

# INFORMATION PROCESSING WITH POPULATION CODES

*Alexandre Pouget\**, *Peter Dayan*‡ and *Richard Zemel*§

Information is encoded in the brain by populations or clusters of cells, rather than by single cells. This encoding strategy is known as population coding. Here we review the standard use of population codes for encoding and decoding information, and consider how population codes can be used to support neural computations such as noise removal and nonlinear mapping. More radical ideas about how population codes may directly represent information about stimulus uncertainty are also discussed.

## NEURONAL NOISE

The part of a neuronal response that cannot apparently be accounted for by the stimulus. Part of this factor may arise from truly random effects (such as stochastic fluctuations in neuronal channels), and part from uncontrolled, but non-random, effects.

*\*Department of Brain and Cognitive Sciences, Meliora Hall, University of Rochester, Rochester, New York 14627, USA.*

*‡Gatsby Computational Neuroscience Unit, Alexandra House, 17 Queen Square, London WC1N 3AR, UK. §Department of Computer Science, University of Toronto, Toronto, Ontario M5S 1A4, Canada. e-mails: alex@bcs.rochester.edu; dayan@gatsby.ucl.ac.uk; zemel@cs.toronto.edu*

A fundamental problem in computational neuroscience concerns how information is encoded by the neural architecture of the brain. What are the units of computation and how is information represented at the neural level? An important part of the answers to these questions is that individual elements of information are encoded not by single cells, but rather by populations or clusters of cells. This encoding strategy is known as population coding, and turns out to be common throughout the nervous system.

The ‘place cells’ that have been described in rats are a striking example of such a code. These neurons are believed to encode the location of an animal with respect to a world-centred reference frame in environments such as small mazes. Each cell is characterized by a ‘place field’, which is a small confined region of the maze that, when occupied by the rat, triggers a response from the cell<sup>1,2</sup>. Place fields within the hippocampus are usually distributed in such a way as to cover all locations in the maze, but with considerable spatial overlap between the fields. As a result, a large population of cells will respond to any given location. Visual features, such as orientation, colour, direction of motion, depth and many others, are also encoded with population codes in visual cortical areas<sup>3,4</sup>. Similarly, motor commands in the motor cortex also rely on population codes<sup>5</sup>, as do the nervous systems of invertebrates such as leeches or crickets<sup>6,7</sup>.

One key property of the population coding strategy is that it is robust — damage to a single cell will not have a catastrophic effect on the encoded representa-

tion because the information is encoded across many cells. However, population codes turn out to have other computationally desirable properties, such as mechanisms for NOISE removal, short-term memory and the instantiation of complex, NONLINEAR FUNCTIONS. Understanding the coding and computational properties of population codes has therefore become one of the main goals of computational neuroscience. We begin this review by describing the standard model of population coding. We then consider recent work that highlights important computations closely associated with population codes, before finally considering extensions to the standard model suggested by studies into more complex inferences.

The standard model

We will illustrate the properties of the standard model by considering the cells in visual area MT of the monkey. These cells respond to the direction of visual movement within a small area of the visual field.

The typical response of a cell in area MT to a given **motion** stimulus can be decomposed into two terms. The first term is an average response, which typically takes the form of a GAUSSIAN FUNCTION of the direction of motion. The second term is a noise term, and its value changes each time the stimulus is presented. Note that in this article we will refer to the ‘response’ to a stimulus as being the number of action potentials (spikes) per second measured over a few hundred milliseconds during the presentation of the stimulus. This measurement is also known as the spike count rate or the response

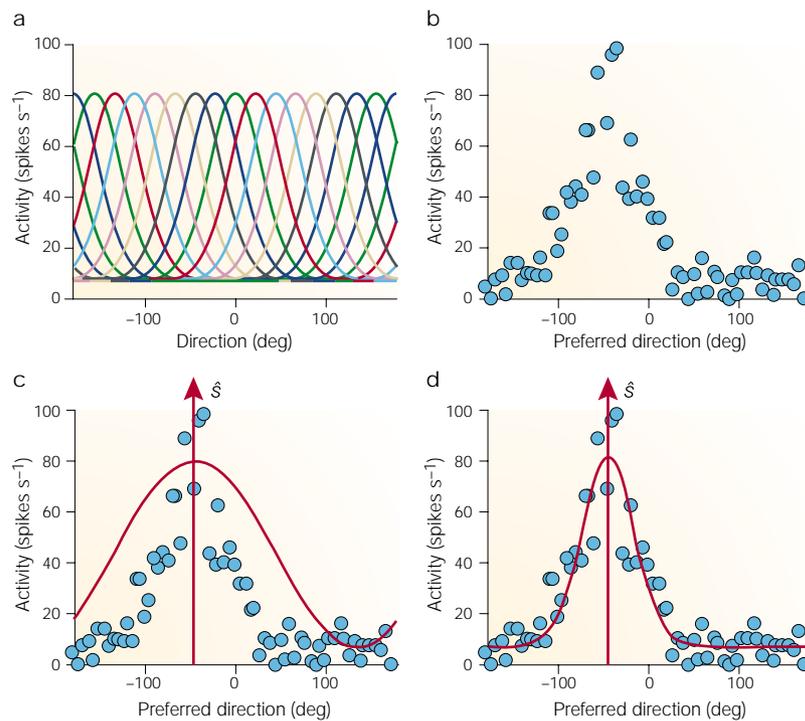


Figure 1 | **The standard population coding model.** **a** | Bell-shaped tuning curves to direction for 16 neurons. **b** | A population pattern of activity across 64 neurons with bell-shaped tuning curves in response to an object moving at  $-40^\circ$ . The activity of each cell was generated using EQN 1, and plotted at the location of the preferred direction of the cell. The overall activity looks like a ‘noisy’ hill centred around the stimulus direction. **c** | Population vector decoding fits a cosine function to the observed activity, and uses the peak of the cosine function,  $\hat{s}$ , as an estimate of the encoded direction. **d** | Maximum likelihood fits a template derived from the tuning curves of the cells. More precisely, the template is obtained from the noiseless (or average) population activity in response to a stimulus moving in direction  $s$ . The peak position of the template with the best fit,  $\hat{s}$ , corresponds to the maximum likelihood estimate, that is, the value that maximizes  $P(\mathbf{r} | s)$ .

rate. Other aspects of the response, such as the precise timing of individual spikes, might also have a function in coding information, but here we shall focus on properties of response rates, because they are simpler and are better understood. (For reviews of coding through spike timing, see REFS 1–3.)

