A Canonical Microfunction For Learning Perceptual Invariances

James V Stone*
Biological Sciences/Cognitive and Computing Sciences,
University of Sussex, Sussex, BN1 9QH, England.
jims@cogs.susx.ac.uk

Abstract

An unsupervised method is presented which permits a set of model neurons, or a *microcircuit*, to learn low level vision tasks, such as the extraction of surface depth. Each microcircuit implements a simple, generic strategy which is based on a key assumption: Perceptually salient visual invariances, such as surface depth, vary smoothly over time. In the process of learning to extract smoothly varying invariances, each microcircuit maximises a *microfunction*. This is achieved using a learning rule which maximises the long-term variance of each unit's output, whilst simultaneously minimising its short-term variance. The learning rule involves a linear combination of anti-Hebbian and Hebbian weight changes, over short and long time scales, respectively. The method is demonstrated on a hyper-acuity task; estimating sub-pixel stereo disparity from a temporal sequence of random-dot stereograms. After learning, the microcircuit generalises, without additional learning, to previously unseen image sequences. It is proposed that the approach adopted here may be used to define a *canonical microfunction*, which can be used to learn many perceptually salient invariances.

Introduction

The ability to learn perceptually salient visual invariances - surface orientation, curvature, depth, texture, and motion - is a prerequisite for the more familiar tasks (e.g. tracking and catching prey) associated with biological vision. This paper addresses the question: What strategies enable neurons to learn these invariances from a temporal sequence of images, without the aid of an external teacher?

The neuroanatomical uniformity of structure across different areas of the mammalian neocortex may correspond to an underlying functional uniformity in terms of its ability to learn (Marr, 1970; Creutzfeldt, 1978; Szenátgoathai, 1978; Douglas, Martin, & Whitteridge, 1989; Barlow, 1985; Ebdon, 1993). One compelling finding consistent with this hypothesis is provided in (Métin & Frost, 1989), where it was demonstrated that the somatic cortex of hamsters developed visually responsive neurons after retinal fibres had been redirected into the somatosensory thalamus. In a similar experiment, young ferrets developed visually responsive neurons in the auditory cortex after retinal fibres had been redirected into the auditory thalamus (Roe, Pallas, Hahm, & Sur, 1990). These findings are consistent with the proposal (Douglas et al., 1989) that different regions of the mammalian cortex may utilise a single type of *canonical microcircuit*. These microcircuits are viewed as a functionally modular unit of processing in the neocortex.

Rather than attempting to construct a detailed functional *and* structural model of a canonical microcircuit, the approach adopted here is to model putative *functional* characteristics implemented by such a circuit. Accordingly, it is assumed that a microcircuit implements a *canonical microfunction* (the term *micro*function is used to emphasise that the function is associated with a microcircuit, rather than with the whole CNS). Specifically, a microcircuit adjusts its internal connections so as to maximise a canonical microfunction.

A canonical microfunction embodies a generic strategy for learning perceptual invariances. If implemented by a microcircuit then such a strategy permits learning of perceptually salient invariances which include depth, colour, and surface orientation. One example of such a strategy is based on an assumption of spatial smoothness (Marr, 1982; Becker & Hinton, 1992). Marr (Marr, 1982) assumed spatial smoothness in order to be able to extract stereo disparity using hand-crafted (non-learning) methods. Marr's observation (Marr, 1982)(p114) that, "disparity varies smoothly almost everywhere" can be used to permit learning of random dot stereograms by a neural network model (Becker & Hinton, 1992).

*Current address: Psychology Building, Western Bank, University of Sheffield, Sheffield, S10 2TP.

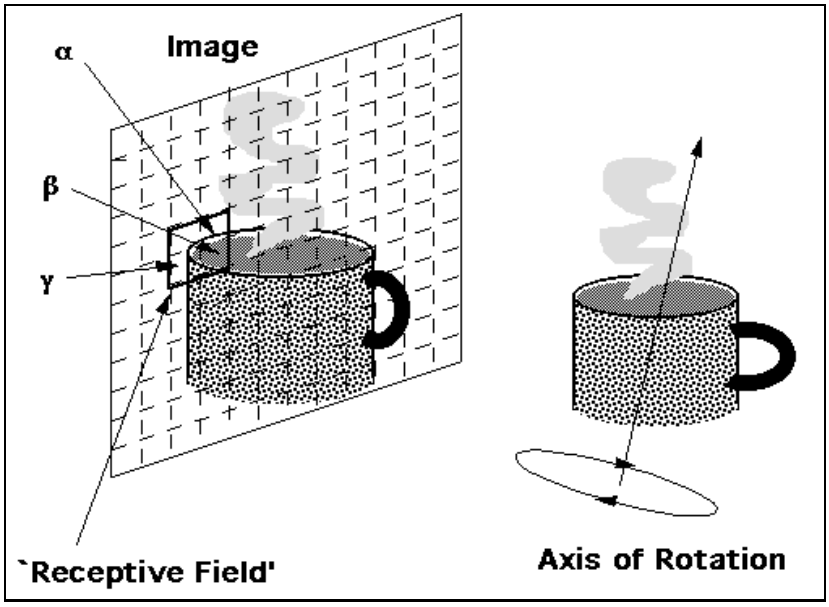

Figure 1: A small amount of object rotation can have a dramatic effect on the intensities (e.g. α, β and γ) of individual image pixels. The intensities of pixels in a 'receptive field' change as the object rotates. In this example, three pixel intensities define an image vector, (α, β, γ), as depicted in Figure 2a.

The approach adopted here is consistent with that of (Douglas et al., 1989) and others, but it is more general because it allows us to concentrate on computational aspects of perceptual learning in a modular neuronal model.

Learning Invariances Using Spatio-Temporal Constraints

According to Gibson (Gibson, 1979), the problem of vision consists of obtaining invariant structure from continually changing sensations. Essentially, Gibson stated that perceptual invariances are quantities which remain the same when subjected to visual transformations, such as changes in view angle. In mathematics, an invariant is any quantity which remains unchanged with respect a given set of transformations. This more general definition permits us to consider quantities such as surface depth in terms of Gibson's theory. For example, it is important to be able to perceive the depth of a surface, irrespective of the texture on that surface. Here, the invariant is surface depth, and the set of transformations is defined in terms of surface texture. Thus, we can legitimately consider a perceptually salient physical parameter such as depth in terms of Gibson's theory of invariances. Accordingly, the terms "invariance" and "parameter" are used interchangeably here.

