

ePPR: a new strategy for the characterization of sensory cells from input/output data

JOAQUÍN RAPELA¹, GIDON FELSE², JON TOURYAN³,
JERRY M. MENDEL¹, & NORBERTO M. GRZYWACZ^{1,4,5}

¹Department of Electrical Engineering, University of Southern California, Hedco Neuroscience Building, Los Angeles, CA, USA, ²Department of Physiology & Biophysics, University of Colorado, Denver, CO, USA, ³SAIC-Boulder, Louisville, CO, USA, ⁴Department of Biomedical Engineering, University of Southern California, Los Angeles, CA, USA, and ⁵Neuroscience Graduate Program, University of Southern California, Los Angeles, CA, USA

(Received 23 April 2009; accepted 22 April 2010)

Abstract

A central goal of systems neuroscience is to characterize the transformation of sensory input to spiking output in single neurons. This problem is complicated by the large dimensionality of the inputs. To cope with this problem, previous methods have estimated simplified versions of a generic linear-nonlinear (LN) model and required, in most cases, stimuli with constrained statistics. Here we develop the extended Projection Pursuit Regression (ePPR) algorithm that allows the estimation of all of the parameters, in space and time, of a generic LN model using arbitrary stimuli. We first prove that ePPR models can uniformly approximate, to an arbitrary degree of precision, any continuous function. To test this generality empirically, we use ePPR to recover the parameters of models of cortical cells that cannot be represented exactly with an ePPR model. Next we evaluate ePPR with physiological data from primary visual cortex, and show that it can characterize both simple and complex cells, from their responses to both natural and random stimuli. For both simulated and physiological data, we show that ePPR compares favorably to spike-triggered and information-theoretic techniques. To the best of our knowledge, this article contains the first demonstration of a method that allows the estimation of an LN model of visual cells, containing multiple spatio-temporal filters, from their responses to natural stimuli.

Keywords: *Single neuron computation, natural scenes, visual system*

Correspondence: Joaquín Rapela, Department of Electrical Engineering, University of Southern California, Hedco Neuroscience Building, Los Angeles, CA 90089-2520, USA. E-mail: rapela@usc.edu

Introduction

One of the most important problems in neuroscience is to functionally characterize how sensory neurons transform their input to spiking output. Two central issues for these characterizations are the stimuli used to probe a cell and the model used to represent it. Regarding the stimuli, sensory neurons have been traditionally studied with small sets of simple stimuli, specifically designed to probe certain aspects of their response properties (e.g., Hartline 1940; Barlow 1953; Kuffler 1953; Liberman 1982), or with large sets of random stimuli (e.g., P. Marmarelis and Marmarelis 1978). However, recent work (Theunissen et al. 2000; David et al. 2004; Felsen et al. 2005; Wooley et al. 2006; Sharpee et al. 2006; Sharpee et al. 2008) has shown that observable properties of sensory cells depend on the statistical properties of the stimuli used to probe them. Therefore, to understand how sensory cells operate in natural conditions, it is important to characterize them from their responses to natural stimuli (Felsen and Dan 2005). Regarding the model, cells of different classes are normally characterized with different parametric models. Based on prior evidence, a model that contains the relevant structure for a class of cells is proposed, and parameters of this model are fitted to experimental data from a cell of this class. A problem with this approach is that the hypothesized model structure may not be correct, making the interpretation of the fitted parameters questionable. An alternative is to use generic models that can well approximate cells from a large set of classes. Here we address both of these issues by developing a method to estimate a generic model that can characterize responses of many classes of cells to arbitrary, including natural, stimuli. We focus the description and evaluation of this method on the visual system, though the method is applicable to other sensory modalities.

Using natural images to characterize visual cells is non-trivial. Natural images are complex (Simoncelli and Olshausen 2001), so the number of descriptors needed to represent them is large, and a generic model would need a very large number of parameters to characterize responses of visual cells to natural stimuli. Due to the curse of dimensionality (Bellman 1961), the amount of data required to estimate the parameters of a generic model grows exponentially with the number of parameters. Therefore, a prohibitively large amount of data—unattainable in standard physiology experiments—would be required to estimate the parameters of generic models of visual cells using natural stimuli as inputs.

A common strategy to overcome this problem is to assume that the response of a cell follows an LN model (Wiener 1958; de Boer and Kuyper 1968; Marmarelis 1993; Chichilnisky 2001; Simoncelli et al. 2004). A generic version of this model is shown in Figure 1. The response of the model at time bin i is assumed to depend on the image presented at time bin i , plus the images presented at the previous D time bins. At each delay d , the model contains M_d filters. To generate its response, the model computes the dot product between the input image at delay d and each of the M_d filters at that delay, generating scalars $g_{1,d}, \dots, g_{M_d,d}$. Then the scalars at all delays, $\{g_{m,d} \mid 0 \leq d \leq D, 1 \leq m \leq M_d\}$, are used as inputs to a nonlinear function N that predicts the cell's spike rate at time bin i^1 .

However, even this generic LN model has a large number of parameters. For instance, for images of size 32×32 pixels, a model with $M_d = 3$ for all d and $D = 4$ will contain $32 \times 32 \times 3 \times (1 + 4) = 15,360$ filter parameters, as well as additional parameters needed to describe the nonlinear function N . To avoid problems

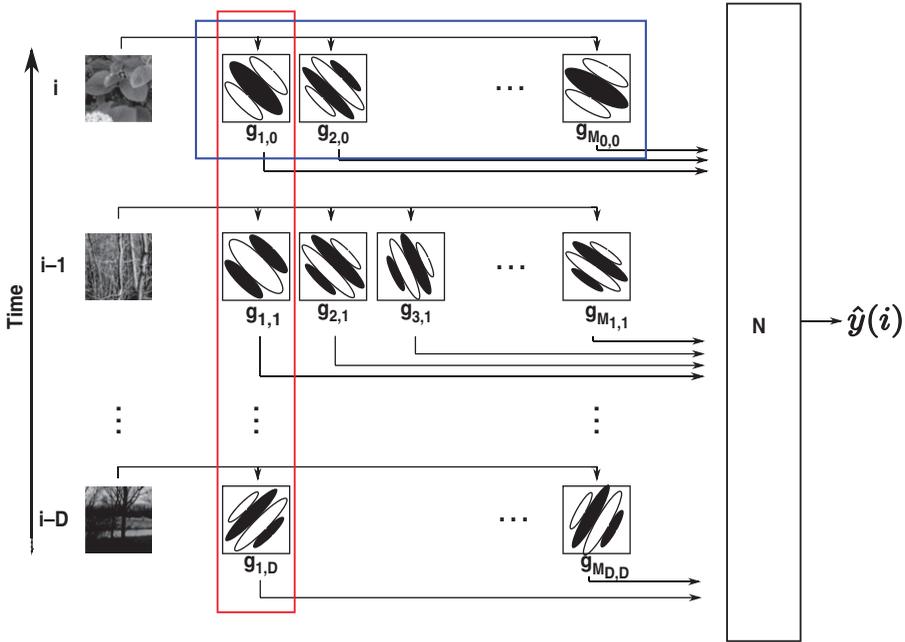


Figure 1. A generic LN model. The prediction of the model at time bin i , $\hat{y}(i)$, depends on the image presented at time bin i , plus the images presented at the previous D time bins. At each delay d , the model contains M_d filters. To generate its output, the model projects the input image at delay d on the M_d filters at that delay, generating scalars $g_{1,d} \dots g_{M_d,d}$. Then, the scalars at all delays, $\{g_{m,d} \mid 0 \leq d \leq D, 1 \leq m \leq M_d\}$, are used as inputs to a nonlinear function N that predicts the cell's spike rate at time bin i . Because this model has too many parameters, previous methods have estimated simplified versions of it. The red box includes the filters of the models estimated by Chichilnisky et al. (2001) and by Sharpee et al. (2006, 2008). The blue box includes the filters of the models estimated by Touryan et al. (2005) and by Rapela et al. (2006).

associated with estimating such a large number of parameters, previous techniques (Chichilnisky 2001; Sharpee et al. 2004, Touryan et al. 2005, Rapela et al. 2006) have estimated simplified versions of this model. Spike-triggered average (STA; de Boer and Kuyper 1968; Chichilnisky 2001) can only estimate one filter per delay, i.e., $M_d=1$ for all d (Figure 1, red box), and requires a radially symmetric distribution of the input images (Chichilnisky 2001; Paninski 2003). Spike-triggered covariance (STC; de Ruyter van Steveninck and Bialek 1988) allows the estimation of multiple filters, but has more stringent convergence conditions than STA, in that the distribution of the input images must be Gaussian (Paninski 2003). Thus, neither STA nor STC work with natural stimuli. Touryan et al. (2005) and Rapela et al. (2006) proposed techniques that can use natural stimuli, but neglect the temporal dimension of the inputs; i.e., $D=0$ (Figure 1, blue box). The method introduced by Sharpee et al. (2004) can estimate spatio-temporal models from responses of visual cells to natural images. But, with physiological data, this method has only been used to estimate models with one filter per delay; i.e., $M_d=1$ for all d (Sharpee et al. 2006, 2008, Figure 1, red box). Using filters estimated with STC and a parametric model for the nonlinear function, Rust et al. (2005) estimated all

parameters of the generic LN model in Figure 1, but at the cost of restricting the stimulus ensemble to vary along a single spatial dimension (i.e., binary random bars), and constraining the statistics of the inputs to be Gaussian white noise.

An alternative to simplifying the general LN model in Figure 1, or constraining the statistics of its inputs, is to use an efficient optimization strategy to enable the estimation of all of its parameters using arbitrary inputs. The Projection Pursuit Regression algorithm (PPR; Friedman and Stuetzle 1981) provides one such strategy. By decomposing a high-dimensional estimation problem into a sequence of lower dimensional ones, PPR is one of the few multivariate methods able to bypass the curse of dimensionality (Huber 1985). In Rapela et al. (2006) we showed that PPR compared favorably with previous methods for the spatial characterization of visual cells; i.e., characterizations as a function of only one image presented to the cell before its response. However, responses of a visual cells are not spatial, but spatio-temporal; i.e., they depend on several images presented to the cell before its response. For spatio-temporal characterizations of visual cells, the efficient optimization strategy of PPR is not sufficient to escape the curse of dimensionality. Here, we introduce the extended Projection Pursuit Regression (ePPR) algorithm, which extends the PPR algorithm to allow the estimation of all of the parameters of the model in Figure 1 using natural images. In addition, the ePPR estimation algorithm is designed to be robust to correlations in natural images, and the ePPR estimation problem is regularized to allow estimations using images of large size as inputs.

The ePPR model is very general. Below we prove that it can uniformly approximate, to an arbitrary degree of precision, any continuous function with inputs in the unit cube. To test this generality empirically, we use ePPR to recover the parameters of an LN model of a complex cell with divisive normalization, and a linear-nonlinear-linear (LNL) extension of it, neither of which can be represented exactly with an ePPR model. Next we test this generality with physiological data, by using ePPR to characterize cortical complex and simple cells. ePPR models can be estimated using stimuli with arbitrary statistics. To validate this, for the simulated and cortical cells studied in this article, we compare ePPR models estimated from their responses to natural and random data.

The rest of the article is organized as follows. The next section “Extended Projection Pursuit Regression” summarizes the ePPR algorithm (Details are provided in Appendix B). The following section “Simulated cell” presents the results of the application of ePPR to recover the parameters of the simulated cells. Next, we use ePPR to characterize a complex cell (Section “Complex cell”) and simple cell (Section “Simple cell”). We discuss advantages and disadvantages of ePPR, and draw final conclusions, in the final section “Discussion”. Supplementary information is provided in the appendices.

Extended projection pursuit regression

This section introduces the ePPR algorithm. Because ePPR extends PPR, the next subsection summarizes the PPR algorithm. The following subsections then describe the extensions introduced to PPR. Detailed algorithmic descriptions of PPR and ePPR are given in Appendix A and Appendix B, respectively.