More formally, we can describe the response of a cell using an encoding model<sup>4</sup>. In one simple such model,

$$r_i = f_i(s) + n_i \quad (1)$$

In EQN 1,  $f_i(s)$ , the average response, is the TUNING CURVE for the encoded variable  $s$  (the direction) and  $n_i$  is the noise. The letter  $i$  is used as an index for the individual neuron; it varies from 1 to  $n$ , where  $n$  is the total number of neurons under consideration. We use the notation  $\mathbf{r}$  to refer to all the activities and  $\mathbf{f}(s)$  for their means. Here,  $\mathbf{r}$  and  $\mathbf{f}(s)$  are vectors with  $n$  components, each of which corresponds to one neuron. Experimental measurements have shown that the noise term ( $n_i$ ) can typically be characterized as following a normal distribution whose variance is proportional to the mean value,  $f_i(s)$  (REF. 5). When  $f_i(s)$  is a gaussian, it can be written as:

$$f_i(s) = ke^{-(s-s_i)^2/2\sigma^2} \quad (2)$$

In EQN 2,  $s_i$  is the direction (the preferred direction) that triggers the strongest response from the cell,  $\sigma$  is the width of the tuning curve,  $s - s_i$  is the angular difference (so if  $s = 359^\circ$  and  $s_i = 2^\circ$ , then  $s - s_i = 3^\circ$ ), and  $k$  is a scaling factor. In this case, all the cells in the population share a common tuning curve shape, but have different preferred directions,  $s_i$  (FIG. 1a). Many population codes involve bell-shaped tuning curves like these.

The inclusion of the noise in EQN 1 is important because neurons are known to be noisy. For example, a neuron that has an average firing rate of 20 Hz for a stimulus moving at  $90^\circ$  might fire at only 18 Hz on one occasion for a particular  $90^\circ$  stimulus, and at 22 Hz on another occasion for exactly the same stimulus<sup>5</sup>. Several factors contribute to this variability, including uncontrolled or uncontrollable aspects of the total stimulus presented to the monkey, and inherent variability in neuronal responses. In the standard model, these are collectively considered as random noise. The presence of this noise causes important problems for information transmission and processing in cortical circuits, some of which are solved by population codes. It also means that we should be concerned not only with how the brain computes with population codes, but also how it does so reliably in the presence of such stochasticity.

#### Decoding population codes

In this section, we shall address the following question: what information about the direction of a moving object is available from the response of a population of neurons? Let us take a hypothetical experiment. Imagine that we record the activity of 64 neurons from area MT, and that these neurons have spatially overlapping receptive fields. We assume that all 64 neurons have the same tuning curve shape with preferred directions that are uniformly distributed between  $0^\circ$  and  $360^\circ$  (FIG. 1a). We then present an object moving in an unknown direction,  $s$ , and we assume that the responses are generated according to EQN 1. If we plot the responses,  $\mathbf{r}$ , of the 64 neurons as a function of the preferred direction of each cell, the resulting pattern looks like a noisy hill centred in the vicinity of  $s$  (FIG. 1b). The question can now be rephrased as follows: what information about the direction  $s$  of the moving object is available from the observed responses,  $\mathbf{r}$ ?

The presence of noise makes this problem challenging. To recover the direction of motion from the observed responses, we would like to assess for each cell,  $i$ , the exact contribution of its tuning curve,  $f_i(s)$ , to its observed response. However, on a single trial, it is impossible to apportion signal and noise in the response. For instance, if a neuron fires at 54 Hz on one trial, the contribution of the tuning curve could be 30 Hz, with 24 Hz due to noise. However, the contributions could just as easily be 50 Hz and 4 Hz, respectively. Nevertheless, given some knowledge of the noise, it is possible to assess probabilities for these unknowns. If the noise follows a normal distribution with a mean of zero and a neuron fires at 54 Hz on a particular trial, it is more likely that the contribution of the tuning curve in our example is 50 Hz rather than 30 Hz.

#### NONLINEAR FUNCTION

A linear function of a one-dimensional variable (such as direction of motion) is any function that looks like a straight line, that is, any function that can be written as  $y = ax + b$ , where  $a$  and  $b$  are constant. Any other functions are nonlinear. In two dimensions and above, linear functions correspond to planes and hyperplanes. All other functions are nonlinear.

#### GAUSSIAN FUNCTION

A bell-shaped curve. Gaussian tuning curves are extensively used because their analytical expression can be easily manipulated in mathematical derivations.

#### TUNING CURVE

A tuning curve to a feature is the curve describing the average response of a neuron as a function of the feature values.

## Box 1 | The Bayesian decoder

Bayes rule is used to decode the response,  $r$ , to form a posterior distribution over  $s$ :

$$P(s|r) = \frac{P(r|s)P(s)}{P(r)} \quad (6)$$

In EQN 6,  $P(r|s)$  is called the likelihood function and  $P(s)$  and  $P(r)$  are called the priors over  $s$  and  $r$ . The first two terms can be readily estimated from the experimental data:  $P(r|s)$  from the histogram of responses,  $r$ , to a given stimulus value,  $s$ , and  $P(s)$  from the stimulus values themselves. Then  $P(r)$  can be determined according to:

$$P(r) = \int_s P(r|s)P(s)ds \quad (7)$$

In EQN 7,  $P(s)$  is important, as it captures any knowledge we may have about  $s$  before observing the responses. For instance, in a standard, two-alternative, forced choice task, in which an animal has to distinguish just two values (say  $s = 0^\circ$  and  $s = 180^\circ$ ) each of which is equally likely on a trial,  $P(s)$  is just a sum of two blips (called weighted delta functions) at these values of  $s$ . The Bayesian decoder would correctly state that the probability that  $s$  took any other value on the trial is zero. The likelihood function,  $P(r|s)$ , can be estimated from multiple trials with stimulus value  $s$ . If all 64 cells are recorded simultaneously, the likelihood function can be directly estimated from the histogram of responses  $r$ . If not enough data are available for such a direct estimation, simplifying assumptions can be used to estimate the likelihood function. For example, if we assume that the noise is independent across neurons (the amount of noise in the firing rate of one neuron does not affect the amount of noise in others),  $P(r|s)$  can be factorized as a product of the individual  $P(r_i|s)$  terms, each of which can be measured (beforehand) separately for each of the neurons in our pool.