The potential of Gibson's approach has recently begun to be realised as a series of connectionist models (Becker & Hinton, 1992; Becker, 1992; Zemel & Hinton, 1991; Foldiak, 1991; Schraudolph & Sejnowski, 1991; Mitchison, 1991; Phillips, Kay, & Smyth, 1995). The model described in this paper is substantially different from these models, although it shares with them a common assumption: A learning mechanism can discover perceptually salient visual invariances by taking advantage of quite general properties (such as spatial and temporal smoothness) of the physical world. These properties are not peculiar to any single physical environment so that such a mechanism should be able to extract a variety of perceptually salient invariances (e.g. 3D orientation and shape) via different sensory modalities such as vision, audition and touch.

There are approximately 10^6 retinal ganglion cells projecting from primate retina to the lateral geniculate nucleus; the activities of these cells can be described in terms of an *image vector* with 10^6 components. Similarly, the activities of m ganglion cells derived from a retinal receptive field can be described in terms of an image vector with m components. In general, each component of an image vector is the the intensity of an image pixel, or the activity of a ganglion cell. *Ultimately, it is the activities of sets of ganglion cells, with each set derived from a localised retinal receptive field, that the visual system receives as input.* The activities of ganglion cells change as the pattern of light falling on a receptive field changes, causing the image vector to describe a trajectory through an m-dimensional space. However, the form of this trajectory is caused by the changes in the physical world, and can therefore be described in terms of a small number of perceptual invariances (e.g. surface depth). It is these invariances that are useful to an organism. A large part of the problem of perception consists of extracting physical invariances implicit in a changing image vector. The remainder of this section is intended to provide insight into how this might be achieved.

The following example uses an image of a rotating object, but the analysis applies to any region (receptive field) in an

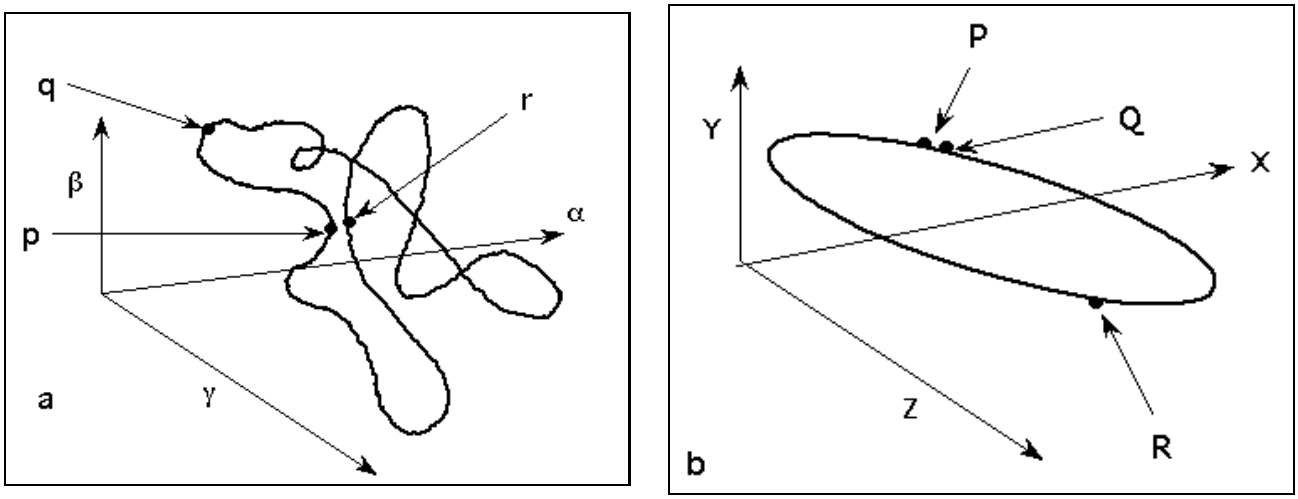

Figure 2: Diagram of (a) image vector and (b) parameter vector. Three physical parameters (rotation of an object around the X, Y and Z axes) defines a 3D parameter vector which changes over time (b). The corresponding changes in three image pixels (α, β and γ in Figure 1) are represented as the change in position of a 3D image vector over time (a).

image, and also to perceptually salient physical parameters other than rotation (e.g. depth, motion and curvature).

Figure 1 shows a rotating textured object and its image I. Clearly, a small change in the orientation of the object can give rise to a relatively large change in the intensity of individual pixels as the texture moves across the image plane. In general, a perceptually salient parameter such as rotation is characterised by variability over time, but the rate of change of the parameter is usually small, relative to that of the intensities of individual image pixels.

The intensities α, β and γ of three of the n pixels in I are plotted along perpendicular axes in Figure 2a. Any point, such as p, defines a value for the intensities of these pixels at one time. The intensities of pixels change rapidly but continuously as the object rotates, and the image vector[1] (α, β, γ) sweeps out a trajectory as shown in Figure 2a. Just as (α, β, γ) defines a 3D image vector, so, the n pixels in I define an nD image vector. A similar type of trajectory is swept out by this nD image vector, but we would need n axes to display it.

In Figure 2b, the three components of a *parameter vector* are shown. Each component of this vector specifies the rotation of an object around a 'world' axis (X, Y or Z). The closed loop of the parameter vector in Figure 2b implies complete rotation of the imaged object, so that the object begins and ends in the same orientation. This, in turn, implies that the initial and final images are identical. Therefore, the sequence of image vectors also defines a closed loop, as shown in Figure 2a.

For display purposes, the object in this example rotates at a constant rate, so that the temporal proximity of two parameter vectors is proportional to their proximity along the curve in Figure 2b (e.g. P and Q). Given a sequence of image vectors derived from a rotating object, can the corresponding sequence of parameter vectors be recovered?

It is tempting to assume that if two image vectors are near to each other then they were derived from a single object at similar orientations. Unfortunately, a rotating object can generate quite different image vectors, even though the amount of object rotation which separates these image vectors is small. Therefore, the similarity between successive image vectors tells us little about the similarity of the object orientations which generated those image vectors. For example, the similar object orientations implied by the nearby parameter vectors P and Q in Figure 2b correspond to quite widely separated image vectors p and q in 2a. This occurs if the orientation of a textured object changes by a small amount and causes a large change in the intensity of image pixels. However, images at successive time frames usually depict the object at similar orientations in 3-space, even though these images define quite different image vectors (e.g. p and q). This suggests that a perceptual system should infer that the images corresponding to p and q show a single object at similar orientations.

Conversely, two image vectors such as p and r may be similar, but if they were widely separated in time (as implied by the distance between P and R in Figure 2b) then it is likely that they were generated by quite different object orientations. In this case, a perceptual system would do well to ignore the spurious similarities between images corresponding to p and r, because they occurred at quite different times.

