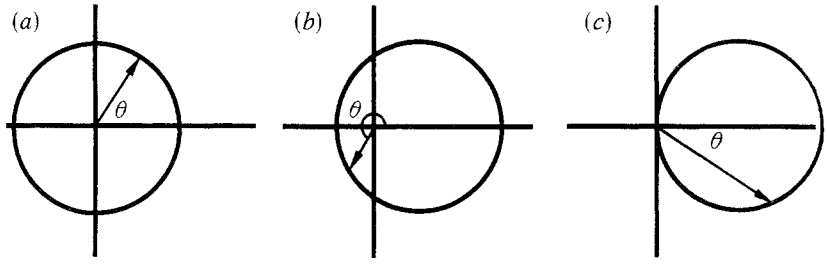

Figure 10. Geometrical summary of illumination and reflectance model under three illumination conditions, of which the second was used in the shape-from-shading model. The angle between the surface normal and the predominant illumination direction is represented by θ, and the length of the vector from the origin to the circle indicates the light intensity reflected from the surface. (*a*) completely diffuse illumination; (*b*) partly diffuse illumination; (*c*) completely directed illumination.

For completely diffuse illumination, the surface is uniformly lit from all directions and there is no shading. This is represented in figure 10*a* by a circle centred on the origin, in which the reflection vector has constant length independent of orientation. Going the other extreme, completely directed illumination (figure 10*c*) is represented by a circle tangent to the ordinate. Here the length of the reflection vector is $\cos\theta$, which is Lambertian reflection, provided $-90 < \theta < 90$. Those limits correspond to the occurrence of sharp shadow edges, because the reflectance vector is zero beyond them. Because we wanted illumination properties between these two extremes, we modelled it by simply shifting the circle to an intermediate position, as shown in figure 10*b*.

The equation for this is

$$R = a\cos\theta + \sqrt{(a^2(\cos^2\theta - 1.0) + b^2)}, \tag{2.1}$$

where

$$a = 0.5 - (R_{\min}/2.0),$$

$$b = 0.5 + (R_{\min}/2.0).$$

This is the equation for a circle in polar coordinates, where b is the radius of the circle whose centre has been shifted by a from the origin. $R_{\min}$ is the minimum intensity of light reflected from the surface (the left edge of the circle), which occurs when the surface normal is 180° from the predominant illumination direction. The

parameters a and b both depend on $R_{\min}$. Intuitively, a determines the contribution of scattered illumination and b is a normalization factor setting maximum reflected intensity to 1.0 (i.e. it expands or contracts the circle so that its right edge lies at a unit distance from the origin). This normalization is not shown in figure 10, where all circles are of equal diameter.

If $R_{\min} = 0$ then equation 2.1 reduces to Lambertian reflectance $R = \cos\theta$, with light coming entirely from one direction. At the other extreme, if $R_{\min} = 1.0$ then equation 2.1 reduces to the constant $R = 1.0$. In this case the illumination is entirely isotropic, and there is no shading. By setting $0.0 < R_{\min} < 1.0$, the degree of anisotropy in the illumination can be continuously varied. For the work presented here $R_{\min}$ was always 0.05.

This model is an empirical method for handling diffuse illumination, and is not based on a consideration of the physics of the situation. Scattered light is handled by the fiction that there is non-zero reflection of the incident light when θ lies beyond the range $-90°$ to $90°$, conditions for which the surface in reality is not directly exposed to the illumination.

Moving to a different topic, the terms 'tilt' and 'slant' have been used to described illumination direction. These have the following meanings. Let the image plane be defined by the x and y axis, the depth plane by the z axis, and the illumination direction by the vector (x, y, z) from the surface to the light source. Tilt is the angle formed by the projection of the illumination vector within the x–y plane (arctan (y/x)) and slant is the angle formed by the projection of the illumination vector within the z–y plane (arccos(z)). These two angles will uniquely identify the direction of illumination.

Appendix 3. Response function of input and output units

Input units

The equation for the spatial receptive fields of the input units was

$$R(x, y) = \{1 - [(x^2 + y^2)/\sigma^2]\} \exp - [(x^2 + y^2)/\sigma^2] \tag{3.1}$$

(illustrated in figure 3*a*), which is the Laplacian of a two-dimensional gaussian. Equation 3.1 defines an on-centre unit. An off-centre unit would be $-R(x, y)$. The space constant σ was 0.07°. The diameter of the receptive-field centre was determined by the zero-crossings of $R(x, y)$ and was equal to 2σ. Receptive-field centres in the hexagonal array of the input layer were separated by a distance σ.

Unit responses were normalized to 1.0 when presented with an optimal light stimulus. This was a light intensity whose spatial extent exactly matched the excitatory regions of the receptive fields, leaving the inhibitory regions in darkness. Also, responses of the input units were rectified so that they could not assume negative values. Positive convolution values were assigned to the on-centre unit and the off-centre unit at the same location was set to zero. If the convolution was negative, the on-centre unit was set to zero and the corresponding off-centre unit was set to the absolute value of the convolution.

