



Efficient computation and cue integration with noisy population codes

S. Deneve¹, P. E. Latham² and A. Pouget¹

¹ Department of Brain and Cognitive Sciences, University of Rochester, Rochester, New York 14627, USA

² Department of Neurobiology, UCLA, Los Angeles, California 90095, USA

Correspondence should be addressed to A.P. (alex@bcs.rochester.edu)

The brain represents sensory and motor variables through the activity of large populations of neurons. It is not understood how the nervous system computes with these population codes, given that individual neurons are noisy and thus unreliable. We focus here on two general types of computation, function approximation and cue integration, as these are powerful enough to handle a range of tasks, including sensorimotor transformations, feature extraction in sensory systems and multisensory integration. We demonstrate that a particular class of neural networks, basis function networks with multidimensional attractors, can perform both types of computation optimally with noisy neurons. Moreover, neurons in the intermediate layers of our model show response properties similar to those observed in several multimodal cortical areas. Thus, basis function networks with multidimensional attractors may be used by the brain to compute efficiently with population codes.

Many computations performed by the brain involve nonlinear transformations, cue integration or both. For instance, reaching for a target requires a transformation from the eye-centered coordinates of the target into joint coordinates, the set of joint angles of the arm that would bring the hand to the target. This transformation, like most sensory–motor transformations, is nonlinear¹. Likewise, recognizing an object requires a nonlinear transformation from its retinal image to its identity². Often, multiple cues are available to perform these transformations. When reaching for an object that can be seen and heard, for example, the brain combines visual and auditory information to refine the reaching motor command. Similarly, when estimating the three-dimensional structure of an object, the visual cortex integrates several cues such as stereopsis, structure from motion, and perspective.

In the cortex, these computations require manipulating variables encoded in the activity (the firing rate, or number of spikes per second) of populations of neurons. Typically, population codes involve neurons with bell-shaped tuning curves³. An example of tuning curves for the eye-centered position of an object, denoted x_p , is illustrated in Fig. 1a. These curves are idealizations; from trial to trial, real neurons vary in their response to an identical stimulus. As a result of this noise, the pattern of activity corresponding to a particular value of x_p , for example, $x_p = -20^\circ$, forms a noisy hill (Fig. 1b).

Understanding how cortical circuits perform nonlinear transformations and cue integration is difficult because of the distributed nature of population codes and the presence of neuronal noise. In the case of cue integration, additional complications arise because the reliability of a cue can change rapidly. For instance, a high-contrast image of an object can unexpectedly become low in contrast, making it a less reliable cue. Therefore, the challenge is to understand whether the cortex can perform

optimal cue integration even when the reliability of cues changes from one trial to the next.

Previous models using population codes have typically focused on limited cases, such as noise clean-up^{4,5} or nonlinear transformations with noiseless units². Here, we suggest a neural solution to the general case. More specifically, we demonstrate that a particular type of neural network, a basis function network with a multi-dimensional attractor, can provide a generic architecture for efficiently implementing nonlinear transformations and cue integration with noisy neurons. By “efficiently,” we mean that the network can perform these computations as reliably as possible given the neuronal noise. The type of network we consider here performs efficiently if two conditions are met: neurons exhibit lateral and feedback connections, and they compute basis functions of their inputs. Both conditions exist in the cortex, raising the possibility that cortical circuits approximate basis function networks with multi-dimensional attractors.

Here we first show how function approximation and cue integration can be formalized within the framework of maximum likelihood estimation. Next, we describe a neural architecture implementing a maximum likelihood estimator. The network is obtained by turning a basis function network into a multidimensional attractor network. The former is a well-known architecture for performing nonlinear mappings^{2,6}; the latter is an architecture for performing optimal noise clean-up⁵. We found that the network is indeed a close approximation to a maximum likelihood estimator for function approximation and cue integration. In particular, the network remains optimal even when the reliability of cues changes from trial to trial. Finally, we found that the intermediate units of the network (the basis function units) exhibit partially shifting receptive fields, a type of response observed in several parietal and premotor areas.