Projection pursuit regression

The PPR model is show in Equation 1. For an input $x_i \in \mathbb{R}^n$ the output of the PPR model $\hat{y}(i) \in \mathbb{R}$ is given by the mean response \bar{y} plus M_0 terms of the form $\beta_m \phi_m(\alpha_m^T x_i)$. In these terms, $\beta_m \in \mathbb{R}$ is an importance coefficient, and ϕ_m is a uni-dimensional nonlinear function acting on the dot product between the unit-norm direction $\alpha_m \in \mathbb{R}^n$ and x_i .

$$\hat{y}(i) = \bar{y} + \sum_{m=1}^{M_0} \beta_m \phi_m(\alpha_m^T x_i)$$

$$\text{with } \|\alpha_m\|_2 = 1, \quad \frac{1}{n} \sum_{i=1}^n \phi_m(\alpha_m^T x_i) = 0, \quad \text{and} \quad \frac{1}{n} \sum_{i=1}^n \phi_m^2(\alpha_m^T x_i) = 1 \quad (1)$$

The algorithm used to estimate the parameters of the PPR model finds the prediction function \hat{y} minimizing the sum of squared errors (SSE) with the response function y (Equation 2). To escape from the curse of dimensionality, it estimates one term of Equation 1 at a time, as follows: Having determined the functions $\phi_1, \phi_2, \dots, \phi_{m-1}$ and the unit vectors $\alpha_1, \alpha_2, \dots, \alpha_{m-1}$, the estimation algorithm chooses a unit vector α_m and a function ϕ_m that minimize the SSE in Equation 3. To avoid local minima in this greedy procedure, PPR uses a backward stepwise procedure (Appendix A).

$$SSE = \sum_{i=1}^N (y(i) - \hat{y}(i))^2 \quad (2)$$

$$SSE = \sum_{i=1}^N (r_m(i) - \phi_m(\alpha_m^T x_i))^2, \quad \text{where } r_m(i) = y(i) - \sum_{j=1}^{m-1} \beta_j \phi_j(\alpha_j^T x_i) \quad (3)$$

Approximation theory results for PPR.

What types of functions can be well approximated by PPR models?

PPR models can represent exactly a large class of functions. For instance, any multi-dimensional polynomial can be represented exactly by a PPR model (Proposition 1 in Appendix C). Although not all functions admit an exact PPR representation, any continuous function, with inputs in $[0, 1]^p$, can be uniformly approximated, to an arbitrary degree of precision, by a PPR model. This follows from (1) the Stone-Weierstrass theorem (Rudin 1976), which implies that polynomials are dense on $[0, 1]^p$, and (2) the previous result that polynomials can be represented exactly by PPR models.

It is conceivable that a given function could be well approximated by a PPR model but that the greedy PPR estimation algorithm does not converge to the optimal approximation. So, the following question becomes relevant.

Under what conditions will the PPR estimation algorithm converge to its best approximation?

For inputs x , samples from a probability distribution P uniform on the unit ball or multivariate Gaussian, Donoho et al. (1985) announced a proof of

strong convergence; i.e., $r_m \rightarrow 0$ in the norm of $L_2(P)$. In addition, Jones (1987) proved strong convergence for general P , when the nonlinear functions are given by the conditional expectations (Equation 4) and the estimated projection directions are uniformly close to the optimal greedy projection directions (Equation 5).

$$\phi_m(z) = E(r_m(X)|\alpha_m^T X = z) \quad (4)$$

$$E(\phi_m(\alpha_m^T X))^2 > \rho \sup_{b^T b=1} E(\phi_m(b^T X))^2, \quad \rho \text{ fixed, } 0 < \rho < 1. \quad (5)$$

However, with empirical data these conditions cannot be verified and one cannot know if PPR has converged to the optimal solution.

Spatio-Temporal models

In Rapela et al. (2006), we estimated spatial PPR models of visual cells. The responses of these models depended on a single image presented to the cell before its response. However, we would like to estimate spatio-temporal PPR models; i.e., models whose response depends on several past images, as is known to be the case for most visual cells.

To predict the response of a cell at time bin i , in Rapela et al. (2006) we used as input to PPR the vector representation of the image presented at time bin i^{th} . Calling $I_i \in \mathbb{R}^{p \times p}$ the image presented at time i , and $\text{vec} : \mathbb{R}^{p \times p} \rightarrow \mathbb{R}^{p^2}$ the operation that transforms an image into its vector representation, the input to PPR in Rapela et al. (2006) was $x_i = \text{vec}(I_i) \in \mathbb{R}^{p^2}$.

A first type of spatio-temporal model uses the spatial PPR model in Equation 1, but taking as input the concatenation of the images presented at time bins $\{i, \dots, i-D\}$; i.e., $x_i = [\text{vec}(I_i)^T, \dots, \text{vec}(I_{i-D})^T]^T \in \mathbb{R}^{(D+1)p^2}$. As PPR models, spatio-temporal models of this type can uniformly approximate, to an arbitrary degree of precision, any continuous functions with inputs in the unit cube. A limitation of this type of model is that the dimensionality of the input to each term in Equation 1 grows in proportion to the memory D of the model, which complicates the estimation of the filters α_m . Also, to build these models, the memory D must be determined in advance.

A second type of spatio-temporal model is obtained by adding to Equation 1 extra terms operating on images at different delays, as shown in Equation 6. The estimation algorithm for this type of model requires only simple modifications to the PPR estimation algorithm (Appendix B). An advantage of this type of model is that the dimensionality of the input to each term in Equation 1 does not grow with the memory D of the model, simplifying the estimation of the filters. In addition, the estimation algorithm learns, in a single run, the optimal memory, D , required to characterize a cell. A disadvantage is that Equation 6 is not the most general spatio-temporal extension of the PPR model. In particular, pixels of images occurring at different delays are contained in different terms in Equation 6, and different terms are combined linearly. Therefore, nonlinear interactions between pixels of images at different delays cannot influence the predictions of this type

of model.

$$\hat{y}(i) = \bar{y} + \sum_{d=0}^D \sum_{m=1}^{M_d} \beta_{m,d} \phi_{m,d}(\alpha_{m,d}^T x_{i-d})$$

$$\text{with } \frac{1}{n} \sum_{i=1}^n \phi_{m,d}(\alpha_{m,d}^T x_{i-d}) = 0, \quad \text{and} \quad \frac{1}{n} \sum_{i=1}^n \phi_{m,d}^2(\alpha_{m,d}^T x_{i-d}) = 1 \quad (6)$$

We call models of the first type ePPR models with time interactions, and models of the second type ePPR models without time interactions. Selecting between these types of models presents a tradeoff between model generality and optimization feasibility. Models with time interactions are very general, but the estimation of its filters, from realistic amounts of physiological data, becomes infeasible for large memory D . On the other hand, models without time interactions are less general, but allow the estimation of models of visual cells with large memory D . When the memory of a cell is short, and/or the response sampling rate is small, and/or there is enough physiological data to make possible the estimation of the larger spatio-temporal filters, then models with time interactions are preferable. One possibility to determine the memory of the cell is to estimate an ePPR model without time interactions. Then, if the memory, D , of this model is sufficiently short, or the amount of data is sufficiently large, a more general ePPR model with time interactions can be estimated.

In what follows we will refer to Equation 6 as the ePPR model, with the caveat that for models with time interactions several images are concatenated into a single input, as indicated above, and the memory of the model is set to $D = 0$.

Robustness to correlations in natural images

With simulated data we observed that the filters of PPR models estimated from natural data were significantly worse than those estimated from random data. As noted in Appendix A, a key step in PPR is the solution of a nonlinear least-squares problem for estimating a projection direction α_m (minimization of the SSE in Equation 15). In PPR, this problem is solved using the Gauss-Newton method (Nocedal and Wright 2006), which, in turn, requires the solution of a linear system of equations $Ax = b$ (Equation 18). The Gauss-Newton method is guaranteed to converge if the eigenvalues of A are bounded away from zero (Nocedal and Wright 2006). This is the case for the matrix A constructed from random data, but not for the one constructed from natural data, which cause some eigenvalues of A to be zero. Thus, for natural data, the Gauss-Newton method is not guaranteed to converge. To overcome this problem, we replaced it with a Trust Region method (Nocedal and Wright 2006). The latter algorithm is guaranteed to converge under very general conditions, which do not require non-zero eigenvalues (Nocedal and wright 2006).

Smooth prior for large-dimensional filters

Responses of visual cells contain high levels of noise. Estimating ePPR parameters by minimizing Equation 2 then leads to estimates that overfit the noise; i.e., narrowly focusing on the training error is likely to return estimates that describe the

training set well, but perform poorly in predicting responses to novel data. A common strategy to overcome this problem is to penalize the estimated function, \hat{y} , based on some a priori measure of how likely \hat{y} is to have overfit noise. In ePPR, \hat{y} is penalized for containing non-smooth projection directions $\alpha_{m,d}$. The ePPR objective function (Equation 7) contains a penalty term $\lambda \sum_{d=0}^D \sum_{m=1}^{M_d} \|L\alpha_{m,d}\|^2$. In this term, L is a smoothing operator such that $\|L\alpha_{m,d}\|^2$ will be large when $\alpha_{m,d}$ is non-smooth. In turn, the regularization parameter λ controls the tradeoff between fitting the responses accurately and estimating smooth filters $\alpha_{m,d}$. For the results shown below, we chose L to be a 3×3 Laplacian operator. To select the parameters λ , we used the procedure described in Appendix B.2.

$$\mathcal{J} = \sum_{i=1}^N (y_i - \hat{y}(x_i))^2 + \lambda \sum_{d=0}^D \sum_{m=1}^{M_d} \|L\alpha_{m,d}\|^2 \quad (7)$$

In summary, ePPR fits the spatio-temporal model in Equation 6 by optimizing the criterion in Equation 7, using a Trust Region method that is robust to correlations in natural images. An implementation of the ePPR algorithm can be downloaded from <http://vpl.usc.edu/projects/ePPR/>.

Simulated cell

Here we evaluate ePPR for the characterization of a simulated complex cell. For the simulations we used a simplified version of a divisive gain control model, as have been used to describe nonlinear properties of neurons in primary visual cortex (Heeger 1992). The mean response of the simulated cell at time i is given by $\bar{y}(i)$ in Equation 8, where $[x(i)^T, x(i-1)^T, x(i-2)^T]$ stands for the (transpose of the) concatenation of the images at time i , $i-1$, and $i-2$. The noisy response is a Poisson random variable with this mean. The spatio-temporal filters, f_1, f_2, f_3 , used in this model are shown in Figure 2a. The filters f_1 and f_2 are facilitatory, because larger dot products with these filters produce stronger mean responses, while the filter f_3 is suppressive, because larger dot products with this filter produce weaker mean responses. The parameter γ controls of the mean of the responses and, because the noise is Poisson, it also controls the level of noise in the responses. To study the effect of noise in the estimated models, we varied γ to generate four sets of 24,000 responses with different levels of noise. For all sets we fixed the inhibitory constant ω so that the mean of the denominator in Equation 8, for all simulated responses, was 4.26; i.e., on average the divisive normalization reduced the unnormalized response of the cell by more than four times. We study the effects of varying the amount of inhibition in Appendix E. Further details on the simulation procedure are given in Section “Methods: Simulated responses”. Note that, due to the divisive normalization, Equation 8 cannot be represented exactly by an ePPR model. Thus, this model tests the generalization of ePPR. To further test this generalization, in subsection “Linear-nonlinear-linear model” we use ePPR to characterize an LNL extension of this complex cell model.

$$\bar{y}(i) = \gamma \frac{([x(i)^T, x(i-1)^T, x(i-2)^T]f_1)^2 + ([x(i)^T, x(i-1)^T, x(i-2)^T]f_2)^2}{1 + \omega([x(i)^T, x(i-1)^T, x(i-2)^T]f_3)^2} \quad (8)$$

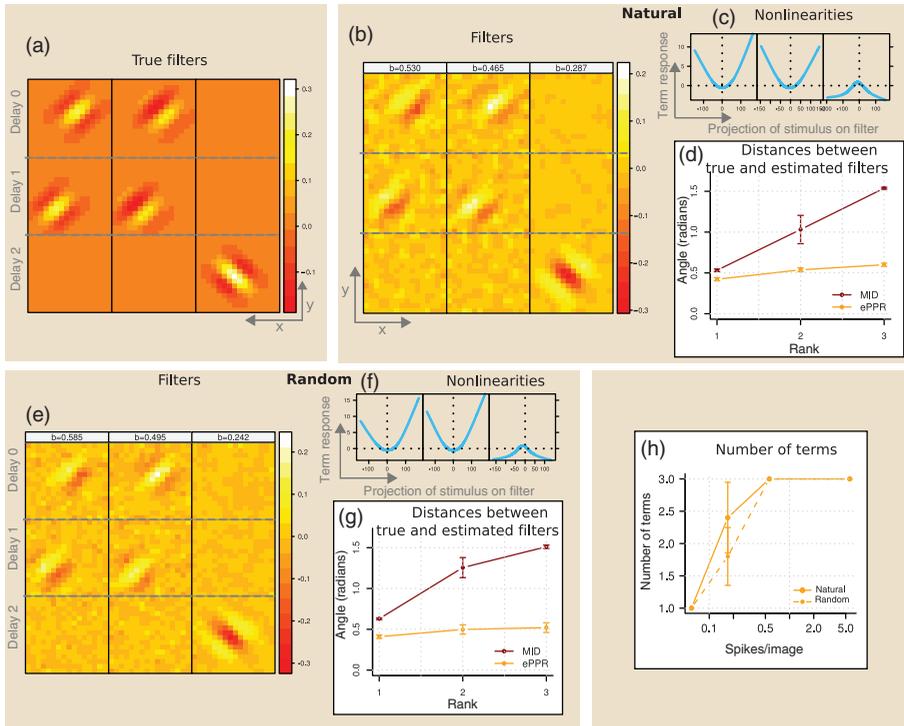


Figure 2. Simulated cell: ePPR models. (a): filters of the simulated model (Equation 8; f1: left, f2: center, f3: right). (b, c): filters (b) and nonlinear functions (c) of the example model estimated from natural data. The titles in (b) are the corresponding β coefficients. (d): principal angles between the true filters and those of ePPR (orange curve) or MID (red curve) filters estimated from responses with the reference noise level. (e–g): as (b–d) but for models estimated from random data. (h) average number of terms in ePPR models estimated from natural and random data. For both example models, the estimated filters are similar to the true filters, and the nonlinear functions correctly indicate the facilitatory/suppressive nature of the corresponding filters.