In summary, the temporal proximity of image vectors provides a *temporal binding* of parameter values, such as orientation in 3-space. It is this temporal binding which permits us to infer legitimately that temporally proximal images are derived from similar physical scenarios. It also permits us to infer that images separated by long time

[1]Each image vector defines a point, and the terms "point" and "vector" are used interchangeably below.

intervals are likely to be derived from different physical scenarios.

It is noteworthy that conventional unsupervised learning techniques (e.g. Kohonen maps (Kohonen, 1984), Hebbian learning (Oja, 1982)) which cluster input vectors according to their Euclidean distance would not, in general, be capable of clustering together images which were generated by similar physical scenarios. In contrast, the method presented here takes advantage of the temporal proximity of (often dissimilar) input vectors to discover which invariances they share.

So far, the general characteristics of how perceptually salient parameters change over time have been described. These observations can be used to constrain the outputs of a model microcircuit such that its outputs come to reflect these general characteristics. This can be achieved without specifying the desired output (target) value for any input to the microcircuit. An 'economical' way for a microcircuit to generate such a set of outputs is to adapt its connection weights so that the outputs specify some invariance which is implicit in the microcircuit's inputs.

The Learning Method

A model which uses a type of *temporal smoothness* constraint can be made to learn visual invariances. The degree of smoothness of the output or *state* of a model unit can be measured in terms of the 'temporally local', or *short term*, variance associated with a sequence of output values. A sequence of states defines a curve which is maximally smooth if the variance of this curve is minimal (the straighter the curve, the smoother the output). However, minimising only the short term variance has a trivial solution. This consists of setting all model weights to zero, generating a horizontal output curve. This is consistent with one characteristic, smoothness, of perceptually salient invariances, but it does not conform to the other characteristic, variability over time. The output can be made to reflect both smoothness *and* variability by forcing it to have a small short-term variance, and a large *long-term* variance. Thus the variance of the output over small intervals should be small, relative to its variance over longer intervals.

The general strategy just described can be implemented using a multi-layer model. Units in the input, hidden and output layers are labelled i, j and k, respectively. Input and output layers have linear units, and the hidden layer has $tanh$ units. The state of an output unit u_k at each time t is $z_{kt} = \sum_j w_{jk}\, z_{jt}$, where w_{jk} is the value of a weighted connection from the jth hidden unit to u, and z_{jt} is the state of the jth hidden unit.

We can obtain the desired behaviour in z_k by altering inter-unit connection weights such that z_k has a large long-term variance V, and a small short-term variance U. That is, by making V/U large. These requirements can be embodied in a microfunction F:

$$F = log\, \frac{V}{U} = log\, \frac{1/2 \sum_{t=1}^{T} (\overline{z}_{kt} - z_{kt})^2}{1/2 \sum_{t=1}^{T} (\tilde{z}_{kt} - z_{kt})^2} \tag{1}$$

Where z_{kt} is the state of unit k at time t. (The $\frac{1}{2}$'s are formally redundant, but have been introduced to simplify the derivatives of U and V). The cumulative states $\tilde{z}_{kt}$ and $\overline{z}_{kt}$ are both temporal exponentially weighted sums of states z_k:

$$\tilde{z}_{kt} = \lambda_S\, \tilde{z}_{k(t-1)} + (1 - \lambda_S)\, z_{kt-1} \quad : 0 \leq \lambda_S \leq 1 \tag{2}$$

$$\overline{z}_{kt} = \lambda_L\, \overline{z}_{k(t-1)} + (1 - \lambda_L)\, z_{kt-1} \quad : 0 \leq \lambda_L \leq 1 \tag{3}$$

The half-life h_L of λ_L is much longer (typically 100 times longer) than the corresponding half-life h_S of λ_S.

Learning consists of adjusting the microcircuit weights so as to maximise F. After learning, the difference between the short-term temporal mean $\tilde{z}_{kt}$ at each time step and the 'current' state z_{kt} tends to be small. This implies that each state z_{kt} tends to be similar to states which preceded it, so that the sequence of z_k states varies smoothly over time. Conversely, the difference between the long-term mean $\overline{z}_{kt}$ and the current state z_{kt} tends to be large. This implies that the sequence of z_{kt} states has a large range. A simple way to comply with these two constraints is for the microcircuit to adjust its weights such that z_{kt} varies smoothly in time over a relatively large range.

An information-theoretic interpretation of F is given in (Stone, 1995b). The derivative of F with respect to output weights results in a learning rule which is a linear combination of *Hebbian* and *anti-Hebbian* weight update, over long and short time scales, respectively[2]:

$$\frac{\partial F}{\partial w_{jk}} = \frac{1}{V} \sum_t (\overline{z}_{kt} - z_{kt})(\overline{z}_{jt} - z_{jt}) - \frac{1}{U} \sum_t (\tilde{z}_{kt} - z_{kt})(\tilde{z}_{jt} - z_{jt}) \tag{4}$$

[2]Thanks to Harry Barrow and Alistair Bray for pointing this out.

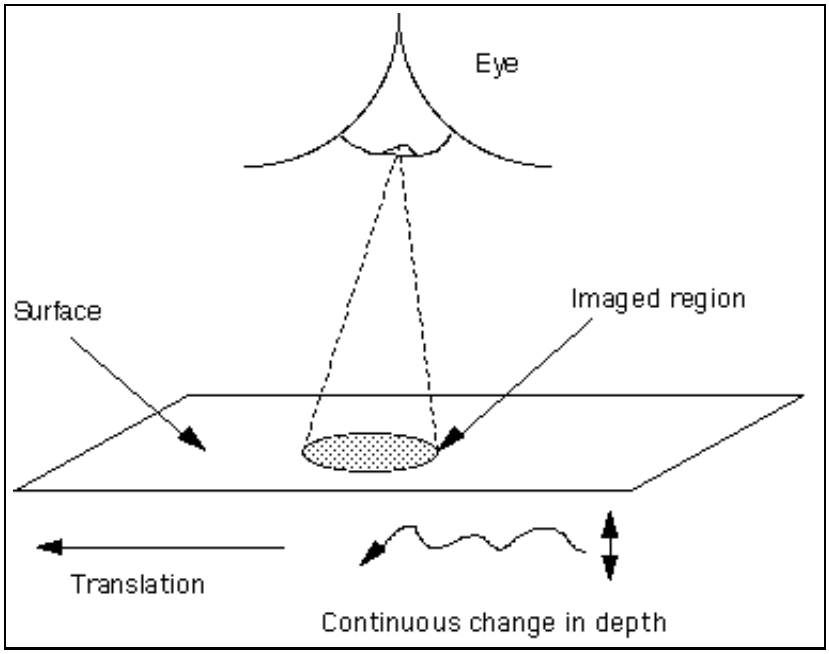

Figure 3: Variation of surface depth over time.

