LETTERS

Efficient auditory coding

Evan C. Smith[1,2] & Michael S. Lewicki[2,3]

The auditory neural code must serve a wide range of auditory tasks that require great sensitivity in time and frequency and be effective over the diverse array of sounds present in natural acoustic environments. It has been suggested[1–5] that sensory systems might have evolved highly efficient coding strategies to maximize the information conveyed to the brain while minimizing the required energy and neural resources. Here we show that, for natural sounds, the complete acoustic waveform can be represented efficiently with a nonlinear model based on a population spike code. In this model, idealized spikes encode the precise temporal positions and magnitudes of underlying acoustic features. We find that when the features are optimized for coding either natural sounds or speech, they show striking similarities to time-domain cochlear filter estimates, have a frequency-bandwidth dependence similar to that of auditory nerve fibres, and yield significantly greater coding efficiency than conventional signal representations. These results indicate that the auditory code might approach an information theoretic optimum and that the acoustic structure of speech might be adapted to the coding capacity of the mammalian auditory system.

A fundamental issue in auditory coding is the nature of the computations that transform the raw sensory signal into a representation that is useful for auditory tasks. The response properties of cochlear nerves have been studied extensively, and models based on these results capture many properties of the neural response[6,7]. However, these properties and auditory coding are still poorly understood in terms of underlying theoretical principles. Of the many sensory codes that could have existed, why has nature chosen one in particular, either through adaptation or evolution?

Traditional views describe auditory coding in terms of spectral features, such as frequency, intensity and phase, that are estimated from the signal. This perspective focuses on the properties and response of the system rather than its purpose. In contrast, theoretical approaches seek to predict properties of the system from underlying principles. What are these principles? One hypothesis is that of efficient coding, which posits that one of the primary goals of sensory coding is to form an efficient code, namely one that maximizes the amount of information conveyed about the sensory signal to the rest of the brain[1–5]. In the peripheral auditory system, the incoming acoustic signal is transmitted mechanically to the inner ear and undergoes a highly complex transformation before it is encoded by spikes at the auditory nerve. If the auditory code is efficient, it should be possible to predict its properties from a theoretically ideal code.

To test this hypothesis, we must start with a mathematical description of an acoustic waveform that can then be used to derive theoretically optimal codes. Here we use a model in which sounds are encoded as a pattern of spikes[8–10]. The signal, $x(t)$, is encoded with a set of kernel functions, ϕ_1, …, ϕ_m, that can be positioned arbitrarily and independently in time. The mathematical form of the representation with additive noise is

$$x(t) = \sum\sum s_i^m \phi_m\left(t - \tau_i^m\right) + \varepsilon(t) \qquad (1)$$

where τ_i^m and s_i^m are the temporal position and coefficient of the ith instance of kernel ϕ_m, respectively. Note that the number of instances of ϕ_m need not be the same across kernels. To allow the kernel functions to assume arbitrary potential shapes, we represented each kernel ϕ_m by a vector of length L_m, where each element is an independent parameter of the model. Both the kernel shapes and their lengths were adapted to optimize coding efficiency; in the results below, the kernels take on a variety of shapes and range in length from ten to several hundred milliseconds. This provides a mathematical description of sound waveforms that has sufficient flexibility to encode arbitrary acoustic signals and encompass a broad range of potential auditory codes.

The key theoretical abstraction of the model is that the acoustic signal can be encoded most efficiently by decomposing it in terms of discrete acoustic elements, each of which has a precise amplitude and temporal position. This also yields a code that is time-relative and does not depend on artificial blocking of the signal[10]. One interpretation of each analogue τ_i^m, s_i^m pair is that it represents a local population of (binary) auditory nerve spikes firing probabilistically in proportion to the underlying analogue value. The form of the model allows for the case in which the coefficients s_i^m are constrained to be binary, but for computational reasons we have used analogue spikes as an approximation. To optimize the theoretical model to code natural sounds efficiently, we first need to address two problems: first, encoding (determining the optimal values of τ_i^m and s_i^m) and second, learning (determining the optimal kernel functions ϕ_m). From equation (1), coding efficiency can be defined approximately as the number of spikes required to achieve a desired level of precision, which is defined by the variance of the additive noise $\varepsilon(t)$. This assumes that the goal of coding is to represent the entire acoustic signal and that coding efficiency is most closely related to the number of spikes in the code. Other definitions are possible within this framework, but this definition has the advantage of starting from a minimal set of assumptions.

It is important to distinguish between the code and the encoding algorithm. For a given code (for example equation (1)), there are many different encoding algorithms that make different trade-offs in terms of coding efficiency, representational precision and computational complexity. Although the generative form of the model is linear, in other words the signal is a linear function of the representation, inferring the optimal representation for a signal is highly nonlinear and computationally complex. In fact, the problem of finding the optimal sparse representation by using a generic dictionary of functions is NP-hard[11], so only approximate algorithms are feasible. Here we compute the values of τ_i^m and s_i^m for a given signal by using a matching pursuit algorithm[12], which iteratively approximates the input signal and has been shown to yield highly efficient representations for a broad range of sounds[10]. Note also that although we have assumed a representation that consists of spikes (that is, a localized representation of the time position of an underlying acoustic feature[13]), spikes themselves are a consequence

[1]Department of Psychology, [2]Center for the Neural Basis of Cognition and [3]Department of Computer Science, Carnegie Mellon University, 4400 Fifth Avenue, Pittsburgh, Pennsylvania 15213, USA.

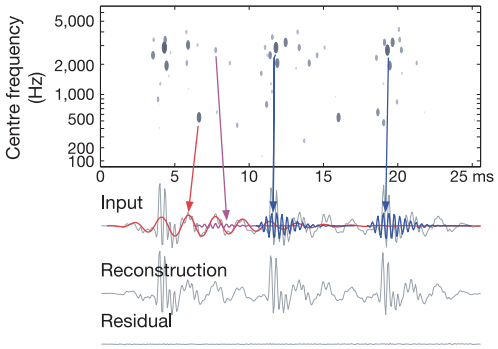


Figure 1 | Representing a natural sound with the use of spikes. A brief segment of the word 'canteen' (input) is represented as a spike code (top). Each spike (oval) represents the temporal position and centre frequency of an underlying kernel function, with oval size and grey value indicating kernel amplitude. The coloured arrows illustrate the correspondence between the spikes and the underlying acoustic structure represented by the kernel functions. Alignment of the spikes with respect to the kernels is arbitrary and is an issue only for plotting. We choose the kernel centre of mass, which for a delta-function input yields aligned spikes across the kernel population. A reconstruction of the speech from only the 60 spikes shown is accurate with little residual error (reconstruction and residual).

of optimizing the efficiency of the representation: if the coefficients s_i^m are assumed to be continuous in time and then optimized to represent the signal efficiently, only a discrete set of temporally sparse coefficients emerges[8–10,14].