Output units

This section describes the ideal output values of the network, and expands on equation (1) of the main text

$$R(M,O) = A(M)\,B(O), \tag{3.2}$$

where the output unit's response was a function of both the magnitude, M, and orientation, O, of the stimulus' principal curvatures. The first term, $A(M)$, was a log-normal function of curvature magnitude, normalized to have a peak value of 1.0:

$$A(M) = \exp -\{[\log (M/M_{\text{peak}})]^2/\sigma^2\}, \tag{3.3}$$

where M_{peak}, the peak of the curvature magnitude tuning curve, equalled $+8$ deg^{-1} or -8 deg^{-1}, and $\sigma = 0.69$ deg^{-1}, set to give a tuning half-width of one octave at $1/e$ height. The second term, $B(O)$, was a gaussian function of curvature orientation:

$$B(O) = E(M_{\text{L}},M_{\text{S}}) \exp -\{(O-\mu)^2/[\sigma/E(M_{\text{L}},M_{\text{S}})]^2\}, \tag{3.4}$$

where orientation tuning bandwidth $\sigma = 30°$ and orientation tuning peaks $\mu = 0°$, 30°, 60°, 90°, 120°, and 150°. M_{L} and M_{S} refer to the absolute magnitudes of the larger and smaller of the two principal curvatures.

The factor $E(M_{\text{L}},M_{\text{S}})$ in equation (3.4) is the eccentricity function, which is a sigmoid function of the ratio of the two principal curvatures.

$$E(M_{\text{L}},M_{\text{S}}) = 1.0/\{1.0+\exp -[(M_{\text{L}}/M_{\text{S}}-m)/n]\}, \tag{3.5}$$

where $m = 1.3$ and $n = 0.14$. It is called such because it is a function of the 'roundedness' or eccentricity of the elliptical surface cross sections, and has nothing to do with position in the visual field. Its effect is to make orientation tunings of output units increasingly broad and shallow as the ratio of the two principal curvatures approaches unity. This was desired because when $M_{\text{L}}/M_{\text{S}} = 1.0$, curvature in every direction is identical and it becomes meaningless to talk about curvature orientation. Under this condition, we wanted all orientation mechanisms to respond equally, and also respond poorly, Including the eccentricity function was a device for *creating* the network, and was not in place during operation of the mature network.

References

Andrews, B. W. & Pollen, D. A. 1979 Relationship between spatial frequency selectivity and receptive field profile of simple cells. *J. Physiol., Lond.* **287**, 163–176.

Barto, A. G., Sutton, R. S. & Anderson, C. W. 1983 Neuronlike adaptive elements that can solve difficult learning control problems. *IEEE Trans. Systems, Man Cybernet.* **13**, 835–846.

Brady, M. 1979 Inferring the direction of the sun intensity values on a generalized cone. In *Proc. Int. Joint Conf. Art. Intel.* pp. 88–91.

Bulthoff, H. & Mallot, H. 1988 Integration of depth modules: stereo and shading. *J. Opt. Soc. Am.* **A5**, 1749–1758.

DeValois, K. K, DeValois, R. L. & Yund, E. W. 1979 Responses of striate cortex cells to grating and checkerboard patterns. *J. Physiol., Lond.* **291**, 483–505.

Dobbins, A., Zucker, S. W. & Cynader, M. S. 1987 End-stopped neurons in the visual cortex as a substrate for calculating curvature. *Nature, Lond.* **329**, 438–441.

Georgopolous, A. P., Schwartz, A. B. & Kettner, R. E. 1986 Neuronal population coding of movement direction. *Science, Wash.* **233**, 1416–1419.

Gilbert, C. D. 1983 Microcircuitry of visual cortex. *A. Rev. Neurosci.* **6**, 217–247.

Gilbert, C. D. & Wiesel, T. N. 1979 Morphology and intracortical projections of functionally characterized neurones in the cat visual cortex. *Nature, Lond.* **280**, 120–125.

Gregory, R. L. 1970 *The intelligent eye.* New York: McGraw Hill.

Helmholtz, H. von 1909/1962 *Physiological optics* (New York: Dover, 1962); English translation by J. P. C. Southall for the Optical Society of America (1924) from the 3rd German edition of *Handbuch der physiologischen Optik* (Hamburg: Voss, 1909).

Hinton, G. E. 1989 Connectionist learning procedures. *Artif. Intell.* **40**, 185–234.

Hinton, G. E., McClelland, J. L. & Rumelhart, D. E. 1986 Distributed representations. In *Parallel distributed processing: explorations in the microstructure of cognition,* vol. 1 (*Foundations*) (ed. D. E. Rumelhart & J. L. McClelland), pp. 77–109. Cambridge Massachusetts: MIT Press.

Horn, B. 1986 *Robot vision.* Cambridge, Massachusetts: MIT Press.

Hubel, D. H. & Wiesel, T. N. 1962 Receptive fields, binocular interactions, and functional architecture in the cat's visual cortex. *J. Physiol., Lond.* **160**, 106–154.