(For hidden unit weights, additional terms resulting from the *tanh* hidden unit activation function are required (see Appendix)).

The pre- and post-synaptic means used in conventional Hebbian learning rules (e.g. (Sejnowski, 1977)) have been replaced by the exponentially weighted means, $\overline{z}_{jt}$ and $\overline{z}_{kt}$ (respectively) in the Hebbian part of (4), and by $\tilde{z}_{jt}$ and $\tilde{z}_{kt}$ in the anti-Hebbian part of (4). In contrast, the rule described in (Bienstock, Cooper, & Munro, 1982) uses the exponentially weighted mean of only the post-synaptic output to modulate learning, and this learning is *either* Hebbian or anti-Hebbian, depending on the state of the post-synaptic unit. Thus, the rule defined in (4) uses the exponentially weighted mean of both the pre- *and* post-synaptic states to modulate both the Hebbian *and* anti-Hebbian learning applied to every weight.

The learning rule can be interpreted as follows. If V is small relative to U then learning is principally Hebbian, which has the effect of increasing the variability of outputs over long periods. That is, it prevents the state z_k of the output unit from being constant. However, if V is large relative to U, then learning is principally anti-Hebbian. These changes tend to generate an output sequence of z_k values which (like a perceptual invariance) has a large range, but which varies smoothly over time. Thus, in the process of maximising F, the microcircuit's output behaves increasingly like a perceptual invariance. An economical way to achieve this is for the microcircuit to find a set of weights which effectively 'discovers' an invariance in the microcircuit's input sequence.

The learning algorithm consists of computing the derivative of F with respect to every weight in the model to locate a maximum in F. The derivatives of F with respect to weights between the input and hidden unit layers are required. These derivatives are computed using the chain rule (but not the learning method) described in (Rumelhart, Hinton, & Williams, 1986). The cumulative result of these computations is used to alter weights only after the entire sequence of inputs has been presented. However, storage requirements are minimal because all quantities can be computed incrementally (see Appendix). This learning method also works if on-line weight update is used (Stone & Bray, 1995).

A conjugate gradient method (Williams, 1991) was used to maximise F. An more conventional iterative weight update rule which took steps of size $\eta\, \partial F/\partial w$ (where η is the learning rate) requires about 10 times as many iterations as the conjugate gradient method. As expected, both methods generate equally good solutions. The only reason for using the conjugate gradient method is that it is faster than the iterative method.

In conclusion, three noteworthy characteristics are: 1) The learning method is *unsupervised*, so that the microcircuit in not informed what the 'correct' output should be at any time. 2) Whereas weight changes depend on the recent history of inputs to a unit, a unit's output z is a function only of the current input. 3) Equation (1) is invariant with respect to the magnitude of z_k, and therefore with respect to the magnitude of the weights. Therefore, *no weight normalisation is required*. During learning, the pattern of weights alters, but the average magnitude varies relatively little.

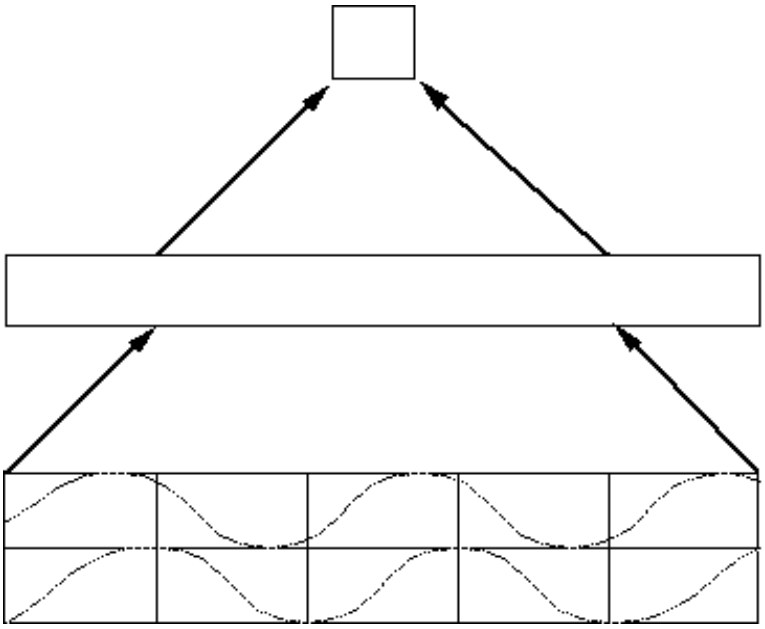

Figure 4: Network architecture, see text for details.

Experiments

Model Architecture

As shown in Figure 4, the model consists of three layers of units. Every unit in each layer is connected to every unit in the next layer. The first layer consists of two rows of five linear units. The 5-pixel left and right images of a stereo pair are presented to the upper and lower input row, respectively. The hidden layer consists of 10 units. The state of a unit in the hidden layer is $z = tanh(x)$, where x is the total input to a hidden layer unit from units in the input layer. The input to the jth hidden unit is $x_j = \sum_i (w_{ij} z_i + \theta_j)$, where w_{ij} is the value of a weighted connection from the ith input unit to the jth hidden unit, and z_i is the state of the ith input unit. All and only units in the hidden layer have a bias weight θ from a unit with constant output of 1. These bias weights are adapted in the same way as all other weights in the model. The output layer consists of a single linear unit.

Input Data: Random Dot Stereograms

The input data used during learning were designed to simulate a surface moving sinusoidally in depth (see Figure 3). These data were derived from an array of random dots S, similar to one member of the pair shown in Figure 5. The dot density used in all images in this paper is 0.167. The array S was convolved with a Gaussian filter (with standard deviation of one dot width) to simulate the blurring effect of the cornea on the retinal image. S was then normalised to have zero mean and unit variance. It might be argued that this represents an unreasonable amount of pre-processing. However, the normalisation does not produce a *locally* normalised image luminance. It thus represents less pre-processing than occurs in the primate retina where neuronal circuitry ensures that retinal ganglion cells receive inputs which are normalised relative to the local image luminance (Douglas, Martin, & Nelson, 1993).