Thus, for all neurons, we can assess the probability over the mean responses,  $f(s)$ , given the observed responses,  $r$ . We can then use this probability to recover the probability distribution,  $P(s|r)$ , which reports how likely each direction is, given the observed responses. This distribution is known as the posterior distribution. A Bayesian decoding model<sup>6–9</sup> provides a method for combining the information from all the cells, giving rise to a single posterior distribution,  $P(s|r)$  (BOX 1).

#### Maximum likelihood and other estimates

What can we do once we have recovered the probability distribution,  $P(s|r)$ ? If further computations are to be done over  $s$  (such as computing the probable time and position of impact of a rapidly approaching object), it is best to preserve the probability density and to do the computations over it in its entirety (as is common in Bayesian settings<sup>10,11</sup>). Often, however, we need a single value or estimate of the direction,  $s$ , on each trial. There are several ways to select this estimate. One choice is the direction with the highest probability density, that is, the direction maximizing  $P(s|r)$ . This estimate is known as the maximum a posteriori (MAP) estimate<sup>6–9</sup>. Alternatively, the mean direction for the posterior density,  $P(s|r)$ , can be used. Another solution is to select the direction that maximizes the likelihood function,  $P(r|s)$ <sup>12,13</sup>. This is called the maximum likelihood (ML) estimate. The ML and MAP estimates are the same in cases for which the prior  $P(s)$  is a flat function, that is, situations in which we have no prior knowledge about  $s$ . Note that it is not necessary to calculate  $P(r)$  in EQN 7 to determine either the MAP or ML estimates. For large populations of neurons, the ML estimate is often optimal in the sense that, first, it is right on average (a

property called unbiasedness), and second, it has the minimum variance among unbiased estimators. In other words, were we to present the same stimulus on many trials, the average estimate over all the trials would equal the true direction of motion of the stimulus, and the variance would be as small as possible<sup>12–14</sup>.

Although ML is often the optimal decoding method<sup>13</sup>, it requires substantial data, as a precise measurement of the tuning curves and noise distributions of each neuron is needed. When this information is not available, an alternative approach to decoding can be used that relies only on the preferred direction of each cell. Such methods are known as voting methods, because they estimate direction by treating the activity of each cell on each trial as a vote for the preferred direction of the cell (or for a value related to the preferred direction of the cell). The well-known population vector estimator belongs to this category of estimators<sup>15</sup>. Further details about these voting methods can be found in recent reviews<sup>7,9</sup>. Using such methods can be sub-optimal in some situations<sup>16</sup>, but, because they require limited knowledge of the properties of the neuron, they have been extensively used in experimental contexts.

It is worth mentioning that both ML and population vector methods are template-matching procedures, which effectively slide an idealized response curve (the template) across the actual responses to find the best fit. The population vector method fits a cosine function through the noisy hill and uses its phase as an estimate of actual direction<sup>13,14</sup> (FIG. 1c). Likewise, ML fits a template and uses the peak position as an estimate of the direction<sup>14</sup> (FIG. 1d). However, the template used for ML is derived directly from the tuning curves of the cells, which is why it is optimal (see FIG. 1d for details).

Interestingly, in such template-matching procedures, the neurons contributing the most to the final estimate (in the sense of those that have a greater effect on the final estimate if their activities are modified) are not the ones whose tuning curves peak at the current estimate, that is, the most active neurons. Instead, the most critical neurons are those for which the current estimate lies on the flank of the tuning curve, that is, the part of the tuning curve with the highest derivative<sup>12,13,17</sup>. Although this idea may sound quite counterintuitive for the population vector, as neurons vote according to how active they are, it is nevertheless true. This property is consistent with the observation that, in fine discrimination tasks, human subjects also seem to rely more on those neurons with high derivatives, and not those that are the most active<sup>18</sup>.

As we shall see, thinking about estimation in terms of template matching also makes it easier to understand how these methods can be implemented in neural hardware.

#### Computational processing

Most of the early computational focus on population codes centred on the observation that they offer a balance between coding a stimulus with a single cell and coding a stimulus across many cells. Single cell encoding

**SYMMETRIC LATERAL CONNECTIONS**

Lateral connections are formed between neurons at the same hierarchical level. For instance, the connections between cortical neurons in the same area and same layer are said to be lateral. Lateral connections are symmetric if any connection from neuron *a* to neuron *b* is matched by an identical connection from neuron *b* to neuron *a*.

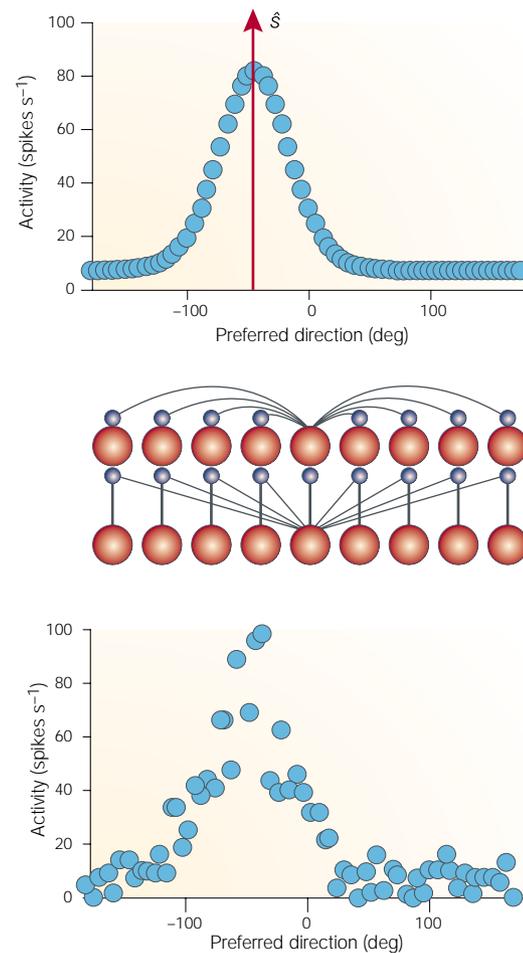
**HYPERCOLUMN**

In the visual cortex, an orientation hypercolumn refers to a patch of cortex containing neurons with similar spatial receptive fields but covering all possible preferred orientations. This concept can be generalized to other visual features and to other sensory and motor areas.