Figure 1 illustrates the spike code model and its efficiency in representing speech. The spoken word 'canteen' was encoded with a set of spikes with the use of a fixed set of kernel functions (because the kernels can have arbitrary shape, for illustration purposes here we have chosen gammatones, mathematical approximations of cochlear filters). A brief segment from the input speech signal (Fig. 1, input) consists of three glottal pulses in the /a/ vowel. The resulting spike code is shown above it. The coloured arrows and curves indicate the relationship between the spikes (grey ovals) and the acoustic components they represent. The figure shows that a small set of spikes (for comparison, the sound segment contains about 400 samples) is sufficient to produce a very accurate reconstruction of the sound (Fig. 1, reconstruction and residual).

The spike-coding algorithm provides a way to encode signals given a set of kernel functions, but the actual efficiency of this code depends on how well the kernel functions capture the acoustic structure of the sound ensemble. To optimize the kernel functions we derived a gradient-based algorithm for adapting each kernel in shape and length to improve the fidelity of the representation (Supplementary Methods). Information theory states that there is a fundamental relationship between the efficiency of a code and the degree to which it captures the statistical structure of the signals being encoded. Thus, one of the primary tenets of efficient coding theory is that sensory codes should be adapted to the statistics of the relevant sensory environment. To make predictions, it is necessary to optimize the code to an ensemble of sounds to which the auditory system is thought to be adapted. However, this poses a problem because the precise composition of the natural acoustic environment is unknown, and many common sounds, such wind noise, may have much less behavioural relevance than other sounds.

To address this issue, we made the generic assumption that the auditory system is adapted to an unknown mixture of three broad categories of natural sounds. The kernel functions were optimized to an ensemble of natural sounds that consisted of mammalian vocalizations[15] and two subclasses of environmental sounds (Supplementary Methods). These sound classes represent a wide range of acoustic structure. Vocalizations tend to be harmonic and more steady-state, whereas environmental sounds have little or no harmonic structure and are more transient. Furthermore, to obtain an ensemble composition that yielded a good match to the physiological data (described below), we found it necessary to divide environmental sounds into two subclasses, namely transient environmental sounds, such as cracking twigs and crunching leaves, and ambient environmental sounds, such as rain and rustling sounds. This approach has the added advantage that we can investigate how the theoretically ideal code changes as a function of the sound ensemble composition.

Figure 2a shows the learned kernel functions (red curves) for the natural sounds ensemble. All kernels are time-localized, have a narrow spectral bandwidth and show a strong temporal asymmetry not predicted by previous theoretical models. The sharp attack and

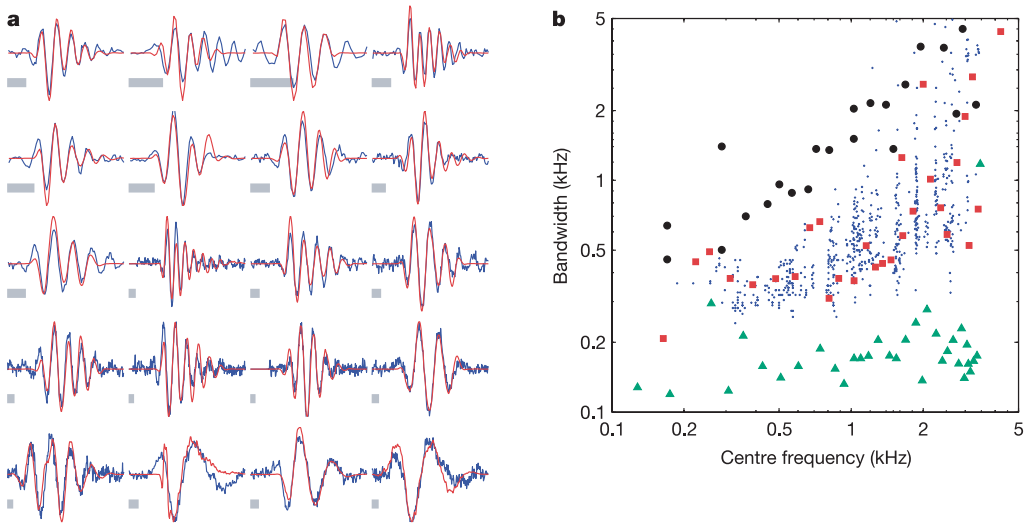


Figure 2 | Efficient codes for natural sounds predict revcor filter shapes and population characteristics. **a**, When optimized to encode an ensemble of natural sounds, kernel functions become asymmetric sinusoids (smooth curves in red, with padding removed) with sharp attacks and gradual decays. They also adapt in temporal extent, with longer and shorter functions emerging from the same initial length (grey scale bars, 5 ms). Each kernel function is overlaid on a revcor function obtained from cat auditory nerve fibres (noisy curves in blue). **b**, The bandwidth–centre-frequency distribution of learned kernel functions (red squares) is plotted together with cat physiological data (small blue dots) and with kernel functions trained on environmental sounds alone (black circles) or animal vocalizations alone (green triangles).