A single *learning sequence* of 1000 stereo pairs was constructed from S. At each time step, a small patch of S was used to generate an input stereo pair of images. First, a sequence of 1000 sinusoidally varying disparity values between ± 1 was generated. The sine had a period of 1000 time steps. For each disparity value, a 1D 5-pixel image V_1 was generated by reading intensity values from a location in S into the 5-pixel image. At each time step, this location was advanced by two pixels to simulate the surface translating at constant velocity. The 5-pixel image vector was one image of a stereo pair presented to the microcircuit at each time step. The other vector V_2 of a pair was generated by shifting the location in S by an amount equal to the sinusoidally varying depth (disparity) value at the current time step. Most disparity values were less than one pixel, so that the second vector V_2 was generated by linear interpolation over intensity values in S.

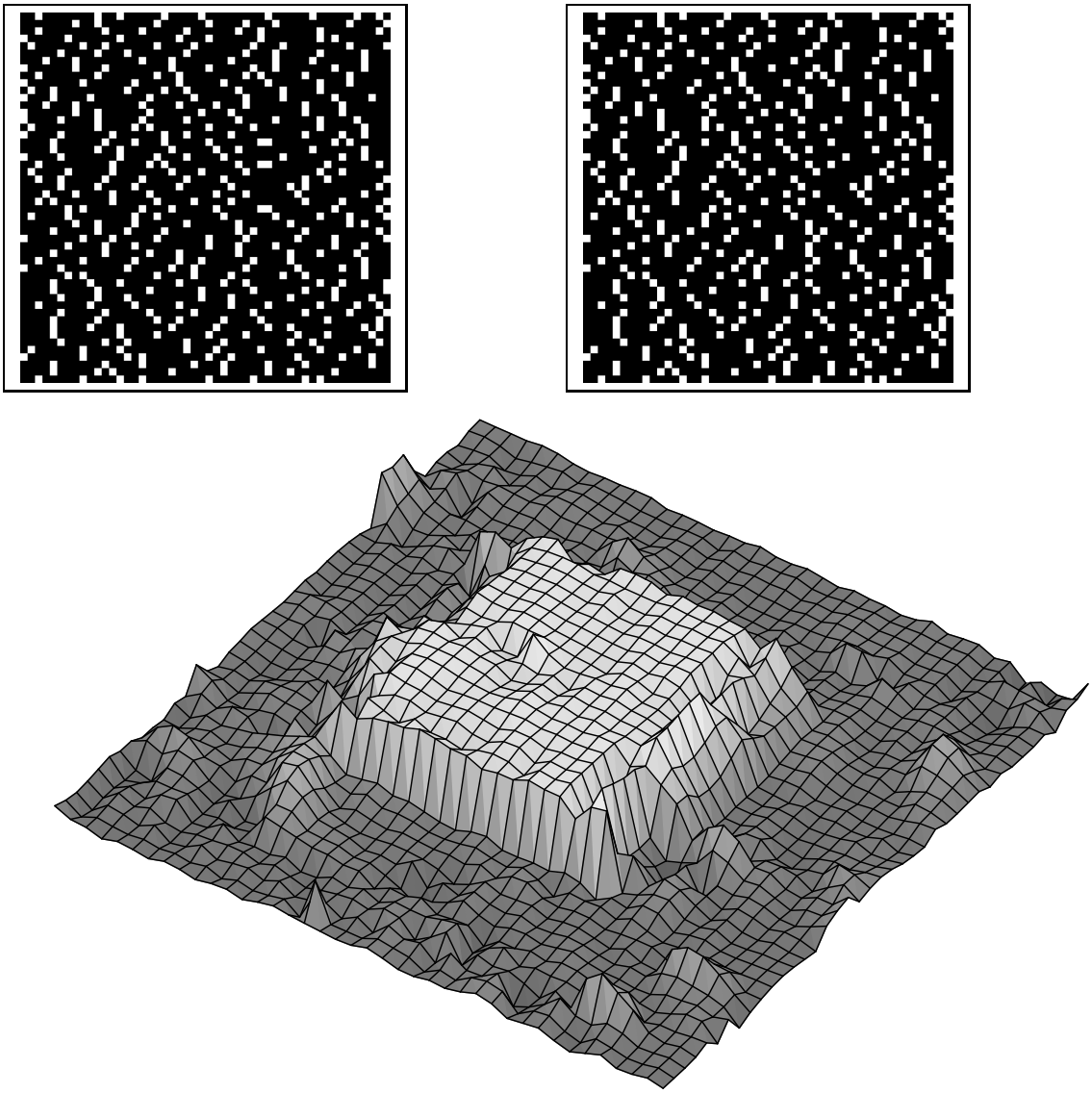

Figure 5: Random dot stereo pair with dot density 0.167, and depth map recovered by the microcircuit. The correlation between values in this depth map and stereo disparity was $r = 0.91$.

Setting Microcircuit Characteristics

The short-term half-life h_S of λ_S was 32 time steps. In this set of experiments, the long-term half-life h_L is effectively set to infinity by defining V to be the variance of z_k, so that $\overline{z}$ is the mean state of a unit. This produces results which are not noticeably different from the method described above in which h_L is large but finite. As stated above, each of the 1000 stereo pairs had a disparity determined by a circular array of 1000 sinusoidally varying disparity values. The initial weights values were drawn from a uniform distribution between ± 0.3.

There are a total of two microcircuit characteristics to set; the number of hidden units, and the half-life h_S. (The model is insensitive to the value of h_L provided $h_L \gg h_S$. For the task described here, a minimum of three hidden units is required (see (Stone, 1995b)). The implications of different values of h_S are discussed below. There is no learning rate to choose because the conjugate gradient method used to maximise F ensures that the learning rate is adjusted automatically.

Results

Performance was measured in terms of the correlation r between the output unit state z_t and the stereo disparity, measured over the set of 1000 input pairs. After 50 epochs $r = 0.939$, and after 800 epochs $r = 0.977$, where an epoch consists of one complete presentation of the learning sequence to the microcircuit.

Recall that the microcircuit has no output target values, so it is not 'told' what the correct output should be for

any input.

An analysis of the structure of the weight vectors of hidden units is beyond the scope of this paper. However, a more detailed analysis of the behaviour of this type of microcircuit is given in (Stone, 1995b, 1995a).

No Hidden Units: As expected for a system which attempts to discover an input/output mapping which is not linearly separable, learning without a hidden layer of units failed to compute disparity. With the output unit connected directly to the input layer, correlations of $|r| > 0.01$ were not exceeded over 10 different simulations with different initial random weights.

Generalisation: If the model has learned disparity (and not some spurious correlate of disparity) then it should generalise to new stereo data sets, *without any learning of these new sets.* Accordingly, the model was tested on two new data sets. For both sets, generalisation was tested using the microcircuit after 800 epochs.