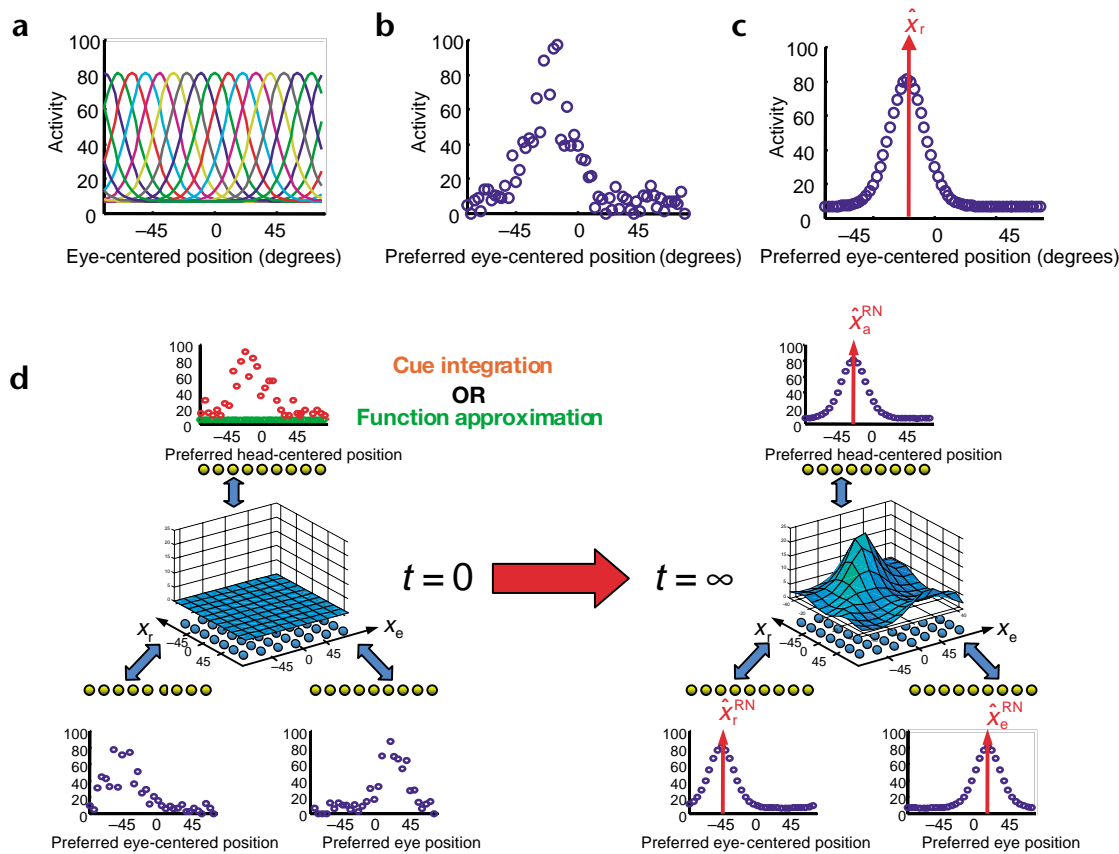


Fig. 1. A network that performs function approximation and cue integration optimally in the presence of noise. **(a)** Population of neurons with bell-shaped tuning curves for the eye-centered position of visual stimuli. **(b)** Noisy pattern of activity across the neuronal population shown in **(a)** in response to an object located at -20° . The activity of each neuron is plotted at the location of its preferred eye-centered position. **(c)** Same as **(b)**, but without neuronal noise. The peak of the smooth hill can be thought as a code for an estimate of eye-centered position, \hat{x}_r . **(d)** Network architecture. Left, initial conditions of the network ($t = 0$); right, the network after it evolved to its stable state ($t = \infty$). The two input layers at the bottom encode the eye-centered position of an object and the position of the eyes. Similarly, the top layer contains a population code for the head-centered position of objects. Two initial conditions are shown in the upper left. Green, function approximation; red, cue integration. The intermediate layer forms a two-dimensional map of basis function units (blue units) and is connected reciprocally to all three input layers (**Fig. 2**). In the simulations, each of the input layers contained 40 units, and the intermediate layer contained 20×20 units. The activity of each unit was obtained at each iteration by computing first the weighted sum of its input followed by a divisive normalization within each layer. (See supplementary information for details.) Right, result of a numerical simulation of cue integration after three iterations (subsequent iterations do not introduce significant changes in this pattern). Middle right, activity in the intermediate (basis function) layer; surrounding plots show the activity in the input layers. Whether the network is initialized with two noisy hills (function approximation) or three noisy hills (cue integration), the activity in the three input layers relaxes onto three smooth hills of which the positions are used as estimates of eye-centered, eye and head-centered positions (\hat{x}_r^{RN} , \hat{x}_e^{RN} , \hat{x}_a^{RN}). These estimates are very close to the maximum likelihood estimates.