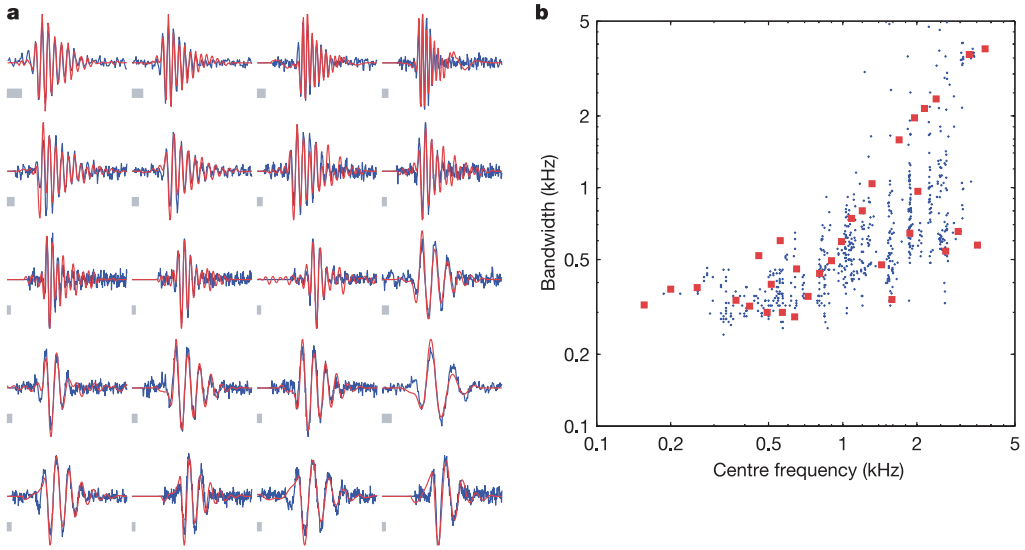


Figure 3 | Human speech is adapted to the mammalian cochlear code. a, As with the kernel functions trained on the natural sounds ensemble, the efficient code for speech consists of asymmetric sinusoids that closely match auditory revcor filters. **b**, The population of speech-trained kernels also matches the population centre bandwidth– frequency relationship of cochlear revcor filters. Details are as in Fig. 2.

gradual decay of the envelope match the physiological filtering properties of auditory nerves as characterized by reverse-correlation, which estimates the impulse response function of individual auditory nerve fibres[16–18], the so-called 'revcor' filters. Kernel functions in Fig. 2a are overlaid on revcor filters obtained from cat auditory nerves[19]. The revcor functions were normalized, and the most similar (as determined by correlation) were aligned in phase with each of the learned kernels. Even though the learned kernel functions were derived entirely independently of the physiological revcor functions, they are strikingly similar. The similarity of the experimental data and theoretical predictions is not a result of selection bias. Statistical comparisons of the different models (both theoretical and parametric) show that the kernels derived theoretically have a slightly higher median correlation with the revcor filters than do parametric gammachirp models[19] fitted to the same set of data (Supplementary Fig. S2). Repeated training from different random initial conditions produced similar results. We can also compare population properties of the learned kernels with those of the physiological data. Figure 2b shows a log–log scatter-plot of the bandwidth against centre frequency for each kernel (red squares) and for the physiological revcor data (small blue dots)[20]. Both the slope and spread of the learned kernel functions match those of the empirical data, and the distribution seems to follow shifts in the slope and spread at low and high frequencies.

In contrast with the efficient code for the natural sounds ensemble, kernel functions optimized for other sound classes predict neither the structure of revcor filters nor their population characteristics (Fig. 2b). Efficient codes for animal vocalizations (green triangles) have kernels with much narrower bandwidths that do not increase with frequency. Kernels optimized for only environmental sounds (black circles) have bandwidths that increase much more rapidly with frequency. Furthermore, their respective kernel functions are significantly different from mammalian revcor filters (Supplementary Fig. S1). However, kernel functions optimized for speech predict both the asymmetric revcor structure and the population distribution just as well as the kernel functions optimized for the natural sounds ensemble (Fig. 3, and Supplementary Fig. S2).

We can quantify the coding efficiency of the learned kernel functions to evaluate the model objectively and compare it quantitatively with other signal representations. Fidelity–rate curves, which plot the fidelity of the encoded signal (the signal-to-noise ratio in decibels) against the coding rate in bits per second, provide a useful, objective measure for comparison. Here we use a previously developed method[10] in which we vary the precision of the spike amplitude and temporal position values, s_i^m and τ_i^m, and compute the resulting fidelity from the reconstruction error. This produces a curve showing the trade-off between quality and cost for a given representation. Fidelity–rate curves for speech coding were calculated for Fourier transform, for discrete wavelet transform and for spike codes by using either gammatones or kernel functions optimized for speech (Supplementary Methods). For all fidelity–rate curves (Fig. 4), increasing the information rate (x axis) always improves fidelity (y axis). Below 30 dB, the spike codes using both the optimized and fixed kernels (gammatones) resulted in more efficient representations of speech than the traditional representations. At a fidelity of 15 dB, the learned spike code is over threefold more efficient than either the Fourier or wavelet codes (8 kilobits s^{-1} versus 30 kilobits s^{-1}). Spike codes using learned kernels are also more

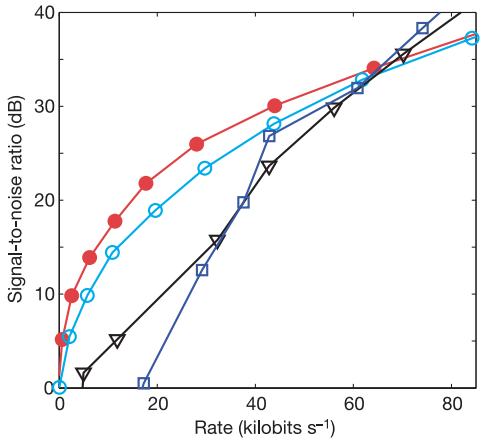


Figure 4 | Fidelity–rate curves for Fourier, wavelet and spike codes. The curves show the trade-off between coding cost and signal fidelity in the representation of speech signals from the TIMIT Speech Corpus testing set. Curves were generated for spike coding by using both speech-trained kernels (red circles) and gammatones (light blue circles) as well as with a discrete Daubechies wavelet transform (dark blue squares) and a Fourier transform (black triangles). Confidence intervals are extremely tight (see text) and are not plotted.

efficient than those using fixed gammatones. At fidelities above 35 dB, the residual signal being coded is not significantly different from gaussian noise[10] and is described more efficiently with the Fourier or wavelet representation.