**OPTIMAL INFERENCE**

This refers to the statistical computation of specifically extracting all the information implied about the stimulus from the (noisy) activities of the population. Ideal observers make optimal inferences.

strategies lead to problems with noise, robustness and the sheer number of cells required. However, coding with many cells is often wasteful, requires complicated decoding computations, and has problems in cases such as transparency, when many directions of motions coexist at the same point in visual space. More recently, attention has turned to other aspects of population codes that are inspired by the idea of connections between population-coding units. These ideas are discussed in the following section.



**Figure 2 | A neural implementation of a maximum likelihood estimator.** The input layer (bottom), as in FIG. 1a,b, consists of 64 units with bell-shaped tuning curves whose activities constitute a noisy hill. This noisy hill is transmitted to the output layer by a set of feedforward connections. The output layer forms a recurrent network with lateral connections between units (we show only one representative set of connections and only nine of the 64 cells). The weights in the lateral connections are determined such that, in response to a noisy hill, the activity in the output layer converges over time onto a smooth hill of activity (upper graph). In essence, the output layer fits a smooth hill through the noisy hill, just like maximum likelihood (FIG. 1d). Deneve *et al.*<sup>14</sup> have shown that, with proper choice of weights, the network is indeed an exact implementation of, or a close approximation to, a maximum likelihood estimator. The network can also be thought of as an optimal nonlinear noise filter, as it essentially removes the noise from the noisy hill.

[Movie online](#)

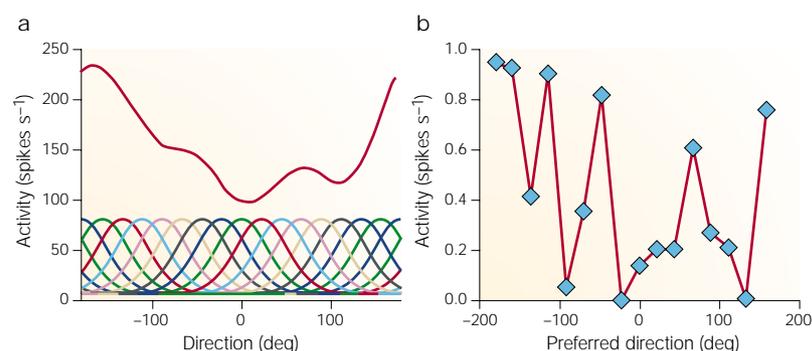
**Neural decoding**

As we saw above, the variability of the neuronal responses to a given stimulus is the central obstacle for decoding methods. Sophisticated methods such as ML may form accurate estimates in the face of such noise, but one might wonder how such complex computations could be carried out in the brain. A possibility is for the population itself to actively remove some of the noise in its response.

Recently, Deneve *et al.*<sup>14</sup> have shown how optimal ML estimation can be done with a biologically plausible neural network. Their neural network is composed of a population of units with bell-shaped tuning curves (FIG. 2). Each unit uses a nonlinear activation function — its input–output function — and is connected to its neighbours by SYMMETRIC LATERAL CONNECTIONS. In essence, this circuit corresponds to a cortical HYPERCOLUMN. Deneve *et al.*<sup>14</sup> used simulations and mathematical analysis to show that this class of network can be tuned to implement a ML estimator, as long as the network admits smooth hills centred on any point on the neuronal array as stable states of activity. This means that in response to a noisy hill, such as the one shown in FIG. 1b, the activity of the network should converge over time to a smooth hill. Once the smooth hill is obtained, its peak position can be interpreted as an estimate of the direction of motion (FIG. 2). The process of finding the smooth hill closest to a noisy hill is reminiscent of the curve fitting procedure used in ML decoding (compare FIG. 1d and FIG. 2), which is why the network can be shown to implement OPTIMAL INFERENCE. This illustrates one of the advantages of population codes, namely their ability to easily implement optimal estimators such as maximum likelihood.

Note that if the hill were always to stabilize at the same position (say 90°) for any initial noisy activities, the network would be a poor estimator. Indeed, this would entail that the network always estimate the direction to be 90° whether the true direction was in fact 35°, 95° or any other value. It is therefore essential that smooth hills centred on any point on the neuronal array be stable activity states, as, in most cases, any value of the stimulus *s* is possible. Networks with this property are called line (or surface) attractor networks. They are closely related to more conventional memory attractor networks, except that, instead of having separated discrete attractors, their attractors are continuous in this space of activities. Interestingly, the same architecture has been used to model short-term memory circuits for continuous variables such as the position of the head with respect to the environment<sup>19</sup>, and similar circuits have been investigated as forming the basis of the neural integrator in the oculomotor system<sup>20</sup>.

An important issue is whether the network used by Deneve *et al.* is biologically plausible. At first glance the answer seems to be that it is not. For example, the tuning curves of the units differ only in their preferred direction, in contrast to the actual neurons in area MT, area MST and elsewhere, whose tuning curves also differ in their width and maximum firing rate. Moreover,



**Figure 3 | Function approximation with basis functions.** **a** | The nonlinear function shown in red was obtained by taking a sum over the multicoloured gaussian functions plotted underneath, weighted by the coefficients shown in **b**. A line connecting the coefficient values is shown for visual convenience. Almost any other nonlinear function can be obtained by adjusting the weights assigned to the gaussian basis functions. Note that a population code is precisely a basis set of gaussian functions, as is evident from comparing panel **a** and FIG. 1a. This is one of the properties that makes population codes so appealing from the perspective of computational processing. Note how a smooth nonlinear function emerges from a jagged set of weights. The smoothness and proximity of the basis functions determines how rough the nonlinear function can be.

#### IDENTITY MAPPING

A mapping is a transformation from a variable  $x$  to a variable  $y$ , such as  $y = x^2$ . The identity mapping is the simplest form of such mapping in which  $y$  is simply equal to  $x$ .

#### BASIS SET

In linear algebra, a set of vectors such that any other vector can be expressed in terms of a weighted sum of these vectors is known as a basis. By analogy, sine and cosine functions of all possible frequencies are said to form a basis set.

#### FOURIER TRANSFORM

A transformation that expresses any function in terms of a weighted sum of sine and cosine functions of all possible frequencies. The weights assigned to each frequency are specific to the function being considered and are known as the Fourier coefficients for this function.

#### BACKPROPAGATION

A learning algorithm based on the chain rule in calculus, in which error signals computed in the output layer are propagated back through any intervening layers to the input layer of the network.