RESULTS

Maximum likelihood computation

We consider computations involving variables coding for the eye-centered position of an object, x_r , the head-centered position of the same object, x_a , and the eye position, x_e . These three variables are related by $x_a = x_r + x_e$ (ref. 7). Although this example is linear, and we are interested in general transformations, especially nonlinear ones, it is nonlinear when implemented with population codes^{2,6}. Indeed, implementing this transformation is conceptually no different than implementing a nonlinear one, $x_a = f(x_r, x_e)$; the network could be modified to implement a nonlinear transformation simply by changing the network connectivity. (See supplementary information, available on the *Nature Neuroscience* web site, http://neurosci.nature.com/web_specials.) Thus, our results apply to nonlinear as well as linear transformations. Further moti-

vation for focusing on $x_a = x_r + x_e$ is that the response properties of the neurons involved have been extensively studied^{8,9}, which allows us to compare our results to experiments.

There are at least two types of computations that could be done with noisy population codes. In the first, called function approximation², the brain is given noisy hills for the variables x_r and x_e , and it must recover a third variable, x_a . For example, this is the computation required to predict the location of a sound source given its visual location. In the second, called cue integration¹⁰, a noisy population code for x_a is given in addition to noisy population codes for x_r and x_e , and the brain must refine its estimates of all three variables by combining information from the three noisy hills. This situation occurs, for instance, when localizing an object through the integration of the image and sound it generates.

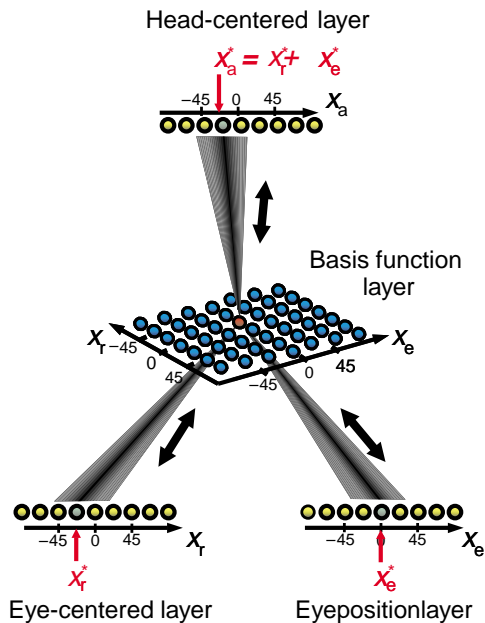


Fig. 2. Network connections. A representative set of weights from and to one particular basis function unit (red). All other weights are translated copies of these weights. The red basis function unit receives three sets of weights, one from each of the input layers. These connections are bi-directional, that is, if the basis function receives a connection from an input unit, it sends back a connection with the same weight. These connections are set so that the basis function unit is characterized by a preferred eye-centered position, a preferred eye position and a preferred head-centered position, denoted x_r^* , x_e^* and x_a^* . These positions correspond to the positions of the input units sending and receiving the largest weights in each of the input layers (green). Surrounding units also send connections to the red basis function unit with weights following a Gaussian profile centered on the preferred units (gray shading in the connections). If the strongest connections come from position x_r^* in the eye-centered layer, x_e^* in the eye position layer and x_a^* in the head-centered layer, then x_r^* , x_e^* and x_a^* obey the relationship $x_a^* = x_r^* + x_e^*$. For example, if the red basis function unit gets its strongest connections from $x_r^* = -20^\circ$ in the eye-centered layer and $x_e^* = 0^\circ$ in the eye position layer, then it must get its strongest connection from $x_a^* = -20^\circ$ in the head-centered layer. This constraint ensures that the network computes the function $x_a = x_r + x_e$, and all permutations thereof such as $x_r = x_a - x_e$. In addition to these feedforward and feedback weights, each layer also contains lateral weights implementing a divisive normalization. (See supplementary information.)