Hubel, D. H. & Wiesel, T. N. 1965 Receptive fields and the functional architecture in two nonstriate areas visual (18 and 19) of the cat. *J. Neurophysiol.* **28**, 229–289.

Ikeuchi, K. & Horn, B. 1981 Numerical shape from shading and occluding boundaries. *Artif. Intell.* **17**, 141–184.

Koch, C. & Poggio, T. 1987 Biophysics of computation: neurons, synapses, and membranes. In *Synaptic Function* (ed. G. M. Edelman, W. E. Gall & W. M. Cowan), pp. 637–697. New York: John Wiley and Sons.

Koenderink, J. J. & van Doorn, A. J. 1980 Photometric invariants related to solid shape. *Optica Acta* **27**, 981–996.

Kohonen, T. 1984 *Self-organization and associative memory.* Berlin: Springer-Verlag.

Kulikowski, J. J. & Bishop, P. O. 1981 Linear analysis of the responses of simple cells in the cat visual cortex. *Expl Brain Res.* **44**, 386–400.

Le Cun, Y. 1987 Modèles de l'apprentisage connexioniste. Ph.D. thesis, Université Pierre et Marie Curie, Paris, France.

Lee, C., Rohrer, W. H. & Sparks, D. L. 1988 Population coding of saccadic eye movements by neurons in the superior colliculus. *Nature, Lond.* **332**, 357–360.

Lehky, S. & Sejnowski, T. J. 1988 Network model of shape-from-shading: neural function arises from both receptive and projective fields. *Nature, Lond.* **333**, 452–454.

Linsker, R. 1986 From basic network principles to neural architecture: Emergence of spatial-opponent cells. *Proc. natn. Acad. Sci. U.S.A.* **83**, 7508–7512.

Lord, E. A. & Wilson, C. B. 1984 *The mathematical description of shape and form.* Chichester: Ellis Horwood.

Mingolla, E. & Todd, J. T. 1986 Perception of solid shape from shading. *Biol. Cyb.* **53**, 137–151.

Parker, D. B. 1985 *Learning logic.* MIT Center for Computational Research in Economics and Management Science, Technical Report no. TR-47.

Pentland, A. P. 1984 Local shading analysis. *IEEE Trans. Pattern Anal. Mach. Intell.* **6**, 170–187.

Pineda, F. J. 1987 Generalization of backpropagation to recurrent neural networks. *Phys. Rev. Lett.* **59**, 2229–2232.

Rall, W. & Segev, I. 1987 Functional possibilities for synapses on dendrites and dendritic spines. In *Synaptic function* (ed. G. M. Edelman, W. E. Gall & W. M. Cowan). New York: John Wiley and Sons.

Ramachandran, V. S. 1988*a* Perception of shape from shading. *Nature, Lond.* **331**, 163–166.

Ramachandran, V. S. 1988*b* Perceiving shape from shading. *Scient. Am.* **259** (August), 76–83.

Rumelhart, D. E., Hinton, G. E. & Williams, R. J. 1986 Learning internal representations by error propagation. In *Parallel distributed processing: explorations in the microstructure of cognition,* vol. 1 (*Foundations*) (ed. D. E. Rumelhart & J. L. McClelland), pp. 318–362. Cambridge, Massachusetts: MIT Press.

Sejnowski, T. J., Koch, C. & Churchland, P. 1988 Computational neuroscience. *Science, Wash.* **241**, 1299–1306.

Sillito, A. M. 1975 The contribution of inhibitory mechanisms to the receptive field properties of neurones in the striate cortex of cat. *J. Physiol., Lond.* **250**, 305–329.

Stone, J. 1983 *Parallel processing in the visual system.* New York: Plenum Press.

Todd, J. T. & Mingolla, E. 1983 Perception of surface curvature and direction of illumination from patterns of shading. *J. Exp. Psychol. Hum. Percep. Perform.* **9**, 583–595.

Toyoma, K., Maekawa, K. & Takeda, T. 1977 Convergence of retinal inputs onto visual cortical cells. I. A study of the cells monosynaptically excited from the lateral geniculate body. *Brain Res.* **137**, 207–220.

Wassle, H., Boycott, B. B. & Illing, R.-B. 1981 Morphology and mosaic of on- and off-beta cells in the cat retina and some functional considerations. *Proc. R. Soc. Lond.* B **212**, 177–195.

Wehmeier, U., Dong, D., Koch, C. & Van Essen, D. 1989 Modeling the mammalian visual system. In *Methods in neuronal modeling* (ed. C. Koch & I. Segev), pp. 335–359. Cambridge, Massachusetts: MIT Press.

Werbos, P. J. 1987 Building and understanding adaptive systems: A statistical/numerical approach to factory automation and brain research. *IEEE Trans. Systems, Man Cybernet.* **17**, 7–20.

Zipser, D. & Andersen, R. A. 1988 A back-propagation programmed network that simulates response properties of a subset of posterior parietal neurons. *Nature, Lond.* **331**, 679–684.