#### HEBBIAN LEARNING RULE

A learning rule in which the synaptic strength of a connection is changed according to the correlation in the activities of its presynaptic and postsynaptic sides.

once the network converges to the attractor, the response of the output units becomes deterministic (that is, noiseless), whereas the response of actual neurons always show a near-Poisson variability throughout the cortex<sup>5</sup>. Fortunately, these problems can be easily resolved. Indeed, it is a straightforward step to implement an attractor network of units with distinct tuning curves. It is also simple to add Poisson noise to the activity of the output units; this noise could be removed at the next stage with a similar network. The network shown in FIG. 2 should therefore be thought of as a single step of a recursive process, in which noise is added and removed at each stage over the course of implementing computations.

The neural implementation of maximum likelihood that we have just presented is essentially performing an optimal IDENTITY MAPPING in the presence of noise, as the input and output population codes in the network encode the same variable. This is clearly too limited, as most behaviours require the computation of nonlinear functions of the input stimuli, not just the identity mapping. As we show in the next section, population codes also have exactly the right properties for performing nonlinear mappings.

#### Static nonlinear mappings

Nonlinear mappings are a very general way of characterizing a large range of neural operations<sup>21</sup>. A particular example whose neural instantiation has been the subject of much investigation is sensorimotor transformation, that is, the computation of motor commands from sensory inputs<sup>21,22</sup>.

Population codes are particularly helpful for nonlinear mappings because they provide what is known as a BASIS SET<sup>21</sup>. A basis set is a set of functions such that (almost) all other functions, including, in particular, nonlinear functions, can be obtained by forming linear combinations of the basis functions. The best-known basis set is that composed of cosine and sine functions

of all possible frequencies; this basis set underlies the FOURIER TRANSFORM. Another example of a basis set is a set of gaussian functions with all possible peak positions. This is, at least approximately, what a population code provides, as shown in FIG. 1a. Thus, for almost any function,  $h(s)$ , of a stimulus,  $s$ , there exists a set of weights,  $\{w_i\}_{i=1}^n$  such that:

$$h(s) = \sum_{i=1}^n w_i f_i(s) \quad (3)$$

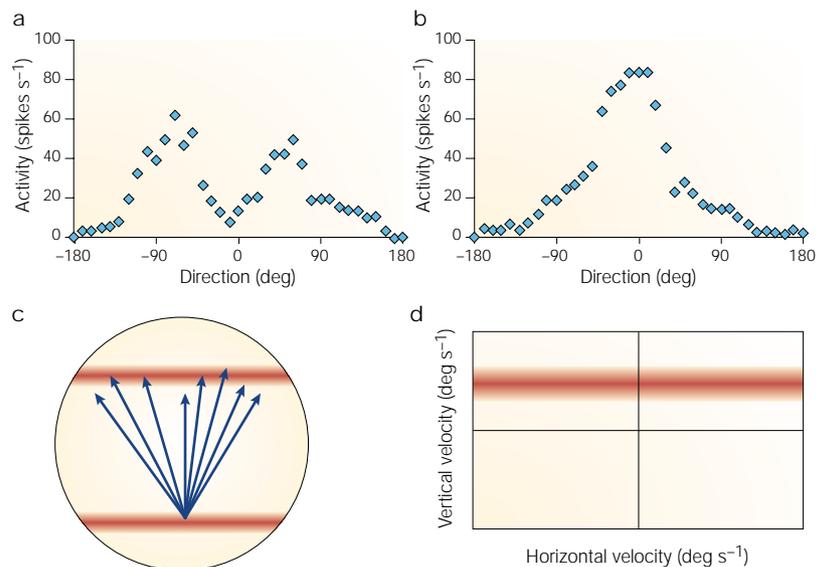
In EQN 3,  $f_i(s)$  is the gaussian tuning curve for neuron  $i$  (EQN 2). FIGURE 3a illustrates how a particular nonlinear function,  $h(s)$ , is obtained by combining gaussian tuning curves with the set of weights shown in FIG. 3b. Strictly speaking, there are many functions that cannot be decomposed as linear combinations over particular basis functions, or, in the gaussian case, cannot be decomposed using only a finite number of peak positions and tuning widths. However, such functions are rarely encountered in the context of mappings such as smooth, sensorimotor transformations.

The basis function approach to nonlinear mappings also works for multiple input variables. For instance, the motor command,  $m$ , to reach to a target is a nonlinear transformation of the retinal location of the target and the current eye position (as well as other postural signals such as head, shoulder and arm position, but we consider only eye position to simplify our discussion). As such, it can be approximated as a linear combination of joint basis functions of retinal location,  $l$ , and eye position,  $e$  (REF. 22):

$$m = \sum_{i=1}^n w_i B_i(l, e) \quad (4)$$

The basis functions in EQN 4 could take the form of two-dimensional gaussian functions over  $l$  and  $e$ , that is, they could take the form of a joint population code for  $l$  and  $e$ . However, this is not the only choice. An alternative option would be to use a product of a gaussian function of  $l$  and a sigmoidal function of  $e$ . Although equivalent from a computational point of view, this alternative is more biologically plausible<sup>22</sup>. Indeed, neurophysiological data show that the selectivity to eye position is sigmoidal rather than gaussian<sup>23,24</sup>.

Basis functions also simplify the learning of nonlinear mappings, which otherwise seem to require powerful, and biologically questionable, learning algorithms such as BACKPROPAGATION<sup>25</sup>. This is the case, for instance, when learning the transformation from retinal location and eye position onto a reaching motor command<sup>26</sup>. Provided that the basis functions  $B_i(l, e)$  of EQN 4 are available, the only free parameters are the linear coefficients  $w_i$ , which can be learned using simple HEBBIAN and DELTA learning rules<sup>27,28</sup>. Of course, one might wonder how the basis functions themselves arise, as these are always nonlinear. Fortunately, this is simpler than it might seem, as the basis functions can be learned using UNSUPERVISED OR SELF-ORGANIZING METHODS, independently of the particular output functions that will eventually be computed. Indeed, there is active interest in unsupervised algorithms that can learn basis



**Figure 4 | Multiplicity and uncertainty in population codes.** Population activity in area MT in response to an image containing two directions of motion. **a** | For direction differences of more than 30°, the population activity is bimodal. **b** | When the two directions differ by less than 30°, the population forms a single hill of activity. Standard decoding techniques such as the one described in FIG. 1c,d would fail to recover the two directions of motion in this case. But psychophysical experiments have revealed that human subjects perceive two distinct directions of motion for direction differences greater than 13°. **c** | The aperture problem. The image of a fuzzy bar moving behind an aperture is consistent with a whole collection of motion vectors that vary in direction and velocity. **d** | The probability density of motion vectors consistent with the image shown in **c**. The standard model of population codes cannot explain how such a density over direction can be represented in the cortex.