The ePPR estimation procedure makes no assumption about the statistical properties of the stimuli used to estimate its model parameters. To validate this, we estimated ePPR models from simulated responses to natural and random ensembles. For the real complex and simple cells the natural ensemble (quasi-natural sequence ensemble, Section “Methods: Stimulus ensembles”) approximates the spatial statistics of natural movies, but is temporally uncorrelated. To mimic this, for the simulated cell we used a reshuffled natural movie as the natural ensemble (natural sequence ensemble). To assess the influence of temporal correlations on ePPR estimates, we also estimated ePPR models from responses to a natural movie (natural movie ensemble, and Appendix F).

Nonlinear interactions between pixels of images at different delays are relevant to the simulated model in Equation 8. Accordingly, predictions from ePPR models with time interactions were significantly better than those of ePPR models without time interactions (Figure 11c, Appendix D). In this section we show the parameters of ePPR models with time interactions, and in Figure 11a and b we show the parameters of an example model without time interactions.

All models in this section were estimated using sets of 20,000 responses, and their predictive power was evaluated using a disjoint set of 4,000 responses. The estimation of the filters and nonlinear function in ePPR is entirely nonparametric. The conjunction of this non-parametric estimation with the very large amount of noise in the simulated responses could lead to estimates with large variability. To study this variability, for each level of noise in the simulated responses, we estimated five models from distinct resampled subsets of the training data set (see Section “Methods: Data partitioning”). Figures 2d, g, and h quantify the variability in the estimated parameters. For simplicity, Figures 2b, c, e, f, and 3 show parameters of example models estimated from responses with the reference noise level, 0.56 spikes/image.

Data from natural images

Estimated filters. Figure 2b shows contour plots of the filters from the example ePPR model estimated using natural data with the reference noise level (Section “Methods: Stimulus ensembles”). The title of each contour plot is the corresponding β coefficient. Qualitatively the estimated filters look very similar to the true filters of the simulated model (Figure 2a). To quantify this similarity, we computed the principal angles between true and ePPR filter spaces (Section “Methods: Similarity between two sets of filters”). These principal angles show how much of the three dimensional structure of the true filter space is well approximated by the estimated filter space. If the three principal angles are relatively small, then the estimated filter space well approximates the true filter space along its three dimensions. But, if only the first n principal angle are small, then the estimated filter space is a good approximation of the true filter space only along n dimensions. The orange curve in Figure 2d plots the averaged principal angles between the true and estimated filter spaces. The three principal angles are relatively small, showing that the filter space estimated with ePPR is a good approximation of the true filter space along its three dimensions. Moreover, the size of the error bars is small indicating little variability in the estimated filters.

For comparison, we also estimated five sets of Maximally Informative Dimensions filters (MID; Sharpee et al. 2004, Section “Methods: MID”) from five different jackknifed subsets of the training dataset. Figure 3a shows an example set of MID filters. As in all our MID estimates of the simulated cell (all noise levels and all jackknife estimates), MID failed to recover the inhibitory filter. The red curve in Figure 2d plots the average principal angles between the true and MID filter spaces. The small first principal angle shows that the MID filter space well approximated the true filter space along one dimension. But the larger second and third principal angles indicate that the MID approximations along more than one dimension were poor.

Estimated nonlinear functions. Figure 2c shows the nonlinear functions of the example ePPR model estimated from natural data. These functions map the dot product of a stimulus and the corresponding filter onto the contribution of the term to the prediction of the model. Points with large magnitude in these nonlinear functions are either all positive or all negative. If they are all positive, the filter

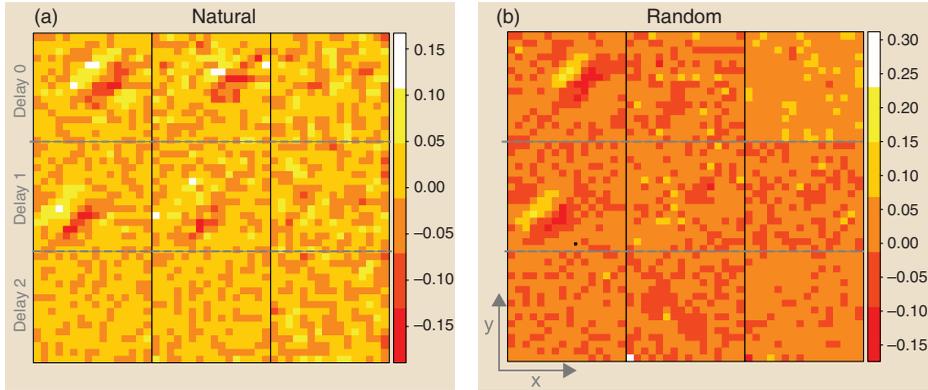


Figure 3. Simulated cell: example set of MID filters estimated from natural data (a) and random data (b). MID well approximated the true filter space along one dimension (see red curves in Figures 2d and g), but approximations along more than one dimension were poor. In particular, in all MID estimates of the simulated cell (all noise levels and all jackknife estimates), MID failed to recover the inhibitory filter.

associated with the nonlinear function will facilitate the response of the cell, and if they are all negative the filter will suppress the response of the cell. Points with large magnitude in the two leftmost nonlinear functions are all positive, while those in the rightmost nonlinear function are all negative. Thus, the two leftmost filters in Figure 2b are facilitatory, while the rightmost filter is suppressive. Since this agrees with the construction of the model cell, the estimated nonlinear functions correctly recovered the facilitatory/suppressive nature of the associated filters.

Predictive power. We compared the predictive power of filters estimated with different methods: ePPR; MID; normalized spike triggered covariance (nSTC; Touryan et al. 2005, Section “Methods: nSTC”), as has been previously employed to characterize complex cells (Touryan et al. 2002; Felsen et al. 2005); and PPR (Appendix A)^{III}. Because some of the above methods do not provide a predictive model, and to make the comparison of the predictive power of the filters independent of the predictive model used by each method, we used a second-order multi-dimensional polynomial (Section “Methods: Polynomial predictive model”) as the predictive model for all the methods. For each number of spikes per image, or noise level, in the simulated responses, and for each type of filter (ePPR, MID, PPR, or nSTC) we constructed a second-order multi-dimensional polynomial, used it to predict responses to the eight testing subsets and computed the Pearson correlation coefficient between each of these predictions and the simulated responses of the cell. Figure 4a plots the mean of these correlation coefficients, with error bars of size two standard deviations, as a function of the number of spikes/image, or the noise level, in the simulated responses. Orange asterisks indicate that correlation coefficients for ePPR filters were significantly larger than those for MID filters (Wilcoxon signed-rank test, $p < 0.01$, Section “Methods: Testing for difference in predictions”). Spatio-temporal filters, ePPR and MID, predicted responses substantially better than spatial filters, PPR and nSTC, and ePPR predicted responses better or at the same level as MID filters. However, the

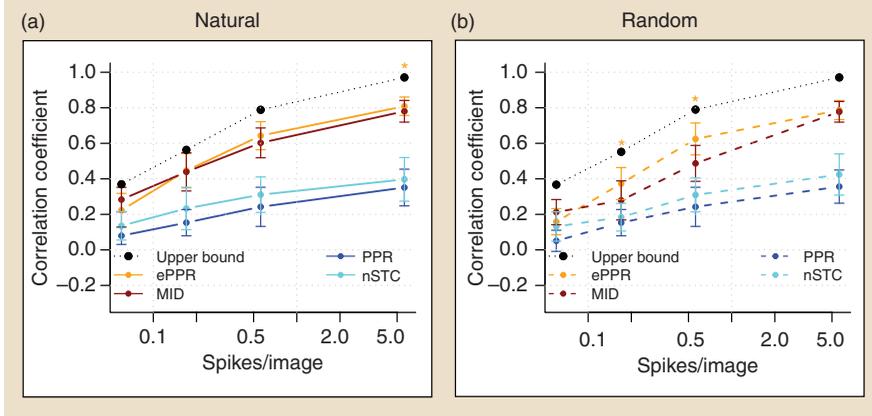


Figure 4. Simulated cell: predictive power of filters estimated with different methods. Pearson correlation coefficient between predictions from polynomial models and responses of the simulated cell, as a function of the mean number of spikes per image, or noise level, in the simulated responses. The orange, red, blue, and cyan curves correspond to polynomials constructed with ePPR, MID, PPR, and nSTC filters, respectively. Black curves: upper bounds on the correlation coefficients. (a): predictions with natural stimuli. (b): predictions with random stimuli. Orange asterisks mark number of spikes/image at which correlation coefficients for ePPR filters were significantly larger than those for MID filters. For both natural and random data, the polynomial models constructed with spatio-temporal filters, ePPR and MID, predict substantially better than those constructed with spatial filters, PPR and nSTC. Also, for all noise levels, ePPR filters yielded better or equal predictions than MID filters. However, the correlation coefficients for the quadratic polynomials is a coarse measure of the quality of the filters (see text).

correlation coefficients for the quadratic polynomials are a coarse measure of the quality of the filters. For example, at the reference noise level, 0.56 spikes/image, the filters estimated with ePPR were more accurate than those estimated with MID (Figures 2a, 2b, 3a, and 2d), but the correlation coefficients for the quadratic polynomials constructed with ePPR filters were not statistically larger than those for the polynomials constructed with MID filters (Figure 4a, orange and red curves at 0.56 spikes/image).

Note that the ePPR model (Equation 6) does not allow nonlinear interactions between the outputs of different filters; i.e., these outputs are combined linearly (after passing through the corresponding nonlinear function, $\phi_{m,d}$ in Equation 6). But the quadratic polynomial used to compare the predictive power of filters in Figure 4, Equation 11, allows nonlinear interactions between the outputs of different filters. Therefore, the predictive power of polynomial models constructed with ePPR filters in Figure 4 could be overestimating the predictive power of ePPR models. However, Figure 11c (Appendix D) shows that this is not the case; at all noise levels ePPR models with time interactions (red curve in Figure 11c) predict better, or at the same level, as polynomial models (orange curve in Figure 11c).

Data from random images

As for natural data, the filters of the example model estimated from random data look similar to the true filters of the simulated model (Figures 2a and e).

This similarity holds for the five ePPR models estimated from the fitting subsets of the training dataset with 20,000 responses (orange curve in Figure 2g). As for natural data, the MID filter space well approximated the true filter space along one dimension, but the approximation along more than one dimension was poor (Figure 3b and red curve in Figure 2g). Also, the nonlinear functions estimated from random data correctly indicated the excitatory/inhibitory nature of the corresponding filters (Figure 2f). Furthermore, second-order multi-dimensional polynomials constructed with ePPR or MID predicted responses substantially better than those constructed with PPR or STC filters (Figure 4b). In addition, polynomial models constructed with ePPR filters predicted responses significantly better than those constructed with MID filters at 0.17 and 0.56 spikes/image.