These results provide a theoretical explanation of the shape of auditory revcor filters. Previous explanations of the revcor filter shapes have been phenomenological, being a consequence of the impulse response function of the basilar membrane. Our results show that the rapid rise and slow decay that is characteristic of the revcor envelope is ideally matched to the statistical structure of natural sounds and indicate that it might not be an arbitrary feature of the biological system. In this sense, the physics of basilar membrane response (and subsequent cochlear processing) is tuned to the vibrations of objects in the natural environment. Rapid onsets followed by slower decays are characteristic of many types of transient natural sound and provide acoustic cues for essential auditory tasks such as sound localization. Note, however, that rapid rise and slow decay are not present in all efficient codes or an artefact of the learning algorithm, because the optimal kernels for vocalizations are largely symmetric and kernels adapted to reverse speech show the opposite pattern: slow rise and rapid decay (Supplementary Fig. S1).

Filterbank models of the cochlear have assumed that for a given frequency there is a fixed filter bandwidth (corresponding to gammatone function of fixed length). The auditory revcor data show a range of bandwidths at any given frequency, which is also matched by the kernel functions derived theoretically. One interpretation of this finding is that to form an efficient code for a variety of sounds it is necessary to have kernel functions of varying lengths to provide a better description of sounds with different temporal correlation constants.

We have compared the learned kernels with the physiological revcor filters, but it is not obvious why the comparison should be appropriate. Reverse correlation provides a first-order characterization of a nonlinear system, but there is no direct way to measure the 'features' that the auditory code is using to describe the acoustic signal. However, the use of reverse correlation to derive the equivalent revcor filters from the theoretical model yields an almost exact match to the learned kernel functions (Supplementary Fig. S3). For reverse correlation to recover the auditory nerve impulse response functions, neural spikes must be uncorrelated, which is often assumed to arise from stochastic firing. In contrast, spikes in the theoretical model are uncorrelated because they precisely encode non-redundant information. This indicates that if neural spikes are uncorrelated for the same reason, the revcor 'filters' might be more analogous to the acoustic features of the auditory code.

Our results depended on obtaining spike representations that are efficient. Learning experiments with an encoding algorithm that yields less efficient representations but is more biologically plausible (one in which the acoustic waveform was convolved with each of the kernel functions and thresholded to obtain the spike trains) produced kernel functions that showed no similarity to the auditory revcor functions (E.C.S and M.S.L., unpublished observations).

Although these results offer an explanation of auditory revcor data, we caution that there remain several challenges in extending the scope of the theory to auditory nerve responses. Algorithms for optimally encoding sound with binary spikes would permit a more direct comparison with auditory nerve firing patterns. A more accurate description of the encoding objective is also possible. Exact representation of the sound pressure waveform, as assumed here, is an unlikely goal, because not all acoustic information is behaviourally relevant. The gradual decrease phase-locking properties for higher-frequency nerve fibres might provide one indication of what information the auditory system encodes. There is also no explanation for the role of adaptation or changes in response with stimulus intensity, which are probably important in maximizing the information conveyed through the auditory nerve.

Finally, we note that deriving efficient codes for speech immediately yielded kernels that closely matched the auditory revcor filters. In contrast, obtaining similar results for the natural sounds ensemble required careful balancing of the different sound categories. This indicates that the acoustic composition of speech itself might be adapted to the mammalian auditory system.

METHODS

Encoding. Signals were encoding with a matching pursuit-based algorithm[10,12], which iteratively decomposed the signal in terms of the kernel functions. The current residual signal (or the original signal) was projected onto the dictionary of kernel functions. The projection with the largest inner product was subtracted out, and its coefficient and time were recorded. For the results reported here, the encoding was halted when s_i^m fell below a preset 'spiking' threshold. Further details are given in Supplementary Methods.

Learning. Equation (1) in the main text can be rewritten in probabilistic form in which we assume that the noise is gaussian and the prior provability of a spike, $p(s)$, is sparse. The kernel functions are optimized by performing gradient ascent on the approximate log data probability,

$$\begin{aligned}\frac{\partial}{\partial \phi_m}\log(p(x|\Phi)) &= \frac{\partial}{\partial \phi_m}\log(p(x|\Phi,\hat{s})) + \log(p(\hat{s})) \\ &= \frac{1}{2\sigma_e}\frac{\partial}{\partial \phi_m}\left[x - \sum_{m=1}^{M}\sum_{i=1}^{n_m}\hat{s}_i^m[x-\hat{x}]_{\tau_i^m}\right]^2 \\ &= \frac{1}{\sigma_e}\sum_i \hat{s}_i^m[x-\hat{x}]_{\tau_i^m}\end{aligned} \quad (2)$$

where $[x-\hat{x}]_{\tau_i^m}$ indicates the residual error over the extent of kernel ϕ_m at position τ_i^m. The estimated kernel gradient is thus a weighted average of the residual error. For training, 32 kernel functions were initialized as 100-sample gaussian noise, and the spiking threshold (minimum value of s_i^m) was set at 0.1. Further details are given in Supplementary Methods.

Sounds. The natural sounds ensemble used in training combined a collection of mammalian vocalizations with two classes of environmental sounds recorded by the authors in both natural and anechoic settings: ambient sounds (rustling brush, wind, flowing water) and transients (snapping twigs, crunching leaves, impacts of stone or wood). The reported 'natural sounds' results were based on (encoded) power proportions of 1.0:0.8:1.2, respectively, which directly reflects the impact of each sound class on kernel function adaptation. Speech was obtained from the TIMIT continuous speech corpus. All sounds were converted to a sampling frequency of 16 kHz, bandpass filtered to be between 100 and 6,000 Hz, and normalized to have a maximum amplitude of 1. Spike encodings of 5–40 s of training sounds were used to estimate the gradients on each update of the kernel functions.

Received 18 June; accepted 30 November 2005.