**DELTA LEARNING RULE**

A learning rule that adjusts synaptic weights according to the product of the presynaptic activity and a postsynaptic error signal obtained by computing the difference between the actual output activity and a desired or required output activity.

**UNSUPERVISED OR SELF-ORGANIZING METHODS**

An adaptation in which a network is trained to uncover and represent the statistical structure within a set of inputs, without reference to a set of explicitly desired outputs. This contrasts with supervised learning, in which a network is trained to produce particular desired outputs in response to given inputs.

**MOTION TRANSPARENCY**

A situation in which several directions of motion are perceived simultaneously at the same location. This occurs when looking through the windscreen of a car. At each location, the windscreen is perceived as being still while the background moves in a direction opposite to the motion of the car.

function representations<sup>29–32</sup>.

The equivalence between basis functions and population codes is one of the reasons that population codes are so computationally appealing. Several models rely on this property, in the context of object recognition<sup>33</sup>, object-centred representations<sup>34,35</sup> or sensorimotor transformations<sup>22,26–28,36,37</sup>. It should be noted, however, that basis function networks may require large numbers of units. To be precise, the number of units required for a transformation grows exponentially with the number of input variables, a problem known as the curse of dimensionality. There are many ways around this problem, such as using dendritic trees to implement the basis function network, but they lie beyond the scope of this review<sup>21,38</sup>.

**Extensions to the standard model**

Although the standard model of population codes that we have described here has been very successful in helping us to understand neural data and has underpinned computational theories, it is clearly not a complete description of how the brain uses population codes. We now wish to consider two related extensions to the standard model, which focus on the problems of encoding MOTION TRANSPARENCY and motion uncertainty.

**Motion transparency.** This occurs when several directions of motion are perceived simultaneously at the same location. Treue and colleagues<sup>39</sup> studied how

transparent motion stimuli are processed, by using neurophysiological methods to study the behaviour of cells in area MT and psychophysical methods to study the percept of the monkey during the presentation of stimuli that produce transparent motion. Each transparent motion stimulus consisted of randomly located dots within the spatial receptive field of a given cell. The dots jump small distances between successive frames. The motion signal provided by the stimulus can be controlled by manipulating the proportion of the dots that move in any given direction. In one version of the experiment, half the dots move in direction  $x$  and half in direction  $-x$ . Recordings of motion-selective cells in area MT indicated that their responses to this stimulus were closely related to the average of their responses to two stimuli, one with only the single direction  $x$ , and one with only the single direction  $-x$ . What does this imply for the activity across the whole population of cells? For separations with  $2x > 30^\circ$ , the activity pattern is bimodal (FIG. 4a). But for separations less than about  $30^\circ$ , there is a unimodal bump or hill centred at  $0^\circ$ , identical to the bump that would be created by a stimulus with only a single direction of motion at  $0^\circ$ , except that it is a bit broader (FIG. 4b). If this hill were presented to an ML decoding algorithm, such as the recurrent network described above, then it would report that there is just  $0^\circ$  motion in the stimulus (and the network, in fact, would sharpen the hill of activity to make it the same breadth as for stimuli that actually contain single directions of motion). However, for separations with  $2x > 10^\circ$ , both monkeys and humans can correctly extract the two directions of motion from such stimuli. If, as seems likely from other experiments, the activities of area MT cells determine the motion percepts of the monkeys<sup>40</sup>, the standard population-coding model cannot be the whole story<sup>41</sup>.

**Motion uncertainty.** The other concern is motion uncertainty, coming, for instance, from the well-known aperture problem (FIG. 4c). This problem arises because the visual images of the motion of stimuli that vary in only one spatial dimension (such as FULL-FIELD GRATINGS or long bars placed behind apertures) provide information about only one of the two possible dimensions of the direction of motion (ignoring motion in depth). The component of motion in the direction along which the stimulus is constant produces no change in the images, and so information about motion in this direction cannot, in principle, be extracted from the images. Of course, the motion of endpoints is unambiguous if they are visible, and motion can also be integrated across spatially separated parts of extended objects. The issue for population codes is how the uncertainty about the component of the motion in one dimension is represented. One way to code for uncertainty is to excite all cells in a population whose preferred direction of motion is consistent with the visual input at a particular location<sup>42</sup>. For example, FIG. 4d shows the set of directions consistent with the motion shown in FIG. 4c. This would mirror the representation that would be given to a visual input con-

taining many transparent layers of motion in all the relevant directions. However, as the standard population-coding model is incapable of dealing with transparent motion correctly, it would also fail to cope with uncertainty coded in this way. In fact, the standard model cannot deal with any coding for uncertainty because its encoding model ignores uncertainty.

Uncertainty also arises in other ways. For instance, imagine seeing a very low-contrast moving stimulus (such as a black taxicab at night), the details of whose movement are hard to discern. In this case, the population code might be expected to capture uncertainty about the direction of motion in such a way that it could be resolved by integrating information either over long periods of observation, or from other sources, such as audition.

The standard population-coding model cannot represent motion transparency and motion uncertainty because of its implicit assumption that the whole population code is involved in representing just one direction of motion, rather than two, or potentially many. The neurophysiological data indicating that the activity of MT cells to transparency may resemble the average of their responses to single stimuli suggests that EQN 1 should be changed to:

$$r_i = \alpha \int g(s) f_i(s) ds + n \quad (5)$$

In EQN 5,  $g(s)$  is a function that indicates what motions are present or consistent with the image, and  $\alpha$  is a constant (see below). Here, the motions in the image are convolved with the tuning function of the cell to give its output<sup>4,43,44</sup>. The breadth of the population activity conveys information about multiplicity or uncertainty. Recent results are consistent with this form of encoding model<sup>45,46</sup>, but it is possible that a more nonlinear combination method might offer a better model than averaging<sup>47</sup>.