Figure 2h plots the average number of terms in ePPR models as a function of the number of spikes/image in the simulated responses. At the reference noise level (0.56 spikes/image) and low noise levels (5.62 spikes/image), all ePPR models estimated from natural or random data contained good approximations of the three filters of the simulated cell, at 0.17 spikes/image some models missed one or two of these filters, and at the highest noise level (0.06 spikes/image) all ePPR models contained only one filter. We note that for every noise level ePPR recovered, statistically, the same number of terms from natural and random data, which will become relevant for interpreting the models estimated from real cells (see below).

Due to the divisive inhibition, the simulated complex cell model cannot be represented exactly by an ePPR model. Nevertheless, ePPR provided very good approximations. These approximations remained good for responses simulated with larger amounts of normalization (Appendix E). In addition, the simulated model in Equation 8 contains nonlinear interactions between pixels of images at different delays. As discussed in Section “Extended Projection Pursuit Regression: Spatio-Temporal models”, these interactions cannot be accounted by ePPR models without time interactions. Nevertheless, these models provided good approximations to the true parameters of the simulated model (Figure 11, Appendix D). This shows that ePPR has good generalization properties.

Linear-nonlinear-linear model

The linear-nonlinear-linear model (LNL, Korenberg and Hunter 1986) is a block-structured model that found early applications in the auditory (Weiss 1966) and visual (Weiss 1996; Spekreijse 1969; Spekreijse and Oosting 1970, Spekreijse and Reits 1982) systems. To further test the generality of ePPR, we used it to characterize an LNL extension of the previous complex cell LN model (Equation 8). The mean responses of the LNL model were generated by filtering the mean responses to natural stimuli of the simulated LN model with the reference noise level, with a linear-phase lowpass filter with a cutoff frequency of $\pi/2$ cycles per sample, and with a length of 15 samples. Responses of the LNL model were Poisson random variables with these means. To illustrate the differences between the mean responses of the LN and LNL models, Figure 5a plots their power spectra. We see that the lowpass filter following the nonlinearity in the LNL model has a strong effect on the mean responses of the LN model.

The forward ePPR model (Appendix B) estimated from responses of the LNL model was configured to contain $n[i]$ terms at delay i with $n = [1, 1, 1, 1, 1, 1, 2, 4,$

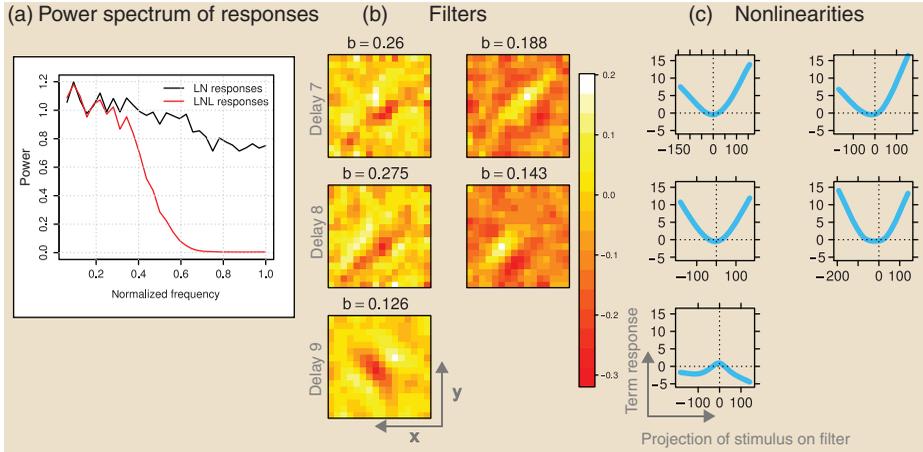


Figure 5. LNL model: (a): power spectra of the responses of the LN (black curve) and LNL (red curve) models. The lowpass filter had a large effect on the responses of the LN model. (b, c): estimated filters (b) and nonlinear functions (c) at delays seven, eight, and nine. The ePPR estimation procedure discarded the irrelevant terms from the forward model and learned the correct model structure for the simulated LNL model, with two facilitatory terms at delay seven, two facilitatory terms at delay eight, and one suppressive term at delay nine. Although the filters for the LNL model are poorer estimates than those for the LN model (Figure 11a, Appendix D), the former filters still are reasonably good approximations of the true filters.

4, 4, 2, 1, 1, 1, 1]. That is, the forward model contained terms at 15 delays. But the ePPR estimation procedure discarded the irrelevant terms, so that the final ePPR estimated model contained only terms at delays seven, eight and nine (Figures 5b and c). Considering that the lowpass filter after the nonlinear function in the LNL model introduces a delay of seven samples, we see that ePPR has learned the correct model structure for this LNL model.

Comparing the filters of the ePPR model without time interactions estimated from responses of the LNL model (Figure 5b) with those estimated from responses of the LN model (Figure 11a, Appendix D), we see that the linear filter following the nonlinearity in the LNL model has a detrimental effect on the quality of the estimated ePPR filters. For example, the true filters at delay zero are positioned in the top-left quadrant (Figure 2a), but the most important estimated filter at delay zero (Figure 5b, top-left panel) is incorrectly positioned at the center. Still, these filters capture important features of the true filters, like their Gabor shape, orientation, and spatial frequency. Also, the nonlinear functions have proper quadratic shapes, and correctly indicate the facilitatory or suppressive nature of the corresponding filters (Figure 5c).

In summary, we have shown that ePPR successfully characterized a simulated complex cell from its responses to natural and random stimuli. ePPR recovered all of the underlying filters and nonlinear functions from the simulated cell, including those that were suppressive. In addition, predictions from ePPR filters were superior to predictions from filters estimated with previous methods. The generality of ePPR was demonstrated by using it to characterize a simulated complex cell with divisive normalization, and an LNL extension of it, neither of which can be represented exactly with an ePPR model.

Complex cell

In the previous section we showed that ePPR was able to characterize a simulated complex cell from its responses to either natural or random data. We now use the method to characterize a real complex cell recorded from an anesthetized cat. Experimental procedures are described elsewhere (Felsen et al. 2005).

A unique characteristic of the data recorded by Felsen et al. (2005) is that individual cells were probed with both natural and “matched” random stimuli (Section “Methods: Overview of experimental paradigm”). Below we first characterize the complex cell from its responses to the natural stimuli, and then from its responses to the matched random stimuli.

To determine the memory of the cell, we first estimated ePPR models without time interactions. Then, because the memory was short, and the amount of data large, we estimated models with time interactions. In this section we present models with time interactions, and Figure 12 (Appendix D) shows a model without time interactions estimated from responses to natural stimuli.

To assess the dependence of ePPR estimates on the size of the training dataset, we estimated models using sets of 3,000, 10,000, and 20,000 responses. For each of these sets, five ePPR models were estimated from different fitting subsets. For simplicity, we show the parameters of example models estimated from a fitting subset of the training dataset with 20,000 responses, but the number of terms (Figure 6e) and correlation coefficients plots (Figures 6f and g) show averages across the five estimated models.

Data from natural images

We recorded responses to four repeats of the quasi-natural sequence ensemble. The mean total number of spikes in these four sets of responses was 87,258. We used the mean of these sets of responses to estimate the parameters of the different models.

The filters of the example ePPR model estimated from natural data (Figure 6a) are consistent with previous estimations of linear subspaces of complex cells from responses to two-dimensional natural (Touryan et al. 2005; Rapela et al. 2006), or random (Movshon et al. 1978a; Chen et al. 2007) images. In particular, the three middle filters have clear Gabor shapes, with similar orientation and spatial frequency, but are shifted in phase. However, note that the bottom frame of the rightmost filter (operating on the image presented between 85 and 126 ms prior to the response of the cell) is cross-oriented with respect to the other filters.

For comparison, we attempted to estimate sets of five MID filters using the training dataset with 20,000 responses to natural stimuli. However, to estimate five filters, the implementation of MID used here required the allocation in memory of two vectors of size 556 megabytes, which was not possible with our computers. So we estimated sets of four MID filters. An example set is shown in Figure 7a. Only the leftmost and rightmost filters in Figure 7a are well structured, and these filters are very similar to the two most relevant ePPR filters, according to the β coefficient (two leftmost filters in Figure 6a). That ePPR recovered more filters with good structure than MID could be explained as a shortcoming of ePPR in estimating

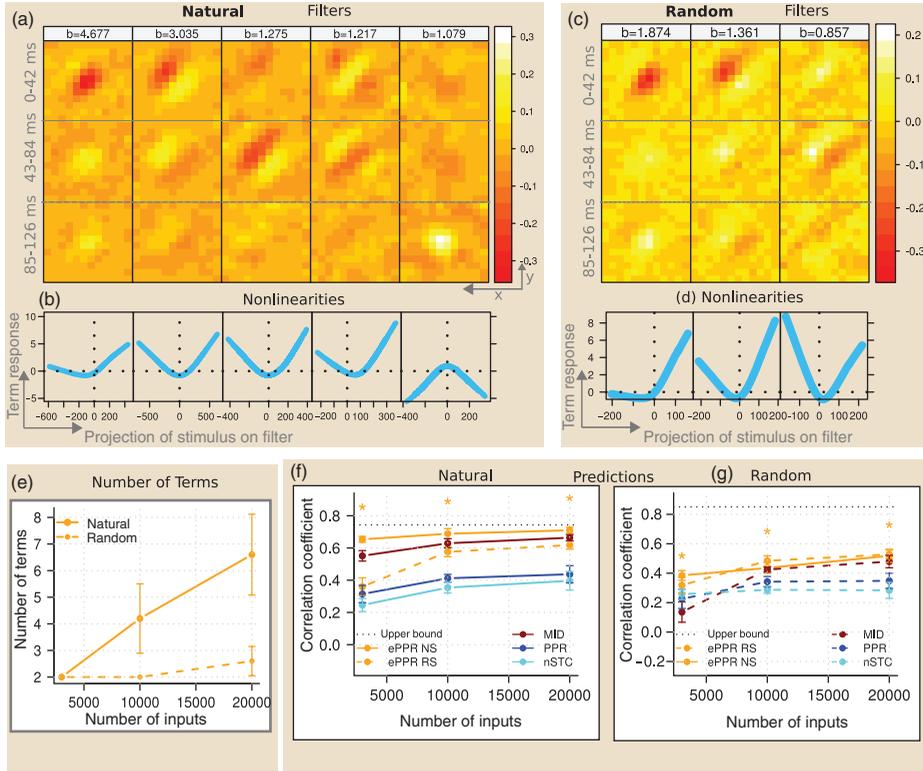


Figure 6. Complex cell: ePPR models. (a, b): filters (a) and nonlinear functions (b) of the example model estimated from natural data. The titles in (a) are the corresponding β coefficients. (c, d): as (a, b) but for models estimated from random data. (e): average number of terms in ePPR models estimated from natural and random data. (f–g) predictive power of filters estimated with different methods with same format as in Figure 4, but in (f) the dashed orange curve shows the predictions of ePPR filters estimated from random data to natural stimuli, and in (g) the solid orange curve shows the predictions of ePPR filters estimated from natural data to random stimuli. The estimated filters and nonlinear functions are consistent with those estimated using previous methods. Models estimated from natural and random data are similar to each other. However, late suppression is only present in the model estimated from natural data. Furthermore, models estimated from natural data recovered more filters than models estimated from random data. For natural and random data, ePPR filters yielded predictions substantially better than filters estimated with other methods, and for natural data, predictions from ePPR filters were close to the upper bound on the predictive power of any model.

spurious filters, or as a shortcoming of MID in failing to recover relevant filters. We will return to this point below.