Because neurons are noisy, we cannot perform these computations with absolute certainty; the best we can do, for both function approximation or cue integration, is to provide estimates of the encoded variables. In both cases, we are given noisy patterns of activity for eye-centered position, eye position and head-centered position, and we seek estimates of these quantities. The only difference between the two tasks lies in the profile of the patterns of activity. For function approximation, two of the patterns are noisy hills and the third is just a flat pattern of activity (Fig. 1d, left, green data points); for cue combination, all three patterns are noisy hills of activity.

Various statistical methods can be used to compute the estimates of x_r , x_e and x_a , denoted \hat{x}_r , \hat{x}_e and \hat{x}_a , from patterns of activity. For the tasks proposed here, however, the optimal method is maximum likelihood^{11–13}. We denote the patterns of activity across the populations of neurons as R_r , R_e and R_a ; all three are activity vectors. (For example, if the eye-centered neuronal population contains 64 neurons, then R_r is a 64-dimensional vector storing the activity of all the neurons.) The maximum likelihood estimates are the values of x_r , x_e and x_a that maximize the likelihood function $P(R_r, R_e, R_a | x_r, x_e, x_a)$, in which the likelihood function is the probability distribution of the neuronal activity for fixed inputs¹⁴. Denoting these values \hat{x}_r^{ML} , \hat{x}_e^{ML} and \hat{x}_a^{ML} , we have the following equation.

$$(\hat{x}_r^{ML}, \hat{x}_e^{ML}, \hat{x}_a^{ML}) = \arg \max_{x_r, x_e, x_a} P(R_r, R_e, R_a | x_r, x_e, x_a) \quad (1)$$

When there are a large number of neurons, the maximum likelihood estimator is optimal. First, it produces estimates that are correct on average; the value of \hat{x}_r^{ML} , averaged over many trials, is equal to x_r , the true value, and similarly for the other two variables, x_e and x_a . Second, it produces estimates that have the smallest possible variance of any estimator¹¹.

Network architecture

Can a biologically plausible neuronal network perform function approximation and cue integration optimally, that is, as well as

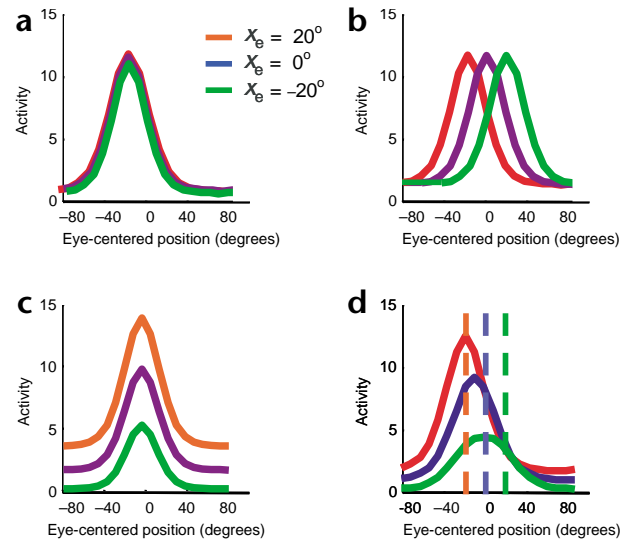
a maximum likelihood estimator? To be biologically plausible, the network should recover population codes for \hat{x}_r^{ML} , \hat{x}_e^{ML} and \hat{x}_a^{ML} , as opposed to scalar values for these variables. Indeed, neurophysiological recordings indicate that population codes are used everywhere in the cortex; that is, there is no stage at which population codes are decoded into scalar values. One way to recover population codes is to build a network that, when initialized with the noisy patterns of activity R_r , R_e and R_a , converges onto smooth hills peaking at the locations \hat{x}_r^{ML} , \hat{x}_e^{ML} and \hat{x}_a^{ML} (example, Fig. 1c). Such a network performs function approximation when two of the three input patterns (R_r and R_e) are noisy hills and the third, R_a , is simply a flat pattern of activity, and it performs cue integration when all three input patterns (R_r , R_e and R_a) are noisy hills (Fig. 1d).