Unfortunately, although this encoding model is straightforward, decoding and other computations are not. In the standard population-coding model, the presence of noise means that only a probability density over the motion direction,  $s$ , can be extracted from the responses,  $r$ . In the model based on EQN 5, the presence of noise means that only a probability density over the whole collection of motions,  $g(s)$ , can be extracted. Of course, in the same way that a single value of  $s$  can be chosen in the standard case, a single function,  $g(s)$  (or even a single aspect of it, such as the two motions that are most consistent with it), can be chosen in the new case<sup>4,44</sup>. So, although we understand how to encode uncertainty with population codes, much work remains to be done to develop a theory of computation with these new codes, akin to the basis function theory for the standard population codes described above.

The multiplier  $\alpha$  in EQN 5 deserves comment, as there is a controversy about the semantics of the decoded motions,  $g(s)$ . This function can describe multiple transparent motions that are simultaneously present, or a single motion of uncertain direction. The

overall level of activity in the population can provide information differentiating between these interpretations, for example if the number of motions is proportional to the total activity. Mechanisms to implement this type of control through manipulations of  $\alpha$  have yet to be defined.

#### Discussion

The theory and analysis of population codes are steadily maturing. In the narrowest sense, it is obvious that the brain must represent aspects of the world using more than just one cell, for reasons of robustness to noise and neuronal mortality. Accordingly, much of the early work on population codes was devoted to understanding how information is encoded, and how it might be simply decoded. As we have seen, more recent work has concentrated on more compelling computational properties of these codes. In particular, recurrent connections among the cells concerned can implement line attractor networks that can perform maximum likelihood noise removal; and population codes can provide a set of basis functions, such that complex, nonlinear functions can be represented simply as a linear combination of the output activities of the cells. Finally, we have considered an extension to the standard model of population encoding and decoding, in which the overall shape and magnitude of the activity among the population is used to convey information about multiple features (as in motion transparency) or feature uncertainty (as in the aperture problem).

We have focused here on population codes that are observed to be quite regular, with features such as gaussian-shaped tuning functions. Most of the theory we have discussed is also applicable to population codes using other tuning functions, such as sigmoidal, linear-rectified, or almost any other smooth nonlinear tuning curves. It also generalizes to codes that do not seem to be as regular as those for motion processing or space, such as the cells in the inferotemporal cortex of monkeys that are involved in representing faces<sup>48,49</sup>. These 'face cells' share many characteristics with population codes; however, we do not know what the natural coordinates might be (if, indeed there are any) for the representation of faces. We can still build decoding models from observations of the activities of the cells, but the computations involving memory and noise removal are harder to understand.

Many open issues remain, including the integration of the recurrent models that remove noise optimally (assuming that there is only one stimulus) with the models allowing multiple stimuli. The curse of dimensionality that arises when considering that cells really code for multiple dimensions of stimulus features must also be addressed. Most importantly, richer models of computation with population codes are required.

#### Links

FURTHER INFORMATION [Computational neuroscience | Gatsby computational neuroscience unit | Motion perception | Alex Pouget's web page | Peter Dayan's web page | Richard Zemel's web page](#)

#### FULL-FIELD GRATING

A grating is a visual stimulus consisting of alternating light and dark bars, like the stripes on the United States flag. A full-field grating is a very wide grating that occupies the entire visual field.