First, a smoothed randomly varying disparity profile with disparities between ± 1 pixels was used to generate a sequence of 1D random dot stereograms. For this sequence, the correlation between microcircuit output and stereo disparity was $r = 0.921$.

Second, test data was derived from a smoothed (Gaussed) version of the stereo pair shown in Figure 5. As with the learning data, the standard deviation of the Gaussian was one dot width, the dot density was 0.167, and each Gaussed array was normalised to have zero mean and unit variance. The right array was identical to the left array, except for a square region which was shifted by a constant disparity d. For display purposes, d in Figure 5 is equal to one dot width, but $d = 0.5$ of a dot width for the microcircuit inputs. This sub-pixel disparity was achieved by using linear interpolation over grey-levels in the array. The stereo pair used to test the microcircuit was therefore identical to that shown in Figure 5, except that the former had a disparity of half a dot width. Each input pair was obtained by reading data from a Gaussed version of the left and right arrays of this stereo pair into the 5-unit upper and lower input rows, respectively, of the microcircuit.

The microcircuit input was scanned across the Gaussed version of the stereo pair, and an output value was obtained for each position. This simulates the action of a single microcircuit at every location in the visual field. Microcircuit outputs from the border were discarded[3], and the resultant 40×40 array of microcircuit outputs were plotted in Figure 5. The correlation between disparity and microcircuit outputs over the resultant 1600 1D input pairs was $r = 0.91$.

Discussion

Disparity and Temporal Smoothness

The model discovers a perceptually salient visual invariance by unsupervised learning from a sequence of images. The stereo disparity task learned is a hyper-acuity task. That is, the amount of disparity is smaller than the width of any single receptor (pixel). This is consistent with psychophysical studies which demonstrate that subjects can discriminate disparities as small as 2 seconds of arc, less than one tenth of the width (30 seconds of arc) of a retinal receptor (Westheimer, 1994). Members of a stereo pair which have a sub-pixel disparity differ in terms of the local slope and curvature of their intensity profiles, and not necessarily in terms of the positions of the peaks and troughs in these profiles. Therefore, detecting disparities of less than one pixel requires more than a simple one-to-one correspondence between the pixels of a stereo pair. The only means available to the model to discover this invariance was the assumption of its temporal smoothness.

The assumption of temporal smoothness was implemented via a time decay constant λ_S with half-life h_S. Consider a model in which each output unit has a different value of h_S. The value h_S implicitly specifies a temporal 'grain size' for a unit, and therefore restricts learning to those parameters which change at a particular rate. At rates of change which are either too high or too low, perceptual events cannot be detected by a given unit. However, if different units have different temporal tuning characteristics (i.e. different half-lives) then perceptual events occurring at a given rate of change can be detected by some sub-population of units. Psychophysical evidence that disparity sensitive neurons may have different spatiotemporal tuning characteristics is presented in (Morgan & Castet, 1995). The value chosen for h_S is important (though not critical) for learning (see (Stone, 1995b)). However, for any 'reasonably smooth' rate of change, some units in a population tuned to different temporal frequencies would learn disparity.

Even if we accept that an invariance which has a 'reasonably smooth' rate of change can be learned by some unit, it might be argued that perceptually salient parameters simply do not vary smoothly over time. In the perceptual world, violations of the smoothness assumption are not hard to find (e.g. the motion of a stone hitting the ground). However, such violations are unlikely to undermine the model's performance. This is because the learning method

[3]Because the microcircuit cannot be centred over a pixel which is less than three pixels from the edge of an array.

requires only that discontinuities in parameter values are *rare*, relative to gradual changes over time. In the example presented in (Stone, 1995b), four discontinuities every 1000 time steps did not disrupt the learning process. Thus, the model requires not that all parameters change smoothly at all times, but, more realistically, that parameters change smoothly *most* of the time. To paraphrase Marr (quoted in the introduction to this paper), "disparity varies smoothly almost every*when*".

Canonical Microfunctions

In terms of the evolution of the neocortex, it makes sense to have a single type of canonical microcircuit which can be used to learn any perceptual invariance. The approach adopted here seeks to delineate functional characteristics of such a circuit in terms of learning perceptual invariances. This microfunctional approach is useful because it permits computational aspects of neocortical microcircuits to be considered, even though the detailed neuroanatomy of such circuits is not known. This represents a compromise between high-level functional models associated with artificial intelligence, and detailed biophysical models.

Note that the approach does *not* require the assumption that all neocortical microcircuits are the same. It requires an assumption that all microcircuits are pre-disposed to learning about particular *types* of properties in their inputs. The neocortex has evolved in a world which has changed little in terms of what constitutes perceptually salient physical entities. Therefore, an economical strategy would be to evolve a canonical microcircuit that uses a single strategy to learn about invariances which remain perceptually important over long periods of evolutionary time. This single strategy can be formalised in terms of a *canonical microfunction* (of which F may be an example).

Important aspects of the microfunctional framework are that it requires models which are implemented as artificial neuronal networks, that these networks learn without the aid of a teacher, and that they are (broadly speaking) functionally consistent with the capabilities of neuronal systems. The term "functionally consistent" is not intended to refer to the particular learning method used, but rather, to the unsupervised nature of the learning method, and to the final input/output mapping learned by the model.

It is possible to decouple a given putative canonical strategy (e.g. maximise temporal smoothness) from the microfunction used to implement that strategy; for every strategy, there are many microfunctions which can be used to implement it. Similarly, it is possible to decouple a given microfunction from the learning method used to maximise that function. Thus, for every learning algorithm which learns a given mapping[4], there exist many others which can learn this mapping in a manner more consistent with the known neurophysiology. An example of this was given by a pair of papers (Zipser & Anderson, 1988; Mazzoni, Anderson, & Jordan, 1991). The receptive field properties of parietal neurons were simulated using the "biologically implausible" backpropagation method (Zipser & Anderson, 1988). Later, similar results were obtained using a more biologically plausible learning method (Mazzoni et al., 1991).

The learning method described above is consistent with, but is not determined by, the 'temporal smoothness' strategy. It is intended that this microfunctional approach will yield other strategies which are more general in application than that described here. At the very least, it is intended that this approach will yield a series of microfunctions which embody the 'temporal smoothness' assumption in a manner which is increasingly consistent with the known function *and* structure of neocortical microcircuits.

Conclusion

Conventional low-level computer vision techniques rely upon the assumption that a parameter value is invariant over some region of *space*(see (Stone, 1992)). The model described in this paper assumes that perceptually salient parameters vary slowly over *time*. When presented with a sequence of images, the model discovered precisely those parameters which describe the behaviour of the imaged surface through time.