Figure 6b shows the nonlinear functions of the example model estimated from natural data. The leftmost nonlinear function is approximately a half-wave rectification, the three middle ones are full-squared, and the rightmost nonlinear function is suppressive. The full-squared nonlinearities are in agreement with the polarity invariance of complex cells (Movshon et al. 1978a). That complex cells can be characterized with half-wave and full-squared nonlinear functions has been previously reported (Rust et al. 2005, Figure 5). And the rightmost nonlinear

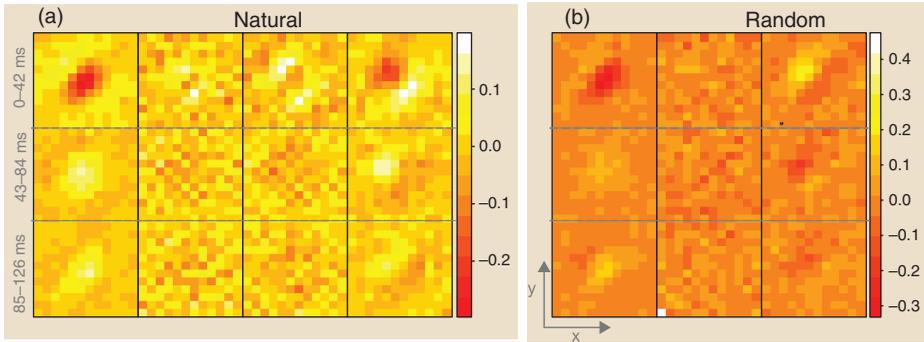


Figure 7. Complex cell: example set of MID filters estimated from 20,000 responses. Same format as in Figure 3. For both natural and random data only the leftmost and rightmost filters are well structured, and are similar to the most important filters estimated with ePPR (Figures 6a and c). The predictive power of the filters estimated with MID is significantly worse than that of the filters estimated with ePPR (Figures 6f and g)

function, corresponding to the filter with a cross-oriented frame at a late delay, reveals cross-oriented inhibition in the response of this complex cell.

As for the simulated cell, we compared the predictive power of filters estimated with ePPR, MID, nSTC, and PPR, using a second-order multi-dimensional polynomial as the predictive method. Figure 6f plots the correlation coefficients between responses of the complex cell and predictions of the polynomial models, as a function of the number of stimuli used to estimate the filters and polynomial models. The dotted line is an upper bound on these correlation coefficients (Section “Methods: Upper bound on predictions”). For all number of stimuli best predictions are obtained with ePPR filters. This indicates that the filters with good structure estimated with ePPR, and not with MID (cf. Figures 6a and 7a), are relevant to predict the responses of the complex cell, and therefore are not spurious. Moreover, ePPR filters estimated using 20,000 inputs closely approximate the upper bound on the correlation coefficients.

Data from random images

We recorded responses to two repeats of the quasi-random sequence ensemble. The mean total number of spikes in these responses was 73,022.5. We used the mean of these sets of responses to estimate parameters of the different models.

The filters obtained with random data (Figure 6c) match well the three most important filters, according to their β coefficient, obtained with natural data (Figure 6a), with the third filters estimated from natural and random data being reversed in polarity. For comparison, Figure 7b shows an example set of three MID filters estimated from 20,000 responses to random stimuli. As for the example set estimated from natural data (Figure 7a), MID recovered only two filters with good structure. Predictions obtained with these filters were significantly worse than those obtained with ePPR filters (Figure 6g, red versus orange dashed curves).

Two important differences emerged between the models estimated from natural and random data: First, a filter with late suppression was recovered from natural, but not from random, data. This difference was not an idiosyncrasy of the example models shown, but was present in most ePPR models estimated with more than 3,000 responses (all five models estimated from 20,000 natural responses, three of the five models estimated from 10,000 natural responses, but no model estimated from random responses, showed late suppression). Second: models estimated from natural data had more filters than those estimated from random data (Figure 6e).

The previous differences between models estimated from natural and random data are functionally relevant. The filters of the former models predict responses to natural stimuli substantially better than the filters of the latter models (Figure 6f, solid versus dashed orange curves). This observation is not due to overfitting, because the predictive power of filters was assessed with data not used for their estimation. One could argue that the filters of the models estimated from natural data were better than those of the models estimated from random data. If this were the case, the former filters should predict responses to random stimuli better than the latter filters. However, this is not the case; ePPR filters estimated from natural data predict responses to random stimuli at the same level as ePPR filters estimated from random data (Figure 6g, solid versus dashed orange line).

In summary, in this section we demonstrated the feasibility of ePPR to characterize a complex cell from its responses to natural and random stimuli. We showed that estimated ePPR models had several features in common with the energy model of complex cells, that ePPR models estimated from natural and random stimuli were very similar to each other, but displayed a few interesting differences. In addition, we showed that predictions from ePPR models estimated with natural data were close to an upper bound on the predictive power of any model.

Simple cell

The ePPR model is general and one can use it to characterize a wide variety of visual cells. Here, we test this generality by using ePPR to model a cortical simple cell (Felsen et al. 2005) from its responses to both natural and random stimuli. We follow the same procedure as in the previous section.

For natural stimuli, we recorded responses to two repeats, obtaining two sets of responses with a mean total number of spikes of 8,753. For random stimuli we also recorded responses to two repeats, obtaining two sets of responses with a mean total number of spikes of 6,673.5.

Figure 8 shows that the estimated filters and nonlinear functions follow the same pattern as for the complex cell (Figure 6). Natural and random stimuli yield similar filters and nonlinear functions. Consistently with earlier data, the filters are Gabor-like and oriented, and their nonlinear functions are half-wave rectifications (Movshon et al. 1978b). However, two important differences are evident: First, as for the complex cells, explaining responses to natural stimuli requires more filters (c.f. Figures 8a and c, see also Figure 8e). Second, the model estimated from random data, with an estimated filter that has frames with similar shape at all delays,

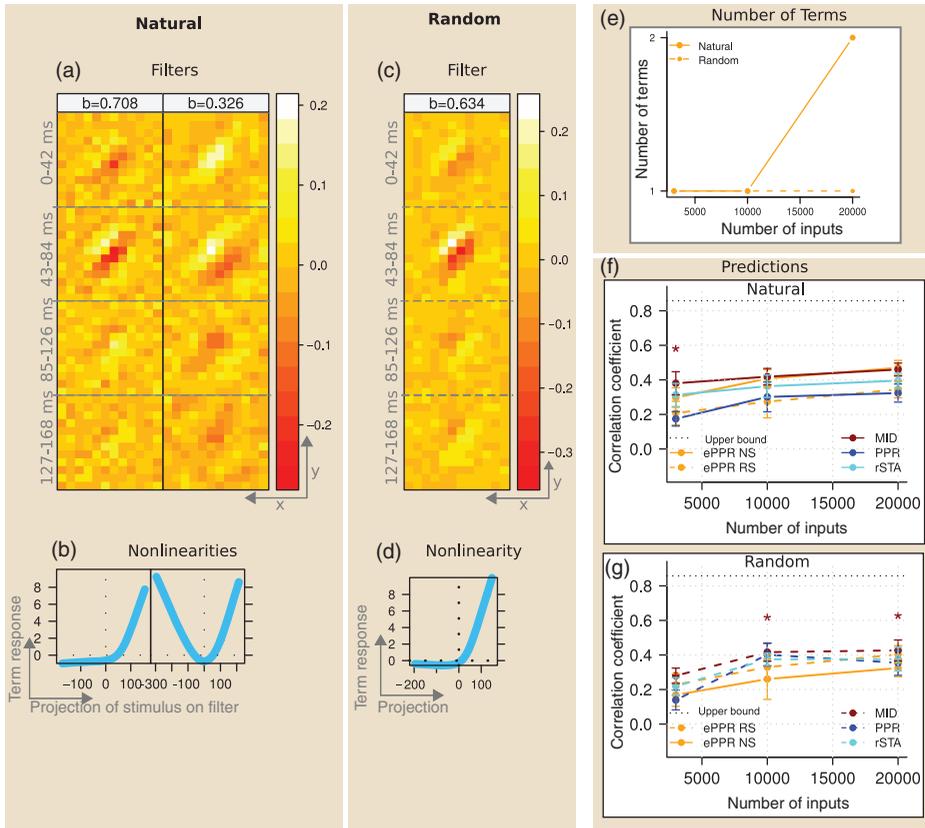


Figure 8. Simple cell: ePPR models. The format of this figure is identical to that of Figure 6, but in (f, g) asterisks mark number of inputs at which correlation coefficients for MID filters are significantly better than those for ePPR filters, and cyan curves correspond to rSTA filters. The figure in (e) does not contain error bars because, for each number of inputs, all estimated models had the same number of terms. The estimated filters and nonlinear functions are consistent with those estimated using previous methods. Models estimated with natural and random data are similar to each other. However, the model estimated from natural data, but not that estimated from random data, has features of a complex cell model. Furthermore, models estimated from natural data contain more filters than those estimated from random data. For natural stimuli, predictions from ePPR and MID filters are similar, and better than those from rSTA and PPR. For random stimuli, predictions from MID filters are slightly, but significantly, better than those from ePPR filters.

but whose amplitudes are modulated in time (Figure 8c), and with a half-wave rectification (Figure 8d), resembles the classical space-time separable model for simple cells (DeAngelis et al. 1993a, 1993b). In contrast, the model estimated from natural data has mixed features of simple and complex cells. For this model, the most important term according to the β coefficient (left column in Figure 8a and b) is also typical of a simple cell. However, the least important term (right column in Figure 8a and b) is consistent with the energy model for complex cells (Adelson and Bergen 1985): the second filter has a similar shape as the first filter, but it is shifted in phase, and its nonlinear function is fully rectified. Therefore, if we look only at the most important term of the model estimated from natural data, we see a typical

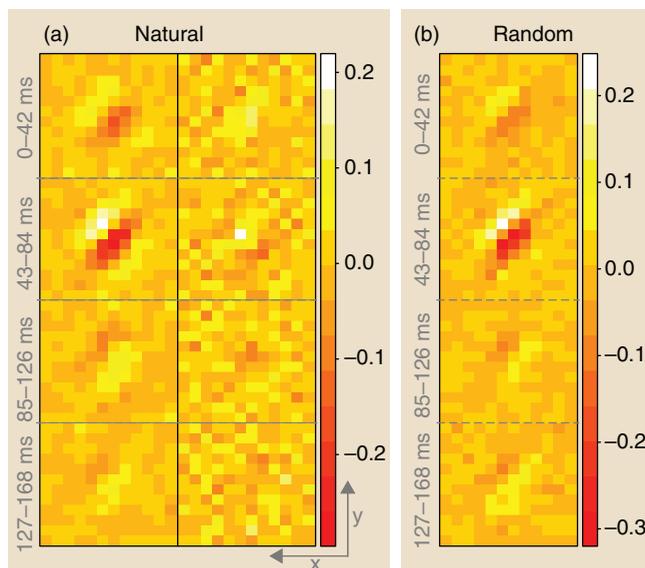


Figure 9. Simple cell: example set of MID filters estimated from 20,000 responses. Same format as in Figure 3. These filters are very similar to those estimated with ePPR, c.f., Figure 8a and c.

space-time separable simple cell. But, if we look at all the terms, we see a mixture of a simple and a complex cell. These differences are functionally significant. Filters estimated from natural (random) data predict responses to natural (random) stimuli significantly better than filters estimated from random (natural) data; as has been previously observed for simple cells (Sharpee et al. 2006), and for cells in the auditory midbrain region (Woolley et al. 2006). As opposed to the case for the complex cell, the ePPR and MID filters are very similar to each other (c.f., Figure 8a with 9a and Figure 8c with 9b).

For natural data, third-order polynomials generated similar predictions from ePPR and with MID filters (Figure 8f)^{IV}. Also, for 20,000 and 10,000 natural stimuli, predictions from ePPR filters were significantly better than those from regularized spike triggered average (rSTA; Smyth et al. 2003, Section “Methods: rSTA”) (one-sided Wilcoxon signed-rank test $p < 0.01$, Figure 8f), and for all number of natural stimuli ePPR filters yielded better predictions than PPR^V filters (one-sided Wilcoxon signed-rank test $p < 0.01$). For random stimuli all four models predicted similarly (Figure 8g). However, predictions derived from MID filters estimated from 10,000 and 20,000 random stimuli were slightly, but significantly (one-sided Wilcoxon signed-rank test $p < 0.01$), better than those derived from ePPR filters.

In summary, in this section we demonstrated the feasibility ePPR to characterize a simple cell from its responses to natural and random stimuli. We showed that estimated ePPR models had several features in common with the standard space-time separable model for simple cells. Also, ePPR models estimated from natural and random data were very consistent with each other but, as for the complex cell, had a few interesting differences.