1. Bair, W. Spike timing in the mammalian visual system. *Curr. Opin. Neurobiol.* **9**, 447–453 (1999).
2. Borst, A. & Theunissen, F. E. Information theory and neural coding. *Nature Neurosci.* **2**, 947–957 (1999).
3. Usrey, W. & Reid, R. Synchronous activity in the visual system. *Annu. Rev. Physiol.* **61**, 435–456 (1999).
4. Zemel, R., Dayan, P. & Pouget, A. Probabilistic interpretation of population codes. *Neural Comput.* **10**, 403–430 (1998).
5. Tolhurst, D., Movshon, J. & Dean, A. The statistical reliability of signals in single neurons in cat and monkey visual cortex. *Vision Res.* **23**, 775–785 (1982).
6. Foldiak, P. in *Computation and Neural Systems* (eds Eeckman, F. & Bower, J.) 55–60 (Kluwer Academic Publishers, Norwell, Massachusetts, 1993).
7. Salinas, E. & Abbot, L. Vector reconstruction from firing rate. *J. Comput. Neurosci.* **1**, 89–108 (1994).
8. Sanger, T. Probability density estimation for the interpretation of neural population codes. *J. Neurophysiol.* **76**, 2790–2793 (1996).
9. Zhang, K., Ginzburg, I., McNaughton, B. & Sejnowski, T. Interpreting neuronal population activity by reconstruction: unified framework with application to hippocampal place cells. *J. Neurophysiol.* **79**, 1017–1044 (1998).
10. Cox, D. & Hinkley, D. *Theoretical Statistics* (Chapman and Hall, London, 1974).
11. Ferguson, T. *Mathematical statistics: a decision theoretic approach* (Academic, New York, 1967).
12. Paradiso, M. A theory of the use of visual orientation information which exploits the columnar structure of striate cortex. *Biol. Cybernet.* **58**, 35–49 (1988).
13. **A pioneering study of the statistical properties of population codes, including the first use of Bayesian techniques to read out and analyse population codes.**
14. Seung, H. & Sompolinsky, H. Simple model for reading neuronal population codes. *Proc. Natl Acad. Sci. USA* **90**, 10749–10753 (1993).
15. Deneve, S., Latham, P. & Pouget, A. Reading population codes: A neural implementation of ideal observers. *Nature Neurosci.* **2**, 740–745 (1999).
16. **Shows how a recurrent network of units with bell-shaped tuning curves can be wired to implement a close approximation to a maximum likelihood estimator. Maximum likelihood estimation is widely used in psychophysics to analyse human performance in simple perceptual tasks in a class of model known as ‘ideal observer analysis’.**
17. Georgopoulos, A., Kalaska, J. & Caminiti, R. On the relations between the direction of two-dimensional arm movements and cell discharge in primate motor cortex. *J. Neurosci.* **2**, 1527–1537 (1982).
18. Pouget, A., Deneve, S., Ducom, J. & Latham, P. Narrow vs wide tuning curves: what’s best for a population code? *Neural Comput.* **11**, 85–90 (1999).
19. Pouget, A. & Thorpe, S. Connectionist model of orientation identification. *Connect. Sci.* **3**, 127–142 (1991).
20. Regan, D. & Beverley, K. Postadaptation orientation discrimination. *J. Opt. Soc. Am.* **4**, 147–155 (1985).
21. Zhang, K. Representation of spatial orientation by the intrinsic dynamics of the head-direction cell ensemble: a theory. *J. Neurosci.* **16**, 2112–2126 (1996).
22. **Head direction cells in rats encode the heading direction of the rat in world-centred coordinates. This internal compass is calibrated with sensory cues, maintained in the absence of these cues and updated after each movement of the head. This model shows how attractor networks of neurons with bell-shaped tuning curves to head direction can be wired to account for these properties.**
23. Seung, H. How the brain keeps the eyes still. *Proc. Natl Acad. Sci. USA* **93**, 13339–13344 (1996).
24. Poggio, T. A theory of how the brain might work. *Cold Spring Harbor Symp. Quant. Biol.* **55**, 899–910 (1990).
25. **Introduces the idea that many computations in the brain can be formalized in terms of nonlinear mappings, and as such can be solved with population codes computing basis functions. Although completely living up to the title would be a tall order, this remains a most interesting proposal. A wide range of available neurophysiological data can be easily understood within this framework.**
26. Pouget, A. & Sejnowski, T. Spatial transformations in the parietal cortex using basis functions. *J. Cogn. Neurosci.* **9**, 222–237 (1997).
27. Andersen, R., Essick, G. & Siegel, R. Encoding of spatial location by posterior parietal neurons. *Science* **230**, 456–458 (1985).
28. Squatrito, S. & Maioli, M. Gaze field properties of eye position neurons in areas MST and 7a of macaque monkey. *Visual Neurosci.* **13**, 385–398 (1996).
29. Rumelhart, D., Hinton, G. & Williams, R. in *Parallel Distributed Processing* (eds Rumelhart, D., McClelland, J. & Group, P. R.) 318–362 (MIT Press, Cambridge, Massachusetts, 1986).
30. Zipser, D. & Andersen, R. A back-propagation programmed network that stimulates response properties of a subset of posterior parietal neurons. *Nature* **331**, 679–684 (1988).
31. Burnod, Y. *et al.* Visuomotor transformations underlying arm movements toward visual targets: a neural network model of cerebral cortical operations. *J. Neurosci.* **12**, 1435–1453 (1992).
32. **A model of the coordinate transformation required for arm movements using a representation very similar to basis functions. This model was one of the first to relate the tuning properties of cells in the primary motor cortex to their computational role. In particular, it explains why cells in M1 change their preferred direction to hand movement with starting hand position.**
33. Salinas, E. & Abbot, L. Transfer of coded information from sensory to motor networks. *J. Neurosci.* **15**, 6461–6474 (1995).
34. **A model showing how a basis function representation can be used to learn visuomotor transformations with a simple hebbian learning rule.**
35. Olshausen, B. & Field, D. Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature* **381**, 607–609 (1996).
36. Bishop, C., Svenson, M. & Williams, C. GTM: The generative topographic mapping. *Neural Comput.* **10**, 215–234 (1998).
37. Lewicki, M. & Sejnowski, T. Learning overcomplete representations. *Neural Comput.* **12**, 337–365 (2000).
38. Hinton, G. E. in *Proceedings of the Ninth International Conference on Artificial Neural Networks* 1–6 (IEEE, London, England, 1999).
39. Poggio, T. & Edelman, S. A network that learns to recognize three-dimensional objects. *Nature* **343**, 263–266 (1990).
40. Salinas, E. & Abbott, L. Invariant visual responses from attentional gain fields. *J. Neurophysiol.* **77**, 3267–3272 (1997).
41. Deneve, S. & Pouget, A. in *Advances in Neural Information Processing Systems* (eds Jordan, M., Kearns, M. & Solla, S.) (MIT Press, Cambridge, Massachusetts, 1998).
42. Groh, J. & Sparks, D. Two models for transforming auditory signals from head-centered to eye-centered coordinates. *Biol. Cybernet.* **67**, 291–302 (1992).
43. Pouget, A. & Sejnowski, T. A neural model of the cortical representation of egocentric distance. *Cereb. Cortex* **4**, 314–329 (1994).
44. Olshausen, B., Anderson, C. & Essen, D. V. A multiscale dynamic routing circuit for forming size- and position-invariant object representations. *J. Comput. Neurosci.* **2**, 45–62 (1995).
45. Treue, S., Hol, K. & Rauber, H. Seeing multiple directions of motion-physiology and psychophysics. *Nature Neurosci.* **3**, 270–276 (2000).
46. Shadlen, M., Britten, K., Newsome, W. & Movshon, T. A computational analysis of the relationship between neuronal and behavioral responses to visual motion. *J. Neurosci.* **16**, 1486–1510 (1996).
47. Zemel, R. & Dayan, P. in *Advances in Neural Information Processing Systems 11* (eds Kearns, M., Solla, S. & Cohn, D.) 174–180 (MIT Press, Cambridge, Massachusetts, 1999).
48. Simoncelli, E., Adelson, E. & Heeger, D. in *Proceedings 1991 IEEE Computer Society Conference on Computer Vision and Pattern Recognition* 310–315 (Los Alamitos, Los Angeles, 1991).
49. Watamaniuk, S., Sekuler, R. & Williams, D. Direction perception in complex dynamic displays: the integration of direction information. *Vision Res.* **29**, 47–59 (1989).
50. Anderson, C. in *Computational Intelligence Imitating Life* 213–222 (IEEE Press, New York, 1994).
51. **Neurons are frequently suggested to encode single values (or at most 2–3 values for cases such as transparency). Anderson challenges this idea and argues that population codes might encode probability distributions instead. Being able to encode a probability distribution is important because it would allow the brain to perform Bayesian inference, an efficient way to compute in the face of uncertainty.**
52. Recanzone, G., Wurtz, R. & Schwarz, U. Responses of MT and MST neurons to one and two moving objects in the receptive field. *J. Neurophysiol.* **78**, 2904–2915 (1997).
53. Wezel, R. V., Lankheet, M., Verstraten, F., Maree, A. & Grind, W. V. D. Responses of complex cells in area 17 of the cat to bi-vectorial transparent motion. *Vision Res.* **36**, 2805–2813 (1996).
54. Riesenhuber, M. & Poggio, T. Hierarchical models of object recognition in cortex. *Nature Neurosci.* **2**, 1019–1025 (1999).
55. Perrett, D., Mistlin, A. & Chitty, A. Visual neurons responsive to faces. *Trends Neurosci.* **10**, 358–364 (1987).
56. Bruce, V., Cowey, A., Ellis, A. & Perret, D. *Processing the Facial Image* (Clarendon, Oxford, 1992).