1. Barlow, H. B. in *Sensory Communication* (ed. Rosenbluth, W. A.) 217–234 (MIT Press, Cambridge, Massachusetts, 1961).
2. Atick, J. J. Could information-theory provide an ecological theory of sensory processing. *Network* **3,** 213–251 (1992).
3. Simoncelli, E. & Olshausen, B. Natural image statistics and neural representation. *Annu. Rev. Neurosci.* **24,** 1193–1216 (2001).
4. Laughlin, S. B. & Sejnowski, T. J. Communication in neuronal networks. *Science* **301,** 1870–1874 (2003).
5. Lewicki, M. S. Efficient coding of natural sounds. *Nature Neurosci.* **5,** 356–363 (2002).
6. Shamma, S., Chadwick, R., Wilbur, W. J., Morrish, K. A. & Rinzel, J. A biophysical model of cochlear processing: Intensity dependence of pure tone responses. *J. Acoust. Soc. Am.* **78,** 1612–1621 (1986).
7. Yang, X., Wang, K. & Shamma, S. Auditory representations of acoustic signals. *IEEE Trans. Inf. Theory* **38,** 824–839 (1992).
8. Lewicki, M. S. & Sejnowski, T. J. in *Advances in Neural Information Processing Systems* (eds Kearns, M. J., Solla, S. A. & Cohn, D. A.) vol. 11, 730–736 (MIT Press, Cambridge, Massachusetts, 1999).
9. Lewicki, M. S. in *Probabilistic Models of the Brain: Perception and Neural Function* (eds Rao, R. P. N., Olshausen, B. A. & Lewicki, M. S.) 241–255 (MIT Press, Cambridge, Massachusetts, 2002).
10. Smith, E. C. & Lewicki, M. S. Efficient coding of time-relative structure using spikes. *Neural Comput.* **17,** 19–45 (2005).
11. Davis, G., Mallat, S. & Avellaneda, M. Adaptive greedy approximations. *Construct. Approx.* **13,** 57–98 (1997).
12. Mallat, S. G. & Zhang, Z. Matching pursuits with time–frequency dictionaries. *IEEE Trans. Signal Process.* **41,** 3397–3415 (1993).
13. de Ruyter van Steveninck, R. & Bialek, W. Realtime performance of a

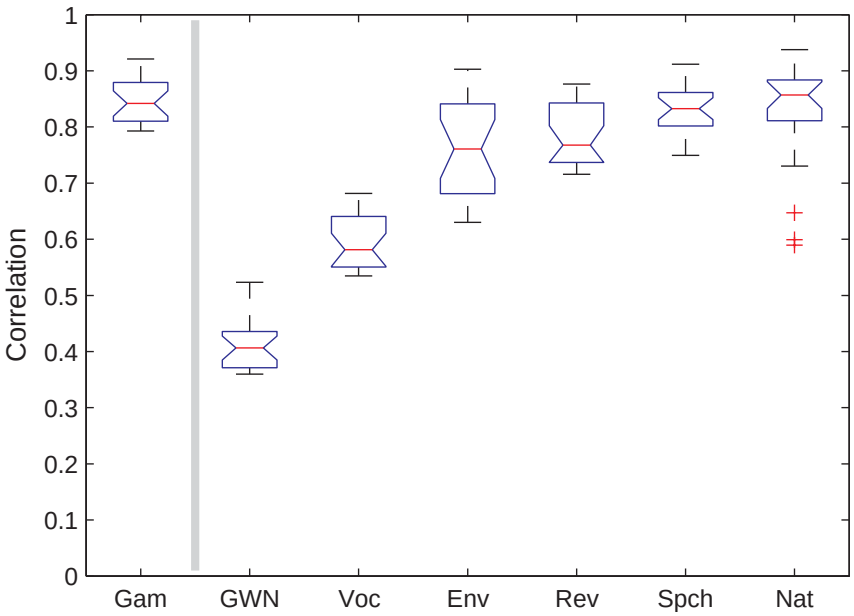
movement sensitive neuron in the blowfly visual system: Coding and information transfer in short spike sequences. *Proc. R. Soc. Lond. B* **234,** 379–414 (1988).
14. Olshausen, B. A. in *Probabilistic Models of the Brain: Perception and Neural Function* (eds Rao, R. P. N., Olshausen, B. A. & Lewicki, M. S.) 257–272 (MIT Press, Cambridge, Massachusetts, 2002).
15. Emmons, L. H., Whitney, B. M. & Ross, D. L. *Sounds of the Neotropical Rainforest Mammals* [audio CD] (Library of Natural Sounds, Cornell Laboratory of Ornithology, Ithaca, New York, 1997).
16. deBoer, E. & deJongh, H. On cochlear encoding: Potentialities and limitations of the reverse correlation technique. *J. Acoust. Soc. Am.* **63,** 115–135 (1978).
17. Carney, L. H. Sensitivities of cells in the anteroventral cochlear nucleus of cat to spatiotemporal discharge patterns across primary afferents. *J. Neurophysiol.* **64,** 437–456 (1990).
18. Recio-Spinoso, A., Temchin, A. N., van Dijk, P., Fan, Y.-H. & Ruggero, M. A. Wiener-kernel analysis of responses to noise of chinchilla auditory-nerve fibers. *J. Neurophysiol.* **93,** 3635–3648 (2005).
19. Irino, T. & Patterson, R. A level-dependent auditory filter: the gammachirp. *J. Acoust. Soc. Am.* **101,** 764–774 (1997).
20. Carney, L. H., McDuffy, M. J. & Shekhter, I. Frequency glides in the impulse responses of auditory-nerve fibers. *J. Acoust. Soc. Am.* **105,** 2384–2391 (1999).

Supplementary Information is linked to the online version of the paper at www.nature.com/nature.