Discussion

The curse of dimensionality (Bellman 1961) is an important problem in the characterization of visual cells from input/output data. For this reason, LN models, like STA and STC, have gained substantial popularity. However, recently it has been shown that the number of filters in LN models of visual cells is large (Rust et al. 2005; Chen et al. 2007). So now this curse affects the estimation of LN models of visual cells. The estimation of these models is further complicated when using natural stimuli as inputs. Consequently, up to now, no method has been able to estimate an LN model of visual cells, containing multiple spatio-temporal filters, from their responses to natural stimuli. This article introduced a method that makes this estimation possible, and demonstrated its good performance with simulated and physiological data from the primary visual cortex.

The ePPR model can approximate a broad class of cells. We proved that ePPR models can uniformly approximate any continuous function to an arbitrary degree of precision. We validated this generality empirically. First, due to the divisive normalization, the model of the simulated complex cell (Equation 8) cannot be exactly represented by an ePPR model (Equation 6). Nevertheless, ePPR approximations were very good (Figure 2), and these approximations remained good for different strengths of divisive normalizations (Appendix E). Second, interactions between pixels of images at different delays are relevant to the responses of the simulated complex cell (Equation 8). These interactions are neglected by ePPR models without time interactions. In spite of this, ePPR models without time interactions provided good approximations (Figure 11, Appendix D). Also, interactions between pixels of images at different delays were relevant for the cortical complex cell, as evidenced by the substantially better predictions of the ePPR model with time interactions over those of the ePPR model without time interactions (orange versus pink curves in Figure 12c, Appendix D). For this cell ePPR models without time interactions were very similar to the ePPR models with time interactions. And third, ePPR provided a reasonable approximation to a linear-nonlinear-linear extension of the simulated complex cell model, Figure 5.

Since ePPR makes no assumptions about the statistical properties of its inputs, it is well-suited to characterize visual cells from their responses to arbitrary, including natural, inputs. We tested this with simulated and physiological data. For the simulated cell we showed that ePPR models estimated from natural and random data well matched the simulated cell model (Figure 2). And for the cortical complex and simple cells, we showed that models estimated from natural data were very consistent with those estimated from random data, and with previous characterizations of these cells (Figures 6 and 8).

Because ePPR is an entirely non-parametric algorithm, and because the responses of the cells were very noisy, ePPR estimates could have been very variable. However, this was not the case (Figures 2d, 2g, 2h, 6e, and 8e). Furthermore, ePPR models estimated from natural and random data, i.e. with stimuli and responses with very different statistics, were very similar to each other, for the simulated cell (Figure 2), for the complex cell (Figure 6), and for the simple cell (Figure 8). Several features of the ePPR estimation procedure help reduce the variability of the estimated parameters. First, the projection pursuit strategy used by ePPR reduces the original

large-dimensional problem of fitting simultaneously all the parameters in the ePPR model, Equation 6, to a sequence of fittings and refittings of only one term in Equation 6 at a time. Second, the optimization criterion for the filters, α in Equation 7, penalizes non-smooth filters. And third, the estimation of the nonlinear functions, ϕ in Equation 7, is performed using smoothing splines with a relatively large penalty for non-smooth estimates.

Parameters of estimated ePPR models displayed properties previously reported in the literature. For both the simple and complex cells, estimated filters had oriented Gabor-like shapes with similar spatial frequencies and orientations (Movshon et al. 1978b, 1978a). For the simple cell, different frames of estimated filters had a very similar shape, but their amplitude was modulated in time, in agreement with the space-time separable model for simple cells (DeAngelis et al. 1993a, 1993b). For the complex cell, inhibition appeared late in time, as previously reported (Rust et al. 2005; David et al. 2004). Most nonlinear functions were full-squared rectifications for the complex cell (Figures 6b and d), while they were half-wave rectifications for the simple cell (Figures 8b and d).

Although the focus of this article is methodological, the results presented here suggest that the response properties of cortical cells may depend on the statistics of the stimuli used to probe them. Recently, Sharpee et al. (2008) showed that spatio-temporal LN models of simple cells, estimated from natural and random stimuli, displayed significant differences. However, these LN models contained only one filter. Here, we showed, for the first time, differences in spatio-temporal LN models, with multiple filters, estimated from natural and random stimuli. We found that ePPR recovered more filters from natural than from random data, for the complex cell (Figure 6e), for the simple cell (Figure 8e), but not for the simulated cell (Figure 2h). Also, ePPR recovered inhibitory terms at later delays in the models of the complex cell estimated from natural, but not from random, data, consistently with the findings of David et al. (2004). Furthermore, the simple cell models estimated from natural responses, but not those estimated from random responses, displayed properties typical of complex cells, supporting the notion by Mechler and Ringach (2002) that simple and complex cells are not two different classes of cells but lie on a continuum. This study thus provides further support for the notion that the observable response properties of sensory cells depend on the statistics of the stimuli used to probe them (Theunissen et al. 2000; David et al. 2004; Felsen et al. 2005; Woolley et al. 2005; Woolley et al. 2006; Sharpee et al. 2006, 2008; David et al. 2009).

However, the previous observations were derived from only two cells and we caution the reader against concluding that they apply to all V1 cells. A forthcoming article will focus on how these observations generalize to a larger population of cells in the primary visual cortex. In addition, differences in parameters of models estimated from responses of cells to stimuli with different statistics could be artifacts of using overly-constrained models (Christianson et al. 2008). But this does not appear to be the case for ePPR and the cells characterized here. Because the ePPR model with time interactions can uniformly approximate, to an arbitrary degree of precision, any continuous function with inputs in the unit cube, it is not overly-constrained to represent a large class of cells. Also, for the simulated cell no large differences were observed in ePPR models estimated from natural and random data (Figure 2).

From the cells we analyzed in the dataset from Felsen et al. (2005), the cortical complex cell shown here is the one for which we obtained the best correlation coefficient. However, similar correlation coefficients, and qualitatively similar ePPR estimates, were obtained for other complex cells (Figure 15, Appendix G).

We compared ePPR with state of the art methods for the estimation of linear-nonlinear models, and ePPR performed better, or at the same level. The method with results most similar to ePPR was MID. Currently MID is the only alternative to ePPR for the estimation of multiple spatio-temporal filters from responses of visual cells to natural stimuli. MID is an elegant algorithm that performs well for the characterization of visual cells whose response depends on one or two filters, e.g., the simple cell. However, MID is affected by the curse of dimensionality and has limitations for the characterization of visual cells whose response depends on several filters. For these cells, i.e., the simulated and cortical complex cells, ePPR performed substantially better than MID. In addition, the ePPR algorithm has advantages over MID in terms of both computational time and memory requirements.

ePPR uncovered more structure than MID in visual cells whose response depended on multiple filters. For the simulated complex cell, MID failed to recover the inhibitory filters with natural (Figure 3a) and random (Figure 3b) stimuli, while ePPR recovered all filters with both types of stimuli (Figures 2b and e). This failure of MID was not limited to the noise level used to estimate the example models, but occurred even at the lowest noise levels. It should be emphasized that there is a real danger in using methods that do not converge to the correct model with increasing signal-to-noise ratios. Also, ePPR better approximated the true filters of the simulated cell than MID, with natural (Figure 2d) and random (Figure 2g) stimuli. From responses of the cortical complex cell to natural stimuli ePPR recovered five filters with good structure (Figure 6a) while MID recovered only two (Figure 7a), and the predictions of ePPR models were significantly better than those of MID models (Figure 6f). And, from responses of the cortical complex cell to random stimuli, ePPR recovered three filters with good structure (Figure 6c), while MID recovered only two (Figure 7b), and again the predictions of ePPR filters were significantly better than those of MID filters (Figure 6g).

The MID algorithm is affected by the curse of dimensionality. To compute the mutual information objective function (Equation 9) for n filters, MID requires the estimation of two n -dimensional probability distributions, and the amount of data required for these estimations grows exponentially with n . For example, the implementation of MID used here estimates these probability distributions using histograms. Then, for $n = 5$ filters, if the projection of an image on a filter is discretized using $N_{\text{bins}} = 15$ bins, an extraordinary large amount of data is needed to fill a histogram with $N_{\text{bins}}^n = 759,375$ bins, so that it well approximates the underlying probability distribution.

The implementation of MID used in this article is very demanding in terms of computational memory and time requirements. Regarding memory, the large requirements of MID did not allow our computers to estimate more than four MID filters (we described this problem on page 15). Regarding time, in order to estimate a set of n filters, the implementation of MID used here requires $n+1$ maximizations;

n one-dimensional maximizations ($n=1$ in Equation 9) to estimate initial conditions for the n filters, and one n -dimensional maximization to jointly estimate all the filters, using the previously computed initial conditions. For instance, on a personal computer with a 3 GHz processor and 3 GB of memory, the estimation of the four MID filters of the complex cell in Figure 7a required approximately eight hours, which is much longer than the 45 minutes required by ePPR to estimate the five filters in Figure 6a.

The greedy estimation of the forward model by PPR (Appendix A) is not guaranteed to converge to the global optimum. For this reason PPR uses the backward procedure to discard spurious terms from the forward model. However, for responses with large amounts of noise, or models estimated with small data sets, the backward procedure does not remove all spurious terms from the forward model. So, in ePPR (Appendix ePPR) we use a model selection procedure to select the best model from the collection of models returned by the backward procedure. Still, the backward and model selection procedures are heuristic and do not guarantee convergence to the global optimum. To study how frequently ePPR returned suboptimal solutions, for the simulated and cortical cells studied here we estimated five ePPR models from different subsets of the training dataset. For data sets with the reference or the lowest noise level, models of the simulated cell were all very good approximations. They contained the correct number of filters (Figure 2h), which were very similar to the true filters and had small variability (Figures 2b, d, e, and g), and with nonlinear functions with correct shape and sign (Figures 2c and f). For large data sets, models of the complex cell estimated from natural data approached an upper bound on the predictive power of any model (Figure 6f). So, for models estimated from good quality data, the convergence to suboptimal solutions does not seem to be a severe problem. ePPR models estimated from datasets with large amounts of noise, or with small or intermediate sizes, recovered fewer terms than the optimal (Figures 2h, 6e, and 8e) but, thanks to a stringent model selection procedure (Section “Methods: ePPR model selection procedure”), did not contain spurious terms.

To estimate spatio-temporal models one can use ePPR models with or without time interactions. Models with time interactions are more general than models without time interactions, but require estimating larger spatio-temporal filters (and therefore more training data), and knowing the memory of the cell in advance. In contrast, models without time interactions contain smaller spatial filters, and the estimation algorithm discovers the memory of the cell in only one run. For the cortical complex and simple cells studied here we first estimated an ePPR model without time interactions, in order to determine the memory of the cell (constant D in Equation 6). Then, because the estimated memory was short ($D=3$ for the complex cell and $D=4$ for the simple cell) and these cells had been probed with large stimuli sets, we were able to estimate ePPR models with time interactions. In most cases ePPR models with time interactions performed better than models without time interactions (Figures 11c and 12c, Appendix D), but for the LNL cell a single estimation of an ePPR model without time interactions revealed that the cell had a memory of $D=9$ previous frames, which is not feasible for an ePPR model with time interactions.

The model used in ePPR is that of an artificial neural network with three layers (input layer, hidden layer, and output layer). Thus, ePPR is related to the neural network method proposed by Prenger et al. (2004). Both methods use non-parametric models that can characterize a large variety of visual cells. Also, because both are regression-based methods, they can estimate their parameters using natural stimuli. However, because the neural network method estimates all its parameters simultaneously, it does not overcome the curse of dimensionality. A detailed comparison between projection pursuit and neural network methods is given in Hwang et al. (1994). To bypass the curse of dimensionality Prenger et al. (2004) used projections of the input images in a few principal components as inputs to the neural network. But, important information about the inputs could be lost in these low-dimensional projections. The strategy used by ePPR to bypass this problem is a more general one.

Below we summarize some advantages and disadvantages of the ePPR algorithm.