Temporal smoothness is a fundamental property of the perceptual world. Given a sequence of inputs, any learning system that did not take advantage of the temporal smoothness of perceptual invariances implicit in that sequence would be discarding a powerful and general heuristic for discovering perceptually salient properties of the physical world.

Acknowledgements: Thanks to Nikki Hunkin and Julian Budd for comments on this paper, and to Alistair Bray for useful discussions. Thanks also to Raymond Lister, David Willshaw and Tom Collett for discussions on the learning method presented here. This research was supported by a Joint Council Initiative grant awarded to Jim Stone, Tom Collett and David Willshaw.

[4]By maximising a given microfunction.

Appendix: The Learning Algorithm

The learning algorithm relies upon batch update of a weight vector $\mathbf{w}$, which contains all weights in the network. At each time step t, a stereo pair is presented at the input layer, and the derivative of F with respect to every weight in the network is computed and added to a cumulative weight derivative vector $\nabla F_{\mathbf{w}}$. This derivative vector is used to update $\mathbf{w}$ only after all $T = 1000$ stereo pairs have been presented at the input units. The same set of stereo pairs is repeatedly presented in the same order during learning. Storage requirements are minimal because all quantities required for learning can be computed incrementally.

Notation: Units in the input, hidden and output layers are indexed by subscripts i, j and k, respectively. For example, a weight which connects hidden unit u_j to output unit u_k is denoted w_{jk}. The state z_{kt} of u_k at time t is:

$$z_{kt} = \sum_j w_{jk}\, z_{jt} \tag{5}$$

Where z_{jt} is the state of u_j. Input and output layer units have linear activation functions, whereas hidden units have non-linear ($tanh$) activation functions. For such a unit u_j, its output z_j is the hyperbolic tangent of its input:

$$z_{jt} = tanh\left(\sum_i w_{ij}\, z_{it}\right) \tag{6}$$

Where w_{ij} is a weight connecting input unit u_i to hidden unit u_j, and z_{it} is the state of u_i at time t. Weights connecting input to hidden units, and hidden to output units, are referred to as *lower* and *upper* weights, respectively.

The function to be maximised is $F = log\ V/U$, where:

$$\begin{aligned} U &= 1/2 \sum_{t=1}^{T} (\tilde{z}_{kt} - z_{kt})^2 \\ V &= 1/2 \sum_{t=1}^{T} (\overline{z}_{kt} - z_{kt})^2 \end{aligned}$$

V is the long-term variance of z_k, U is the short-term variance of z_k, and T is the period over which they are defined. Both V and U are defined in terms of exponentially weighted means of z_k. The weighted means $\tilde{z}_k$ and $\overline{z}_k$ differ only in terms of their respective exponential rates of decay:

$$\tilde{z}_{kt} = \lambda_S\, \tilde{z}_{k(t-1)} + (1 - \lambda_S)\, z_{k(t-1)} \quad : 0 \leq \lambda_S \leq 1 \tag{7}$$

$$\overline{z}_{kt} = \lambda_L\, \overline{z}_{k(t-1)} + (1 - \lambda_L)\, z_{k(t-1)} \quad : 0 \leq \lambda_L \leq 1 \tag{8}$$

Where λ_S and λ_L have half-lives of h_S and h_L, respectively, with $h_L \gg h_S$. The formula obtaining a value of λ for a given half-life h is $\lambda = 2^{1/h}$. (Note that z_{kt-1} contributes to $\tilde{z}_{kt}$ and $\overline{z}_{kt}$, but not to $\tilde{z}_{kt-1}$ and $\overline{z}_{kt-1}$).

Evaluating $\partial F/\partial w$:

The derivative of F with respect to any weight w is:

$$\frac{\partial F}{\partial w} = \frac{1}{V}\frac{\partial V}{\partial w} - \frac{1}{U}\frac{\partial U}{\partial w} \tag{9}$$

The identical form of U and V permits us to derive $\partial U/\partial w$ for lower and upper weights, from which corresponding equations for V can be obtained by substitution.

The incremental computation of U up to time t is:

$$U(t) = U(t-1) + 1/2\,(\tilde{z}_{kt} - z_{kt})^2 \tag{10}$$

Therefore the derivative of $U(t)$ with respect to any weight w is:

$$\frac{\partial U(t)}{\partial w} = \frac{\partial U(t-1)}{\partial w} + (\tilde{z}_{kt} - z_{kt}) \left(\frac{\partial \tilde{z}_{kt}}{\partial w} - \frac{\partial z_{kt}}{\partial w} \right) \tag{11}$$

Where, from (7):

$$\frac{\partial \tilde{z}_{kt}}{\partial w} = \lambda_S \frac{\partial \tilde{z}_{kt-1}}{\partial w} + (1 - \lambda_S) \frac{\partial z_{kt-1}}{\partial w} \tag{12}$$

Thus, the incremental computation of (11) depends upon evaluation of $\partial z_{kt}/\partial w$ in (11) and $\partial z_{k(t-1)}/\partial w$ in (12), for both upper and lower weights. Equations for $\partial z_{kt}/\partial w$ (for upper and lower weights) will be derived next, from which equations for $\partial z_{kt-1}/\partial w$ can be obtained by substitution.

Evaluating $\partial z_{kt}/\partial w_{jk}$:

In the case of an upper weight w_{jk} projecting to an output unit u_k:

$$z_{kt} = \sum_j w_{jk}\, z_{jt} \tag{13}$$

Where z_{jt} is the state of the jth hidden unit at time t, so that:

$$\frac{\partial z_{kt}}{\partial w_{jk}} = z_{jt} \tag{14}$$

Evaluating $\partial z_{kt}/\partial w_{ij}$:

Equation (13) can be re-written in terms of a lower weight w_{ij} which projects to a hidden unit u_j:

$$z_{kt} = \sum_j w_{jk}\, tanh \left(\sum_i w_{ij}\, z_{it} \right) \tag{15}$$

Where z_{it} is the state of the ith input unit at time t. The derivative of z_{kt} with respect to w_{ij} is:

$$\frac{\partial z_{kt}}{\partial w_{ij}} = w_{jk}(1 - z_{jt}^2)\, z_{it}$$

For an upper weight, this yields:

$$\frac{\partial U(t)}{\partial w_{jk}} = \frac{\partial U(t-1)}{\partial w_{jk}} + (\tilde{z}_{kt} - z_{kt}) \left[\lambda_S \frac{\partial \tilde{z}_{kt-1}}{\partial w_{jk}} + (1 - \lambda_S) z_{jt-1} - z_{jt} \right]$$