Implementing function approximation with a network using population codes is a well-known problem, as long as the neurons are noiseless (for review, see ref. 1). Specifically, given noiseless patterns of activity for eye-centered position and eye positions peaking at x_r and x_e , respectively, it is known how to construct a neural network that produces as its output a noiseless hill peaking at $x_a = x_r + x_e$. Neurons coding for eye-centered position and eye position connect to an intermediate layer of units known as basis function units^{2,6}. This layer is two-dimensional, and the projections are chosen so that one-dimensional hills located at x_r and x_e produce a two-dimensional hill at position (x_r, x_e) (Fig. 1d). Suitable projections from this two-dimensional hill to an output layer can then generate a hill of activity peaking at location $x_a = x_r + x_e$. Indeed, the intermediate basis function units can be used to implement a wide range of nonlinear functions of x_r and x_e (ref. 2), not just the addition considered here. The appropriate weights can be found with standard linear regression techniques or a learning procedure such as the delta rule^{1,15}.

We show here that this basis function architecture is not limited to function approximation with noiseless units, but can be extended to deal optimally with function approximation and cue integration even when the neurons are noisy. The key is to add feedback connections from the head-centered layer to the intermediate layer, as if one were building a basis function network to extract eye-centered position from head-centered and eye posi-



Fig. 3. The eye-centered receptive field of the network units for three positions of the eyes (x_e). (a) For a typical unit in the eye-centered layer, the eye-centered receptive field remains at the same location across eye position. (b) For a typical head-centered unit, the eye-centered receptive field shifts by an amount exactly opposite to the change in eye position. (c) Same plot as in (a) and (b) but for an intermediate unit in a network with feedforward connections only (from eye-centered and eye position to head-centered position). The unit exhibits an eye-centered receptive field with a position that does not vary with eye position. The height of the tuning curve, however, is modulated by eye position. This is known as a gain field, a common response pattern throughout the cortex. (d) Same as (c) but in a network with feedforward and feedback connections. The position and height of the eye-centered receptive field changes with eye position. The shift is only half of what is predicted for a unit with a head-centered receptive field; it has a partially shifting receptive field (The three vertical dashed lines show the predicted peak positions for a head-centered receptive field as in b.) This response pattern has been reported in the parietal cortex, superior colliculus and premotor cortex. The amplitude of the shift can be increased or decreased depending on the ratio of feedforward connections from the eye-centered layer and feedback connections from the head-centered layer.



tions (that is, as if the network were doing the reverse computation $x_r = x_a - x_e$). Feedback connections from the eye position to the intermediate layer can similarly be added to compute $x_e = x_a - x_r$. The result is three sets of mirror image connections from the eye-centered location, eye position and head-centered location to the intermediate layer and back (Fig. 2).

The feedback connections make the network recurrent, which allows interactions among all three variables, eye, eye-centered and head-centered positions. Because the activity of recurrent networks evolves in time, stability is an issue. Thus, we use networks with weights set to ensure that activity patterns converge to stable states. These stable states consist of smooth hills of activity with locations that depend on the position of the peaks of the noisy input hills. Such networks are known as multi-dimensional attractor networks⁵. This means that when initialized with noisy hills (noisy population codes), the network stabilizes onto smooth hills of activity (noiseless population codes) in all layers (Fig. 1d), and stays in that state unless reset to zero. This behavior endows the network with the ability to act as an estimator; the peaks of the smooth hills can be used as estimates of the encoded variables whether the network is performing function approximation or cue integration (Fig. 1d).