Advantages

- (1) *Generality.* As discussed above, the ePPR model is very general and can approximate many classes of cells.
- (2) *Finding the model structure in only one estimation.* Provided the number of terms per delay, M_d^L , and the number of delays, D^L , of the forward model are sufficiently large, only one estimation of an ePPR model without time interactions finds the model structure; i.e., number of delays and number of terms for each delay required to characterize a given cell. This feature is nicely illustrated by the ePPR model estimated for the LNL simulated cell.
- (3) *Natural stimuli.* As discussed above, ePPR models can be estimated using natural stimuli. This is an advantage over previous methods that require Gaussian stimuli (Chichilnisky 2001; Rust et al. 2005).
- (4) *Temporal dimension.* Previous methods (Touryan et al. 2005; Rapela et al. 2006; Chen et al. 2007) used only one image in the past to predict the current response, while ePPR uses several past images. As shown in Figures 4, 6f, and 6g, spatio-temporal ePPR models predicted responses substantially better than purely spatial PPR and nSTC models.
- (5) *Spatially 2D model.* Rust et al. (2005) estimated a spatio-temporal LN model of visual cells. However, to reduce the number of parameters in their estimations, the input images varied along a single spatial dimension (optimally-oriented bars). Due to its efficient optimization algorithm, and to the smooth prior for its filters, ePPR allows the use of two-dimensional images.
- (6) *Multiple filters.* We have shown that for the simulated cell (Figure 2), the complex cell (Figure 6), and the simple cell (Figure 8), ePPR recovered more than one filter. This contrasts with STA, which can only estimate one filter, and with MID, which, as shown here, has limitations for the estimation of multiple filters.
- (7) *Suppressive filters.* ePPR successfully recovered the suppressive filter of the simulated cell (Figure 2), and a suppressive filter from responses of the cortical complex cell to natural stimuli (Figure 6). To our knowledge,

ePPR is the first non-parametric algorithm that has been shown to estimate suppressive filters from responses of visual cells to natural stimuli.

- (8) *Improved predictions.* ePPR provided better predictions of the responses of the simulated cell (Figure 4), the complex cell (Figures 6f, and g), and the simple cell (Figure 8f) than previous methods.

Disadvantages

- (1) *Spurious terms.* The analysis of projection pursuit techniques by Huber (1985) concludes with a nice description of this limitation:

Perhaps the practical conclusion to be drawn is that we shall have to acquiesce to the fact that Projection Pursuit will in practice uncover not only true but also spurious structure, and that we must weed out the latter by other methods, for example by validating the results on different data sets.

To weed out the spurious structure from ePPR models we used a model selection procedure based on cross-validation (Section “Methods: ePPR model selection procedure”). With data sets large enough to perform reliable cross-validations, our model selection procedure worked very well. However, large recordings are not very common in visual neurophysiology. So we are currently investigating Bayesian model selection procedures to perform model selection without the need for cross-validation data.

- (2) *Hyper-parameters.* ePPR contains many hyper-parameters. Although we used reasonable heuristics to select their values (Section “Methods: Selection of ePPR hyperparameters”), we would like to learn them from data, as done in Roosen and Hastie (1994) with the degrees of freedom of the splines.
- (3) *Nonlinearities.* The ePPR estimation algorithm avoids the curse of dimensionality by working with one-dimensional projections. The cost is that it is poorly suited to characterize highly nonlinear functions (Huber 1985).
- (4) *Global convergence.* As discussed above, ePPR is not guaranteed to converge to the global optimum. However, this is a problem shared by all non-parametric nonlinear optimization techniques.
- (5) *Speed.* The ePPR estimation algorithm is iterative, requiring several fits and refits of models with different numbers of terms, and is thus very slow. On a personal computer, with a 3 GHz processor and 3 GB of memory, estimating the five ePPR filters in Figure 6a required approximately 45 minutes. The low speed of ePPR contrasts with the high speed of the spike-triggered techniques, nSTC and rSTA, which require only seconds. However, ePPR is fast compared to MID, which required approximately eight hours to estimate the four filters in Figure 7a.

In conclusion, this article has demonstrated the feasibility of ePPR, a very general method for the spatio-temporal characterization of visual cells from arbitrary (including natural) stimuli, and showed that ePPR compared favorably with information-theoretic and spike-triggered techniques.

Methods

Overview of the experimental paradigm

The cortical complex and simple cells analyzed here were subjected to the following sequence of inter-dependent experiments, as described in Felsen et al. (2005).

- (1) Responses from a cell were recorded to a quasi-natural sequence ensemble (see next subsection).
- (2) The filter(s) of the cell were estimated from the responses to natural images recorded in Step 1, using nSTC for complex cells, and rSTA for simple cells.
- (3) An ensemble of random images, matched to the ensemble of natural images used in Step 1, was constructed (see next subsection).
- (4) Finally, responses from the cell in Step 1 were recorded to several interleaved repeats of the random and a novel natural ensemble.

Stimulus ensembles

The following stimulus ensembles were used to characterize the simulated and cortical cells.

Natural movie: The natural movie was recorded from a CCD camera attached to the head of cats as they explored outdoor environments (Betsch et al. 2004). We used the copy of this movie provided in the data set of Dr. Tim Blanche, downloadable from Collaborative Research in Computational Neuroscience (<http://crcns.org/data-sets/pvc/pvc-3>). This movie contains 6,000 frames of size 64×64 pixels. To build a movie with 24,000 frames with frames of size 16×16 pixels, we extracted four movies with patches of size 32×32 pixels from the four quadrants of the original movie, we concatenated these four movies, and downsampled each frame by a factor of two.

Natural sequence: This ensemble consists of the center patches (16×16 pixels) extracted from a digitized natural movie (van Hateren and Ruderman 1998), where the frames in the movie have been randomly resorted. Thus, it contains the spatial, but not the temporal, correlation in natural movies. This ensemble was used as the natural stimuli to probe all simulated cells, with the exception of that in Appendix F.

Random sequence: Each frame of this ensemble was matched to the corresponding frame in the natural sequence ensemble, such that (a) the matched frames had the same mean and root-mean-square contrast, and (b) the dot products of each true filter of the simulated the cell onto the matched frames were equal. Random images contain no spatial structure and are visually indistinguishable from white noise. However, due to the matching procedure, they are not white noise. Further details of the matching procedure are provided in Felsen et al. (2005). This ensemble was the random stimuli used to probe all simulated cells.

Quasi-natural sequence: Raw images were selected at random from a database consisting of a variety of digitized natural movies (van Hateren and Ruderman 1998), and the center patch (12×12 pixels) of each image was retained. These

patches were normalized to have the same root-mean-square contrast and, to maximize their diversity, one patch of each a pair of very similar patches was excluded from the ensemble. Further details are given in Felsen et al. (2005). This was the natural ensemble used to probe the cortical complex and simple cells.

Quasi-random sequence: This ensemble was used to probe with random stimuli the cortical complex and simple cells. It was constructed in the same way as the random sequence ensemble, but with two differences. First, the random frames were matched to the quasi-natural sequence ensemble. Second, because for the cortical cells the true filters are not available, the matching was done using filters estimated with nSTC, for complex cells, and with rSTA, for simple cells.

Simulated responses

To study the dependence of the different methods on the noise level of the simulated responses, we generated four sets of responses with different noise levels. We set the constant ω so that the mean of the denominator in Equation 8 was 4.26; i.e., on average the denominator reduced the un-normalized responses by more than four times. In Equation 8, the constant γ controls the mean, and noise level, of the stimulated responses. We set γ so that the mean of \bar{y} in Equation 8 was $E[\bar{y}] = 5.62, 0.56, 0.17,$ and 0.06 spikes/image. The example models in Figures 2 and 3 were estimated from responses with a reference noise level such that $E[\bar{y}] = 0.56$ spikes/image. At this noise level the mean correlation coefficient between simulated responses and predictions from an ePPR model without time interactions was similar to that between responses to natural stimuli of the cortical complex and predictions from an ePPR model without time interactions (0.58 for the simulated cell versus 0.61 for the complex cell). Simulated responses to random stimuli were generated using the same parameters as for responses to natural stimuli.

We simulated an LNL model by lowpass filtering the mean responses of the simulated model (Equation 8) to natural stimuli. We used a linear-phase lowpass filter with a cutoff frequency of $\pi/2$ cycles per sample and with a length of 15 samples. The simulated model used parameters to generate responses with reference noise level; i.e., $E[\bar{y}] = 0.56$ spikes/image. The output of the lowpass filter at time n was the mean of a Poisson random variable giving the number of spikes at time n . These simulated responses were used to estimate the models in Figure 5.

To assess the dependence of ePPR estimates on the amount of divisive inhibition (Figure 13, Appendix E), we fixed the value of the γ parameter, to the value used above to generate responses with the reference noise level, and varied the inhibitory constant ω so that the mean of the denominator in Equation 8 was 4.26, 20.58, and 40.17.

To examine the effect of temporal correlations on ePPR estimates (Figure 14, Appendix F) we simulated responses to the natural movie ensemble. We adjusted the inhibitory constant ω so that the mean of the denominator of Equation 8 was 4.26, and we then set the constant γ so that the mean of \bar{y} in Equation 8 was $E[\bar{y}] = 0.56$ spikes/image, as for the temporally uncorrelated responses used to estimate the example natural model in Figure 2.

Data partitioning

For the simulated and cortical cells, the complete dataset contained 24,000 responses. We generated training datasets of different sizes: 20,000, 10,000, and 3,000 responses. For each training dataset we use a disjoint set of 4,000 responses as testing dataset. Each testing dataset was further partitioned into eight disjoint testing subsets, with 500 responses each. From each training dataset we generated five fitting subsets, by excluding contiguous and disjoint subsets of 20% of the responses of the training dataset. For each fitting subset, the excluded responses made the validation subset. The validation subset was further partitioned into eight disjoint subsets.

For example, the training subset with 20,000 responses contained responses to images 1 to 20,000. The corresponding testing dataset contained responses to images 20,001 to 24,000. From the training subset we generated five fitting subsets. For instance, the first fitting subset contained responses to images 1 to 16,000, and the corresponding validation subset contained responses to images 16,001 to 20,000. The second fitting subset contained responses to images 1 to 12,000 and 16,001 to 20,000, and the corresponding validation subset contained responses to images 12,001 to 16,000.

To assess the variability in ePPR estimates, for each condition (cell — stimuli type — number of spikes/image or number of inputs) we estimated five ePPR models using the five fitting subsets. Multiple estimations of ePPR models were needed to build Figures 2d, 2g, 2h, 6e, and 8e.

Due to the large computational demands of MID, in most conditions we estimated one set of MID filters, from the first of five jackknifed dataset. We only estimated five sets of MID filters for the characterization of the simulated cell, with the reference noise level, from its responses to natural or random stimuli. Multiple estimations of MID models were needed to build Figures 2d and g.

Similarity between two sets of filters

What matters to determine the similarity between two sets of filters is not the similarity between any pair of filters in the sets, but the similarity between the spaces spanned by the two sets of filters. Call \mathcal{S}_1 and \mathcal{S}_2 the spaces spanned by two sets of filters. If the dimension of \mathcal{S}_1 equals that of \mathcal{S}_2 , then principal angles (Golub and Loan 1996) are good measures to study their similarity. For the simulated complex cell, the dimension of the true filter space is three, which, for the reference noise level, equaled the dimensions of the filter spaces estimated with ePPR and MID. So, we used principal angles to study the similarity between the filters of the simulated cell and those estimated with ePPR or MID (Figures 2d, 2g, and 13a). When comparing two subspaces of dimension n one can compute n principal angles. This turned to be an advantage of principal angles respect to single-valued distance measures because, as explained below, the three principal angles of the simulated cell gave us further insight about the quality of the ePPR and MID estimates than what we could have obtained from a single-valued distance measure.

Principal angles. Let \mathcal{S}_1 and \mathcal{S}_2 be subspaces of \mathbb{R}^m whose dimensions satisfy

$$p = \dim(\mathcal{S}_1) \geq \dim(\mathcal{S}_2) = q \geq 1$$

The principal angles $\theta_1, \dots, \theta_q \in [0, \pi/2]$ between \mathcal{S}_1 and \mathcal{S}_2 are defined recursively by

$$\cos(\theta_k) = \max_{u \in \mathcal{S}_1} \max_{v \in \mathcal{S}_2} u^T v = u_k^T v_k$$

subject to:

$$\begin{aligned} \|u\| &= \|v\| = 1 \\ u^T u_i &= 0 & i = 1 : k-1 \\ v^T v_i &= 0 & i = 1 : k-1 \end{aligned}$$

Note that the principal angles satisfy $0 \leq \theta_1 \leq \dots \leq \theta_q \leq \pi/2$. The vectors $\{u_1, \dots, u_q\}$ and $\{v_1, \dots, v_q\}$ are called the principal vectors. If the columns of Q_1 and Q_2 define orthonormal bases of \mathcal{S}_1 and \mathcal{S}_2 , respectively, then $\cos(\theta_i) = \sigma_i$, with σ_i the i th singular value of $Q_1^T Q_2$.