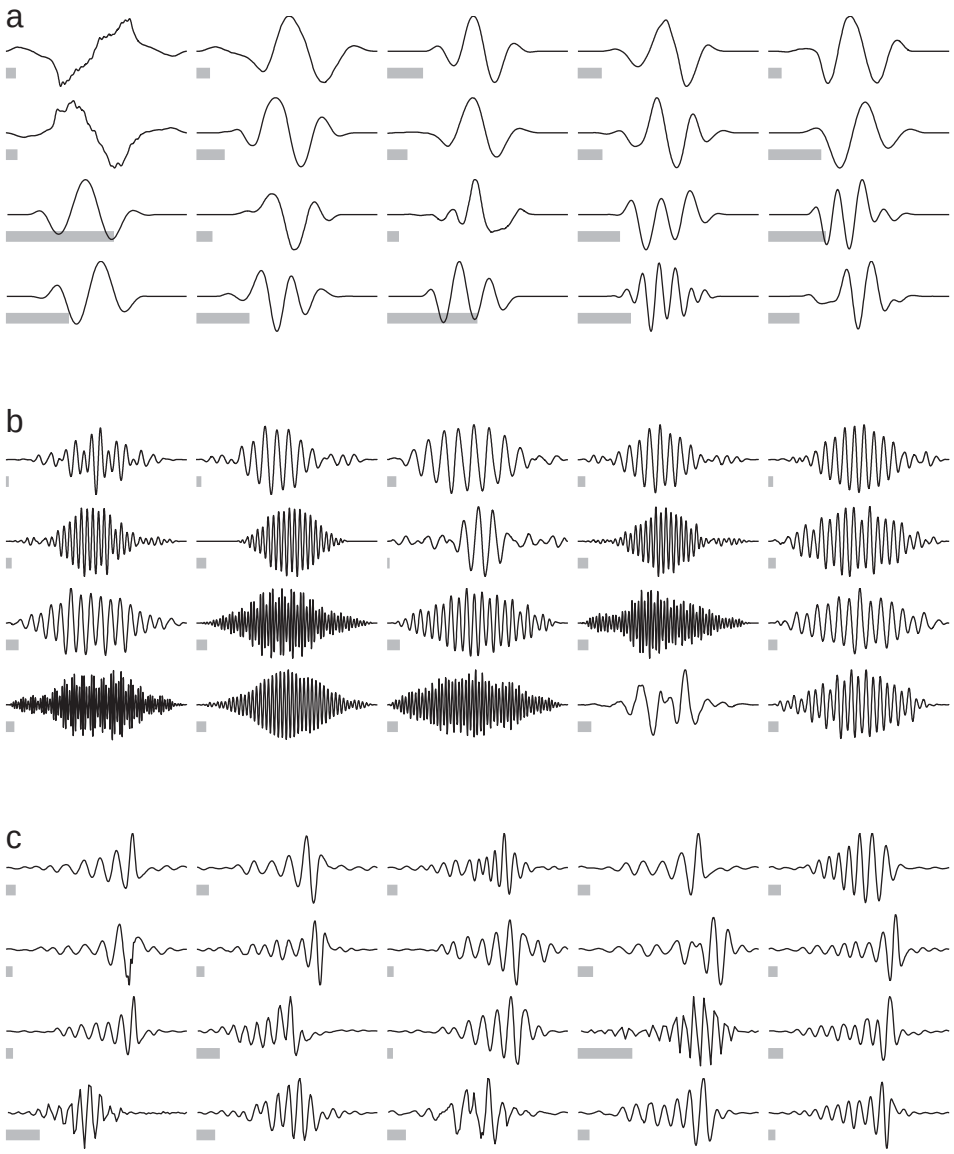
Acknowledgements E.C.S. was supported by a National Institutes of Health training grant. This material is based on work supported by a National Science Foundation grant to M.S.L. Empirical data in this paper were acquired from Boston University's Earlab, an online, freely accessible auditory database (http://earlab.bu.edu).

Author Contributions M.S.L. and E.C.S. developed the model, analysed the results and wrote the paper together; E.C.S. designed and ran the simulations.

Author Information Reprints and permissions information is available at npg.nature.com/reprintsandpermissions. The authors declare no competing financial interests. Correspondence and requests for materials should be addressed to M.S.L. (lewicki@cnbc.cmu.edu).



Supplementary Figure S2: A quantitative comparison of the correlation between cat revcor filters and different codes. Five separate codes were optimized to efficiently represent speech (Spch), natural sounds ensemble (Nat), reversed speech (Rev), environmental sounds (Env) and vocalizations (Voc). For each kernel function in a given code, the best matching revcor filter (as measured by correlation between the normalized function and filter) was found from the cat physiological data shown in Figures 2 and 3. Two additional codes are included for comparison. A dictionary of Gaussian noise functions (GWN) is used to define a baseline (i.e., this is the initialization point for each training run). Correlation coefficients were also computed for gammachirp functions (Gam), a parameterized model of cochlear filters, which represents a loose upper bound. As a gammachirp doesn't define a discrete code the correlation was computed after fitting a gammachirp to the revcor filters that best matched the speech-optimized kernel functions. The boxplot above shows the distribution of correlation coefficients of active kernel functions (see methods). The red line in each box is the median of the coefficient values for that code and the notch indicates the 95% confidence interval for the median. The 25^{th} and 75^{th} quartiles are marked by the lower and upper edges of the "box", while the "whiskers" indicate the 5^{th} and 95^{th} percentiles. Outliers are shown as red pluses. T-tests showed significant differences in the mean for both natural sounds and speech trained kernel functions compared against reversed speech ($p = 3.5 \times 10^{-7}$ and $p = 4.9 \times 10^{-6}$, respectively) and against environmental sounds ($p = 3.7 \times 10^{-12}$ and 0.04). Neither speech nor natural sounds kernels showed a significant difference in the mean compared to the gammachirp fitting. Efficient codes for speech and natural sounds are significantly better predictors of the cochlear code, approaching the fitted gammatone model in accuracy.



Supplementary Figure S1: Alternate efficient codes emerge when training on other sound ensembles. (a) Kernel functions trained on environmental sounds alone (without vocalizations) are extremely brief (gray scale bars = 5 ms), wide-band functions. This code reflects the short temporal correlations present in environmental sounds like rain (which is noise-like) or a snapping twig (a transient impulse). (b) When optimized to code vocalizations, kernel functions become extended, symmetric sinusoids with precise frequency selectivity. Animal vocalizations tend to have long, steady-state components with more defined harmonics. Long kernel functions capture more of this extended structure with fewer spikes, which leads to greater coding efficiency. (c) Reversed speech (where the order of each sample of a recorded speech signal is reversed) is a pseudo-natural sound class with first- and second-order statistics identical to normal speech. The leaned code for reversed speech is a reversed version of the speech code, asymmetric sinusoids with gradual onset and fast decay. Rate-fidelity comparisons show that speech-trained kernel functions are significantly less efficient than reverse-speech kernels for coding reversed speech ($\sim 3\%$ decrease in SNR at all rates), demonstrating that the higher-order structure represented by the kernel envelopes plays a significant role in the efficiency of these codes.