Simulations

The network performs as a maximum likelihood estimator only if the smooth hills peak at locations close to the maximum likelihood estimates, \hat{x}_r^{ML} , \hat{x}_e^{ML} and \hat{x}_a^{ML} . To test whether this is indeed the case, we simulated a basis function network with recurrent connections, as described above (also see supplementary information). We performed 100,000 trials—each of which involved only 3 iterations—and compared the mean and standard deviation of the network estimates to those obtained from maximum likelihood. We used only three iterations because the smooth hills of activity had already reached their stable positions by the end of the third iteration. We found that in all simulations, the network estimate was unbiased; the mean values of estimates, \hat{x}_r , \hat{x}_e and \hat{x}_a , were equal to the true values, x_r , x_e and x_a . Thus, the quality of the network was measured by the standard deviation of the estimated quantities. Because maximum likelihood has the minimum standard deviation of all unbiased estimators, we report our results in terms of percentage above the standard deviation of the maximum likelihood estimator.

The results given below were obtained in networks in which the weights were adjusted to optimize performance. More specifically, the weights were adjusted to change the width of the stable smooth hills until the estimates, \hat{x}_r , \hat{x}_e and \hat{x}_a , had minimum standard deviation in the cue integration task. Once the weights were optimized, the same weights were used for all simulations.

For function approximation, the network was initialized with two noisy hills coding for x_r and x_e , whereas the neurons coding for x_a were initialized to zero. We used a Poisson distribution for the noise, a common approximation of actual neuronal noise¹⁴. The network performed 2.1% worse than maximum likelihood for eye-centered and eye positions and 3.3% worse than maximum likelihood for head-centered position.

In the case of cue integration, the network was initialized with three noisy hills coding for x_r , x_e , and x_a , chosen to satisfy $x_a = x_r + x_e$. To be optimal, the network must integrate in each layer the information from the local hills of activity with the information provided by the other two hills. Once again, the network performance was the same for all variables, and near optimal—it was only 1.9% worse than maximum likelihood. For comparison, we also computed the performance of a maximum likelihood estimator that used the activity of each hill individually, with no integration of information. This non-optimal strategy was 22% worse than maximum likelihood, much worse than the network performance, indicating that each hill is used to refine the estimate of the others.

In most situations, cues are not equally reliable. For instance, in broad daylight, vision is more reliable than audition for localizing objects, whereas at night, audition is the more reliable cue. Optimal estimators like maximum likelihood weight the contribution of the cues based on their reliability¹⁰; for example, vision should be assigned a greater weight in daylight than at night. We explored whether the network could adapt to such changes in reliability by performing simulations in which the height of the head-centered hill was varied as the heights of the other hills were kept constant. Because the noise followed a Poisson distribution, the height of the hill of activity (that is, the gain of the corresponding sensory input) controls the reliability of the cue¹⁶. For example, a decrease in height by a factor of 2 decreases the signal-to-noise ratio by a factor of $\sqrt{2}$. We varied the height of the hill coding for the head-centered position and found that the network performed close to optimal for all heights from 0 (indicat-



ing a flat pattern of activity, implying that the network does function approximation) to 2 (indicating that the head-centered hill was 2 times higher than the hills coding for retinal and eye position and thus $\sqrt{2}$ times more reliable). For this range of heights, the network estimate for head-centered position ranged from 5% to 3% worse than maximum likelihood. These results were obtained with weights optimized for a height of 1; no further optimization was used. Thus, the network has the ability to adapt to changes in reliability.

We also found that the performance of our network is robust to changes in the width of the input tuning curves. For instance, we tested our network for a range of widths extending from 45 to 75 degrees (full width at half maximum amplitude) while keeping the weights constant. Performance for both function approximation and cue integration was, at worst, 5.4% above maximum likelihood in all layers.

The simulations described so far used the noisy input hills to specify the initial state of the network, but do not maintain these inputs thereafter. This assumption is not critical, as revealed by the results of another set of simulations in which we clamped the activity of the noisy hills onto the input units. We used a network in which 20% of the activity of the input unit at each time step depends on the clamped noisy hills and the remaining 80% comes from the recurrent activity (a ratio consistent with cortical anatomy¹⁷). Performance stayed near optimal at 3.5% for function approximation and 4.5% for cue integration.