Where $\partial \tilde{z}_{kt-1}/\partial w$ is recursively defined as in (12). The corresponding derivative for a lower weight is:

$$\frac{\partial U(t)}{\partial w_{ij}} = \frac{\partial U(t-1)}{\partial w_{ij}} + (\tilde{z}_{kt} - z_{kt}) \left[\lambda_S \frac{\partial \tilde{z}_{kt-1}}{\partial w_{ij}} + w_{jk} \left[(1 - \lambda_S)\, (1 - z_{jt-1}^2) z_{it-1} - (1 - z_{jt}^2) z_{it} \right] \right]$$

Thus (11) can be incrementally evaluated up to any time t. Corresponding equations for $\partial V/\partial w$ can be obtained by substitution in the derivation of $\partial U/\partial w$. The quantities U and V are by definition simple to compute incrementally. Therefore, (9) can be computed on-line. For results presented in this paper, weights were adapted only after weight derivatives obtained with 1000 inputs had been accumulated. On-line weight update is possible, if, at each time step t, the cumulative values of $U(t)$ and $V(t)$ are good estimates of $U(T)$ and $V(T)$, respectively. This was achieved (for a different learning task) in (Stone & Bray, 1995) by defining $U(t)$ and V(t) as exponentially weighted moving averages:

$$U(t) = \gamma U(t-1) + 0.5\,(1 - \gamma)(\tilde{z}_t - z_t)^2 \tag{16}$$

$$V(t) = \gamma V(t-1) + 0.5\,(1 - \gamma)(\overline{z}_t - z_t)^2 \tag{17}$$

Where $\gamma \gg \lambda_L$.

Reference

Barlow, H. (1985). Cerebral cortex as a model builder. In Rose, D., & Dobson, V. (Eds.), *Models of the Visual Cortex*, pp. 37–46. John Wiley, New York.

Becker, S. (1992). Learning to categorize objects using temporal coherence. *Neural Information Processing Systems*, 361–368.

Becker, S., & Hinton, G. (1992). Self-organizing neural network that discovers surfaces in random-dot stereograms. *Nature*, *335*, 161–163.

Bienstock, E., Cooper, L., & Munro, P. (1982). Theory for the development of neuron selectivity: orientation specificity and binocular interaction in visual cortex. *Journal of Neuroscience*, *2*, 32–48.

Creutzfeldt, O. D. (1978). The neocortical link: Thoughts on the generality of structureand function of the neocortex. In Brazier, M., & Petsche, H. (Eds.), *Architectonics of the Cerebral Cortex*. Raven Press, New York.

Douglas, J., Martin, K., & Nelson, J. (1993). The neurobiology of primate vision. *Bailliere's Clinical neurology*, *2*(2), 191–225.

Douglas, R., Martin, K., & Whitteridge, D. (1989). A canonical microcircuit for neocortex. *Neural Computation*, *1*, 480–488.

Ebdon, M. (1993). Is the cerebral neocortex a uniform cognitive architecture?. *Mind and Language*, *8*(3), 369–403.

Foldiak, P. (1991). Learning invariance from transformation sequences. *Neural Computation*, *3*(2), 194–200.

Gibson, J. (1979). *The Ecological Approach to Visual Perception*. Boston: Houghton Mifflin.

Kohonen, T. (1984). Self-organization and associative memory. *Springer Verlag, New York*.

Marr, D. (1970). A theory for cerebral neocortex. *Proc Roy Soc London B*, *176*, 161–234.

Marr, D. (1982). *Vision*. Freeman, New York.

Mazzoni, P., Anderson, R., & Jordan, M. (1991). A more biologically plausible learning rule for neural networks. *Proc Natl Acad Sciences*, *88*, 4433–4437.

Métin, C., & Frost, D. (1989). Visual responses of neurons in somatosensory cortex of hamsters with experimentally induced retinal projections to somatosensory thalamus. *Proceedings National Academy of Sciences USA*, *86*, 357–361.

Mitchison, G. (1991). Removing time variation with the anti-hebbian differential synapse. *Neural Computation*, *3*, 312–320.

Morgan, M., & Castet, E. (1995). Stereoscopic depth perception at high velocities. *Nature*, *378*, 380–383.

Oja, E. (1982). A simplified neuron model as a principal component analyzer. *Journal of Mathematical Biology*, pp. 267–273.

Phillips, W., Kay, J., & Smyth, D. (1995). The discovery of structure by multi-stream networks of local processors with contextual guidance. *Network*, *6*, 225–246.

Roe, A., Pallas, S., Hahm, J., & Sur, M. (1990). A map of visual space induced in primary auditory-cortex. *Science*, *250*(4982), 818–820.

Rumelhart, D., Hinton, G., & Williams, R. (1986). Learning representations by back-propagating errors. *Nature*, *323*, 533–536.

Schraudolph, N., & Sejnowski, T. (1991). Competitive anti-hebbian learning of invariants. *NIPS4*, 1017–1024.

Sejnowski, T. (1977). Storing covariance with nonlinearly interacting neurons. *Journal of Mathematical Biology*, *4*(4), 303–321.

Stone, J. V. (1992). Shape from local and global analysis of texture. *Philosophical Transactions Royal Society London(B)*, *339*(1287), 53–65.

Stone, J. V. (1995a). Hierarchical learning of visual invariances via spatio-temporal constraints. In *Int. Conference on Neural Networks, Cambridge*, pp. 110–115.

Stone, J. V. (1995b). Learning perceptually salient visual parameters through spatio-temporal smoothness constraints. *Neural Computation, (Accepted).*

Stone, J. V., & Bray, A. (1995). A learning rule for extracting spatio-temporal invariances. *Network, 6*(3), 1–8.

Szenátgoathai, J. (1978). The neuron network of the cerebral cortex: a functional approach. *Proc Royal Society London (B), 201*, 219–248.

Westheimer, G. (1994). The ferrier lecture, 1992. seeing depth with two eyes: Stereopsis. *Proc Royal Soc London, B, 257*, 205–214.

Williams, P. (1991). A Marquardt algorithm for choosing the step-size in backpropagation learning with conjugate gradients. Cognitive science research paper CSRP 229, University of Sussex.

Zemel, R., & Hinton, G. (1991). Discovering viewpoint invariant relationships that characterize objects. *Technical Report, Dept. of Computer Science, University of Toronto, Toronto, ONT MS5 1A4.*

Zipser, D., & Anderson, A. (1988). A back-propagation network that simulates response properties of a subset of posterior parietal neurons. *Nature, 331*, 679–684.