Supplemental Methods

Encoding. Given a kernel function, ϕ_m, signals can be decomposed into a projection and a residual

$$x(t) =< x(t), \phi_m > \phi_m + R_x(t), \tag{1}$$

where $< x(t), \phi_m >$ is the inner product between the signal and the kernel and is equivalent to s_i^m in equation 1 from the main text. The final term in equation 1, $R_x(t)$, is the residual signal after approximating $x(t)$ in the direction of ϕ_m. The projection with the largest magnitude inner product will minimize the power of $R_x(t)$, thereby capturing the most structure possible *with a single kernel*. Matching pursuit applies this decomposition iteratively. Equation 1 can be rewritten more generally as

$$R_x^n(t) =< R_x^n(t), \phi_m > \phi_m + R_x^{n+1}(t), \tag{2}$$

with $R_x^0(t) = x(t)$ at the start of the algorithm. On each iteration, the current residual is projected onto the dictionary of kernel functions. The projection with the largest inner product is subtracted out, and its coefficient and time are recorded. For the results reported here, the encoding was halted when s_i^m fell below a preset "spiking" threshold.

Learning. Equation 1 in the main text can be rewritten in probabilistic form as

$$\begin{aligned} p(x|\Phi) &= \int p(x|\Phi, s)p(s)ds & (3)\\ &\approx p(x|\Phi, \hat{s})p(\hat{s}) & (4) \end{aligned}$$

where $\hat{s}$, an approximation of the posterior maximum, comes from the set of coefficients generated by matching pursuit. We assume the noise in the likelihood, $p(x|\Phi, \hat{s})$, is Gaussian and the prior, $p(s)$, is sparse (note, though, that the prior has no direct influence on the learning and sparseness is obtained approximately via the encoding algorithm). The kernel functions are optimized by doing gradient ascent on the approximate log data probability,

$$\begin{aligned} \frac{\partial}{\partial \phi_m} \log(p(x|\Phi)) &= \frac{\partial}{\partial \phi_m}[\log(p(x|\Phi, \hat{s})) + \log(p(\hat{s}))] & (5)\\ &= \frac{-1}{2\sigma_\varepsilon^2} \frac{\partial}{\partial \phi_m} \parallel x - \sum_{m=1}^{M} \sum_{i=1}^{n_m} \hat{s}_i^m \, \phi_m(t - \tau_i^m) \parallel^2 & (6)\\ &= \frac{1}{\sigma_\varepsilon^2} \sum_i \hat{s}_i^m [x - \hat{x}]_{\tau_i^m} & (7) \end{aligned}$$

where $[x - \hat{x}]_{\tau_i^m}$ indicates the residual error over the extent of kernel ϕ_m at position τ_i^m. The estimated kernel gradient is thus a weighted average of the residual error.

At the start of training, 32 kernel functions were initialized as 100-sample Gaussian noise, and the spiking threshold (minimum value of s_i^m) was set at 0.1, which allowed for an initial encoding of ~ 12 dB signal-to-noise ratio (SNR).

During learning we allowed the length of each kernel to vary such that low frequency functions and others requiring longer temporal extent could grow from shorter initial seeds, while briefer functions could be trimmed to speed processing and minimize the effects of over-fitting. Each kernel function was zero-padded on both ends by $1/10$ its total length. If a gradient step caused elements within the padding to exceed a threshold, then the padding was extended. If elements at either end of a function fell below threshold the padding was trimmed.

Learning in this algorithm is activity dependent, i.e., kernel functions adapt in proportion to their "spike rate". This means that some functions learn very slowly or not at all. Typically, $\sim 10\%$ of kernel functions are largely inactive during training (though the proportion varies by sound class) and differ very little from their initialization point. Since inactive kernels play no role in the spiking population code and do not reflect the statistics of their training sounds, we discarded any whose activity, $\sum |\hat{s}^m|$, was less than 10% of the median activity for the whole population.

Sounds. The natural sounds ensemble used in training combines a collection of mammalian vocalizations with two classes of environmental sounds recorded by the authors in both natural and anechoic settings: ambient sounds (rustling brush, wind, flowing water) and transients (snapping twigs, crunching leaves, impacts of stone or wood). The reported "natural sounds" results were based on an (encoded) power ratio of 1.0:0.8:1.2, respectively, which directly reflects the impact of each sound class on kernel function adaptation. Speech was obtained from the TIMIT continuous speech corpus. All sounds were converted to a sampling frequency of 16 kHz, bandpass filtered to be between 100-6000 Hz, and normalized to have a maximum amplitude of 1. Spike encodings of 5-40 seconds of training sounds were used to estimate the gradients on each update of the kernel functions.

Rate-Fidelity. Computing the rate-fidelity curves began with associated pairs of coefficients and time values, $\{s_i^m, \tau_i^m\}$, which are initially stored as 32-bit variables. Storing the original time values referenced to the start of the signal is costly because their range can be arbitrarily large and the distribution of time points is essentially uniform. Storing only the time since the last spike, $\delta\tau_i^m$, greatly restricts the range and produces a variable that approximately follows a gamma distribution. A uniform quantizer was used to vary the precision of the $\{s_i^m, \tau_i^m\}$ values between 1 and 16 bits. At all levels of precision, the bin widths for quantization are selected so that equal numbers of values fall in each bin, and all values in a bin are then replaced with their mean value. s_i^m and τ_i^m are quantized independently.

Treating the quantized values as samples from a random variable, we estimate a code's entropy (bits/coefficient) from histograms of the values. Rate is then the product of the estimated entropy of the quantized variables and the number of coefficients per second for a given signal. At each level of precision the signal is reconstructed based on the quantized values, and an SNR for the code is computed based on the residual error. This process was repeated across the TIMIT test set and the results were averaged to produce rate-fidelity curves. Fourier coefficients were obtained

for each signal via Fourier transform. The real and imaginary parts were quantized independently, and the rate was based on the estimated entropy of the quantized coefficients. Reconstruction was simply the inverse Fourier transform of the quantized coefficients. Similarly, coding efficiency using Daubechies wavelets was estimated using Matlab's discrete wavelet transform and inverse wavelet transform functions. The kernel functions were trained on the TIMIT training set; rate-fidelity curves were computed using the TIMIT testing set.