These numerical results indicate that basis function networks with multi-dimensional attractors can provide estimates close to those provided by maximum likelihood. These results are not restricted to the particular function we used in the simulations. We have found analytically that they are common to any non-linear function that can be approximated with a basis function representation. (Our proof is based on an extension of our previous work⁵, but the details of this proof are beyond the scope of the present paper.)

It remains to be seen whether the cortex uses this architecture. To address whether the units of our network model show responses consistent with those reported in the cortex, we plotted the response of a typical unit in the eye-centered layer (Fig. 3a). The eye-centered receptive field of the unit is identical across all three positions of the eyes. In contrast, the same plot for a typical unit in the head-centered layer reveals that its eye-centered receptive field shifts by the same amount as the eyes (20°), but in the opposite direction (Fig. 3b). This is as expected, because the eye-centered location of a stimulus can be obtained from its head-centered location by subtracting the position of the eyes.

For the intermediate (basis function) units, the results depend on whether or not the network contains feedback connections. We first considered a network without feedback connections, that is, a network with feedforward connections only from the eye-centered and eye position layers onto the head-centered layer via the intermediate layer. We found that a typical intermediate unit has a receptive field with a position that is invariant in eye-centered coordinates but with an amplitude that is modulated by the position of the eye (Fig. 3c). This type of response is sometimes known as a gain field, a very common response type observed in areas ranging from the primary cortices to the premotor cortex^{8,18–20}.

In contrast, in a network with feedforward and feedback connections (Figs. 1 and 2), the visual receptive fields of the intermediate units showed both a gain modulation with eye position and a shift in the peak position in eye-centered coordinates (Fig. 3d). Like the head-centered units shown in Fig. 3b, the shift

is in the direction opposite to the change in eye position, but unlike that unit, the amplitude of the shift is only half the amplitude of the change in eye position. The amplitude of the shift is controlled by the amplitude ratio of the feedforward connections from the eye-centered layer and the feedback connections from the head-centered layer. Thus, if the amplitude of the feedback connections is decreased, the shift in eye-centered coordinates decreases accordingly. As a result, the receptive fields of the intermediate units become increasingly more eye-centered (and vice versa if these weights were to be increased).

Suppose we interpret the eye-centered map as a visual map and the head-centered map as an auditory map. Then, in a network with feedforward and feedback connections, the intermediate units are visual-auditory, effectively bimodal (or even trimodal, if one considers the eye position input). We could therefore map the auditory receptive fields of these units. We found that the auditory receptive fields are exactly lined up with the visual receptive fields because, once the stable state of the network is reached, the activity coming from the visual layer is spatially consistent with the activity coming from the auditory layer. These activities are spatially consistent in the sense that the positions of three smooth hills in the three input layers follow the relationship $x_a = x_r + x_e$.

These results predict that some of the multimodal neurons in the cortex should show partially shifting receptive fields in all modalities with varying amount of shift depending on the strength of their sensory inputs. Partially shifting receptive fields have already been reported in the superior colliculus, VIP and premotor cortex^{9,21,22}, but their role has remained obscure. Our work suggests that they could be providing basis functions for reliable function approximation and cue integration in the presence of noise.

DISCUSSION

We have extended the findings of a previous study in which we used a line attractor network to perform optimal noise clean-up⁵. First, we showed that a network could compute in a near-optimal manner with noisy population codes, not just clean up noise. Second, we demonstrated that the new architecture deals equally well with cue integration, even when the reliability of the cue changes from trial to trial. These results indicate that this architecture can be applied to a wide variety of neural computations, including recovering depth from multiple visual cues, multisensory integration in general, and any sensorimotor transformation.

Our work also offers a new perspective on multimodal representations. Most studies of multisensory integration assume that sensory modalities are first remapped into a common frame of reference before being combined. For instance, it has been suggested that auditory inputs are remapped in eye-centered coordinates in the superior colliculus where they are combined with visual inputs²¹. We suggest that multisensory integration might also involve basis function representations in which multiple frames of reference are multiplexed within the same representation. The implications of this representational scheme for our understanding of phenomena such as multimodal hemineglect or motor control toward multisensory targets remain to be explored.