Non-linear nature of the spike coding algorithm

Non-linear coding is necessary when the fundamental structure of the signal lies in a non-linear subspace, i.e. one that is not a linear projection of the original subspace. The smaller this subspace in relation to the dimensionality of the original signal space, the greater the potential coding efficiency. In the case of sounds, if we imagine an underlying generative model in which acoustic events occur at random (and sparse) times, then coding is the inference process of recovering the times and waveforms of the original acoustic events. If the acoustic events are rare, (so that they rarely overlap), and the noise low, this is a relatively easy non- linear inference problem. A simple filter and threshold algorithm will suffice to recover the true underlying structure of the acoustic signal. As the events become more common or the noise becomes higher, the inference problem becomes harder.

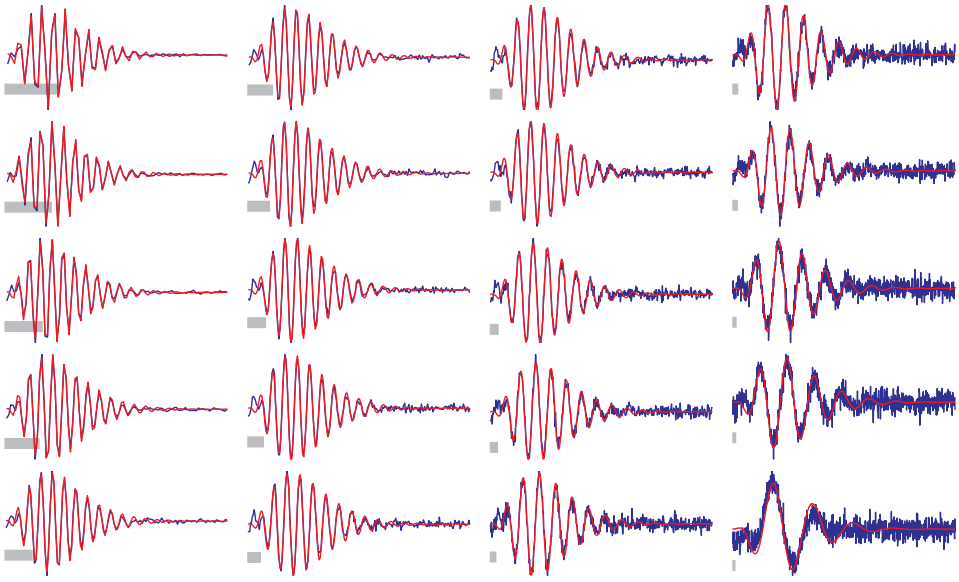
The real acoustic environment is not a mixture of discrete events, but, as the spike coding model demonstrates, it can be efficiently described that way. In this case, the acoustic events are not actually present in the external environment, but instead they acoustic features that occur sparsely in time. The actual degree of sparseness can be measured by making histograms of the spike time intervals. This analysis shows that spikes follow a gamma distribution with an average mode of about 10 milliseconds (the actual distribution varies as a function of the kernel length). In 10 milliseconds there are 160 samples at 16 kHz, so in these discrete terms, each spike is being placed on average in 1 out of 160 possible positions. The spike coding algorithm fundamentally non-linear, because it must determine which of these positions is best. Furthermore, the position of each spike must be coordinated with other spikes so that the information in the waveform is represented non-redundantly. The spike coding algorithm described in the main text is an efficient method of finding an approximate solution to this problem. Quasi-linear models such as integrate and fire (or convolution followed by probabilistic spiking) lead to redundancy in both time and across kernels and thus do not yield efficient codes. It has been shown that this simple spike coding model yields a typical coding fidelity of less than 10 dB SNR. Purely linear models, such as a bank of filters, are even less efficient, because they transform a single analog waveform into a set of waveforms.

Comparison of learned kernels and auditory revcor filters

Auditory revcor filters are estimates of the impulse response functions of auditory nerves, and are thus a first-order characterization of the auditory nerve response. Are these comparable to the spike code kernels? The goal the spike coding algorithm is to determine the precise temporal locations of the underlying acoustic features represented by the kernels. The first step in this algorithm is to convolve (or filter) the signal with kernels. Convolution is also the first step in the implicit model used in reverse correlation. The two models differ in how the analog filter output is converted to a spike train. In the case of the revcor model, spiking is simply a probabilistic function of the filter output. In the case of spike coding, the spikes are chosen so as to represent the signal with max-

imal efficiency. In both cases, the kernels (i.e. the acoustic features) can be recovered by reverse correlation (spike-triggered averaging) in response to Gaussian noise, because the resulting spike trains are uncorrelated. For the revcor spiking model, the spike times are uncorrelated, because the input signal is uncorrelated and probabilistic spiking does not introduce any correlations. In the case of the spike coding algorithm, the spikes are uncorrelated because the code is non-redundant (both across time for an individual kernel and across kernel functions, although the later is irrelevant for purposes of reverse correlation). Thus, for each of these models, we would expect reverse correlation to recover the underlying acoustic features. Supplementary figure S3 confirms that this is the case for the spike coding model.

What about the case of an unknown non-linear coding algorithm (such as the peripheral auditory system)? The revcor filters themselves do not directly tell us whether the auditory spike code is efficient – that would require an analysis of the auditory spike trains in both time and across nerve fibers and a more detailed model of how the acoustic waveform is encoded by binary spikes – but the close match between the theoretically ideal kernels and the auditory revcor filters is consistent with what would be expected in a system that was forming an efficient spike code.



Supplementary Figure S3: Characterization of spike code kernel functions by reverse correlation. A neuron's estimated transfer function, called a revcor filter in the auditory literature, is produced by cross-correlating a white noise input signal with the cell's output spike train. In an identical fashion, we can apply reverse correlation to the spike coding model, cross-correlating a white noise input signal with the analog spike output. This gives a description of the spike coding model in terms of revcor filters, which can be compared directly to the physiological revcor filters. The figure above shows kernel functions (red), which we initialize as a set of gammatones, and the estimated revcor filter (blue) for each channel. The filters are virtually identical to the shape of the kernel functions from which they are estimated, despite the significant nonlinearities of matching pursuit.