ACKNOWLEDGEMENTS

A.P. and S.D. are supported by grants from NIH (MH57823-05), ONR (N00014-00-1-0642) and a fellowship from the McDonnell-Pew Foundation. P.E.L. is supported by a grant from NIH (MH62447-01). We thank J.-R. Duhamel for discussions.



ACCEPTED 26 JUNE 2001

1. Pouget, A. & Snyder, L. Computational approaches to sensorimotor transformations. *Nat. Neurosci.* 3, 1192–1198 (2000).
2. Poggio, T. A theory of how the brain might work. *Cold Spring Harb. Symp. Quant. Biol.* 55, 899–910 (1990).
3. Hubel, D. & Weisel, T. Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. *J. Physiol. (Lond.)* 160, 106–154 (1962).
4. Douglas, R. J., Koch, C., Mahowald, M., Martin, K. A. & Suarez, H. H. Recurrent excitation in neocortical circuits. *Science* 269, 981–985 (1995).
5. Deneve, S., Latham, P. & Pouget, A. Reading population codes: a neural implementation of ideal observers. *Nat. Neurosci.* 2, 740–745 (1999).
6. Pouget, A. & Sejnowski, T. Spatial transformations in the parietal cortex using basis functions. *J. Cogn. Neurosci.* 9, 222–237 (1997).
7. Westheimer, G. & Tanzman, I. Qualitative depth localization with diplopic images. *J. Opt. Soc. Am.* 46, 116–117 (1956).
8. Andersen, R., Essick, G. & Siegel, R. Encoding of spatial location by posterior parietal neurons. *Science* 230, 456–458 (1985).
9. Duhamel, J., Bremmer, F., BenHamed, S. & Graf, W. Spatial invariance of visual receptive fields in parietal cortex neurons. *Nature* 389, 845–848 (1997).
10. Yuille, A. L. & Bulthoff, H. H. in *Perception as Bayesian Inference* (eds. Knill, D. C. & Richards, W.) 123–162 (University Press, New York, Cambridge, 1996).
11. Papoulis, A. *Probability, Random Variables, and Stochastic Process* (McGraw-Hill, New York, 1991).
12. Paradiso, M. A theory of the use of visual orientation information which exploits the columnar structure of striate cortex. *Biol. Cybern.* 58, 35–49 (1988).
13. Seung, H. & Sompolinsky, H. Simple model for reading neuronal population codes. *Proc. Natl. Acad. Sci. USA* 90, 10749–10753 (1993).
14. 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).
15. Salinas, E. & Abbot, L. Transfer of coded information from sensory to motor networks. *J. Neurosci.* 15, 6461–6474 (1995).
16. McAdams, C. J. & Maunsell, J. R. H. Effects of attention on orientation-tuning functions of single neurons in macaque cortical area V4. *J. Neurosci.* 19, 431–441 (1999).
17. Braitenberg, V. & Schuz, A. *Anatomy of the Cortex* (Springer, Berlin, 1991).
18. Boussaoud, D., Barth, T. & Wise, S. Effects of gaze on apparent visual responses of frontal cortex neurons. *Exp. Brain Res.* 93, 423–434 (1993).
19. Bremmer, F., Ilg, U., Thiele, A., Distler, C. & Hoffman, K. Eye position effects in monkey cortex. I: Visual and pursuit-related activity in extrastriate areas MT and MST. *J. Neurophysiol.* 77, 944–961 (1997).
20. Trotter, Y. & Celebrini, S. Gaze direction controls response gain in primary visual-cortex neurons. *Nature* 398, 239–242 (1999).
21. Jay, M. F. & Sparks, D. L. Sensorimotor integration in the primate superior colliculus: I. Motor convergence. *J. Neurophysiol.* 57, 22–34 (1987).
22. Graziano, M., Yap, G. & Gross, C. Coding of visual space by premotor neurons. *Science* 266, 1054–1057 (1994).