The first principal angle, θ_1 , measures the dissimilarity between the closest vectors in \mathcal{S}_1 and \mathcal{S}_2 . Thus, θ_1 can be interpreted as an indication of how well \mathcal{S}_1 approximates \mathcal{S}_2 along one dimension. And for $k > 1$, the k th principal angle, θ_k , measures the dissimilarity between the closest vectors in the subspace of \mathcal{S}_1 orthogonal to the space spanned by $\{u_1, \dots, u_{k-1}\}$, and the subspace of \mathcal{S}_2 orthogonal to the space spanned by $\{v_1, \dots, v_{k-1}\}$. Then, θ_k indicates how well \mathcal{S}_1 approximates \mathcal{S}_2 along k dimensions. For the simulated cell, this interpretation of principal angles showed that MID approximations were good along one dimension, but poor along more than one dimension (Figures 2d and g).

ePPR model selection procedure

The backward-stepwise procedure, Listing 8, returns a set of models. We used a cross-validation procedure to select the best model from this set, as described here. To give a concrete example, we provide details of a cross-validation procedure used to obtain an ePPR model with time interactions estimated from simulated responses to natural stimuli with large levels of noise. For this estimation, the backward-stepwise procedure returned a set of models having between one and six terms. We then used each of these six models to predict responses to the eight validation subsets. Figure 10a plots the correlation coefficients between these predictions and the responses of the simulated cell, as a function of the number of terms of the ePPR models. For each number of terms, n , the value j along the y -axis, $1 \leq j \leq 8$, is the correlation coefficient between the responses from the simulated cell, to images in the j th validation subset, and the predictions of the ePPR model with n terms.

We seek to select the model maximizing predictive power, while containing the fewest terms; thus, the predictions of this model should be better than those of the models with fewer terms, and not worse than those of the models with more terms. To compare the predictive power between two models we test, using a non-parametric Wilcoxon signed-rank test, if the correlation coefficients of one of the models are larger/smaller than those of the other model. For the example cross-validation procedure, with the correlation coefficients shown in Figure 10a, we found the ePPR model with four terms yielded better predictions than the ePPR models with fewer terms ($p < 0.05$ for all Wilcoxon signed-rank tests). Also,

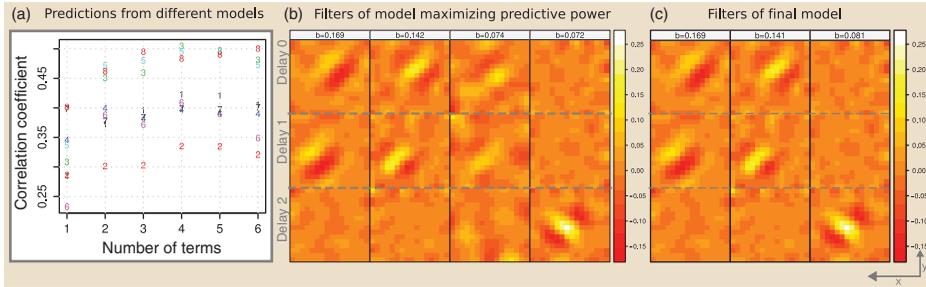


Figure 10. Selection of the best ePPR model by cross-validation. (a) correlation coefficients between predictions from ePPR models and responses of the cell, as a function of the number of terms in ePPR models. For each number of terms, n , the value j along the y -axis, $1 \leq j \leq 8$, is the correlation coefficient between the responses from the simulated cell to images in the j th validation subset, and the predictions of the ePPR model with n terms. The ePPR model with four terms predicts better than the models with smaller number of terms ($p < 0.05$ for all Wilcoxon signed-rank tests). We could not conclude that the ePPR models with four terms predicted worse than the model with more than four terms ($p > 0.1$ for all Wilcoxon signed-rank tests). Thus, the ePPR model maximizing predictive power contains four terms. (b): filters of the ePPR model maximizing predictive power. The third filter from the left is spurious. For the simulated cell, spurious filters appeared only in models estimated from responses with the two largest noise levels (0.16 and 0.35 spikes/image). They were removed from the model using the “Removal of spurious terms” procedure. (c): filters of final ePPR model obtained by applying the “Removal of spurious terms” procedure to the ePPR model maximizing predictive power.

we could not conclude that the ePPR model with four terms yielded worse predictions than the ePPR models with more terms ($p > 0.1$ for all Wilcoxon signed-rank tests); thus, to characterize this simulated cell from its responses to natural data, the best ePPR model contains four terms. The filters of this model are shown in Figure 10b.

For responses with high levels of noise, the ePPR model maximizing predictive power can contain spurious terms (e.g., third filter from the left in Figure 10b). To remove these spurious terms we use the following procedure.

Removal of spurious terms. In the plot of correlation coefficients versus number of ePPR terms (Figure 10a), we see that before reaching the number of terms at which the predictive power of ePPR models saturates (four terms in Figure 10a), in most cases, when the backward-stepwise procedure drops a term from a model with n terms, it yields a new model with $n - 1$ terms having worse predictive power. This means that the term dropped by the backward-stepwise procedure contributed to improve the predictions of the model with n terms; however, occasionally, the model with $n - 1$ terms has equal or better predictive power than the model with n terms (e.g., $n = 3$ in Figure 10a). This means that the term dropped by the backward-stepwise procedure did not contribute to improve the predictions of the model with n terms. With simulated data we verified that these dropped terms are spurious ones. Then, to detect spurious filters, for each number of terms, n , between two and the maximum number of terms, we test whether the correlation coefficients of the ePPR model with n terms are significantly larger than those of the ePPR model with $n - 1$ terms (Wilcoxon signed-rank test, $p < 0.05$). If the test does not reach

significance, the term dropped from the model with n terms is removed from the model maximizing predictive power.

In the example cross-validation procedure considered above, the correlation coefficients of the models with three terms were not significantly larger than those of the models with two terms (Wilcoxon signed-rank test, $p > 0.1$). We therefore removed from the ePPR model with four terms (Figure 10b) the term dropped by the backward-stepwise procedure from the model with three terms, obtaining a final model which filters are shown in Figure 10c.

Note that the procedures that we use to select the ePPR model maximizing predictive power and to remove spurious terms depend only on the outcomes of hypothesis tests, and not on user-dependent subjective criteria. With simulated data, these procedures enabled us to recover the true model, even under considerably noisy conditions. And with physiological data, for models estimated from different fitting subsets, or from natural and random data, this procedure produced very consistent results.

Upper bound on predictions

For the simulated responses, for which we knew the firing rate, $r = \bar{y}$ in Equation 8, we used as the upper bound the correlation coefficient between the noisy responses, y , and the firing rate, upper bound = $\rho(r, y)$.

For the responses of the real cell, for which we did not know the firing rate, we estimated it as the mean response of the M repeats of N stimuli, $\hat{r}(i) = \frac{1}{M} \sum_{m=1}^M y_m(i)$, $1 \leq i \leq N$, and then used as the upper bound the maximum correlation coefficient between the estimated firing rate and the response to any repeat of the stimuli, upperbound = $\max_{1 \leq m \leq M} \rho(\hat{r}, y_m)$. Note that when the number M of repeats is small, as in the complex cell characterized with random data, or the simple cell, the mean of the M repeats will be a poor estimate of the firing rate, and the resulting upper bound will be loose (Figures 6g, 8f and 8g).

ePPR hyper-parameters

For each estimation of an ePPR model (i.e., for each cell, stimuli type, and number of spikes/image or training data set size) we selected by cross validation the best λ parameter. This is the only hyperparameter that varied across estimated models of one cell. The memory of the forward model, D^L , and the number of terms at delay d of the forward model, M_d^L , were selected by cross-validation for each cell, and kept constant across different conditions (i.e., noise levels, type of stimuli, and number of spikes/image or training data set size). The values selected for these parameters for the example cells shown in this article appear in Table 1. The remainder ePPR hyper-parameters were kept constant for all the cells analyzed in this article ($d = 5$, addTermsCV = 0.01, refitCV = 0.001, r0 = 1, rmax = 1,000, and iterlim = 1,000).

MID

The maximally informative dimension algorithm (Sharpee et al. 2004) estimates a set of filters $\{\hat{v}_1, \hat{v}_2, \dots, \hat{v}_n\}$ that maximize the mutual information between cell

Table 1. ePPR hyper-parameters for the cells shown in this article. Only the values of the regularization parameter, λ , and the number of terms per delay for the forward model, M_d^L , are presented. The remainder hyperparameters were fixed for all the ePPR estimations (see text). $M_d^L = (i, j, k, \dots)$ stands for i terms at delay zero, j terms at delay one, k terms at delay two, \dots

Figure	λ	M_d^L
2b	15	6
2e	15	6
6a	75	10
6c	75	10
8a	10	10
8c	75	10
11a	15	6, 6, 4, 2
12a	75	6, 6, 6, 3
13b	15	6
14a	15	6, 6, 4, 2
14b	150	6
15b	3	3, 6, 6, 4, 3, 3

responses and projections of the stimuli on those filters; i.e., Equation 9, where X and Y are random variables associated with the stimuli and responses, respectively. This maximization is performed using a combination of gradient ascent and simulated annealing.

$$\hat{v}_1, \dots, \hat{v}_n = \arg \max_{v_1, \dots, v_n} I(\{X^T v_1, \dots, X^T v_n\}, Y) \tag{9}$$

In vision, MID has only been used to characterize simple cells, with only one spatio-temporal filter (Sharpee et al. 2006, 2008). But recently an implementation of MID for the estimation of multiple filters became available from the laboratory of Dr. Sharpee (<http://cnl-t.salk.edu/Download/>). Although this implementation has not been described in the literature, or demonstrated with physiological data, we used it in this article to estimate multiple MID filters.

Selection of good initial conditions is essential for the numerical optimization of a nonlinear function with as many parameters as those of the mutual information in Equation 9. To estimate an initial condition for the spatio-temporal filter v_1 , the implementation of MID used in this article solves Equation 9 with respect to only one filter; i.e., $v_1^0 = \arg \max_v I(X^T v, Y)$. Having found initial conditions for the first j filters, $\{v_1^0, \dots, v_j^0\}$, the initial condition for the filter $j + 1$ is found by again solving Equation 9 with respect to only one filter, but constraining this filter to be in the orthogonal space to that spanned by the first j filters; i.e., $v_{j+1}^0 = \arg \max_{v \in \{v_1^0, \dots, v_j^0\}^\perp} I(X^T v, Y)$. Once the initial conditions for all the filters have been estimated, the final MID estimates are obtained by solving Equation 9 jointly with respect to all the filters.

In this article we used our best estimate of the dimension of the true low-dimensional space as the number n of MID filters to estimate. For the simulated cell we used $n = 3$. For the cortical cells we used the number of filters in ePPR models estimated with the largest number of inputs. For the cortical complex cell probed

with natural stimuli we wanted to use $n=5$, but we could not do it due to the large memory requirements of the implementation of MID used in this article, and we had to use $n=4$. For the cortical complex cell probed with random stimuli we used $n=3$, and for the cortical simple cell probed with natural and random stimuli we used $n=2$ and $n=1$, respectively.

In all our MID estimates we used 1000 iterations, 15 bins, and the default annealing parameters (`max_annealing_iterations=1`, `start_temperature=0.01`, `stop_temperature=1.0e-5`, `down_temperature_factor=0.95`, `up_temperature_factor=5.0`, `function_tolerance=5.0e-5`, `updateFactor=10`).

nSTC

For Gaussian white noise stimuli, if a cell is selective to a set of relevant dimensions, then the variance of stimuli that elicit spikes (spike-triggered stimuli) along these dimensions should be higher or lower than the variance along non-relevant dimensions. The dimensions with high or low variance correspond to the eigenvectors of the autocovariance matrix with correspondingly high or low eigenvalues. For Gaussian stimuli, STC identifies the relevant dimensions of a cell as the eigenvectors of the autocovariance matrix of the spike-triggered stimuli that correspond to significantly high or low eigenvalues.

Touryan et al. (2005) proposed a modification to STC for natural stimuli. This modification starts by whitening the natural stimuli. Denote by U the matrix of eigenvectors of the autocovariance of the stimuli (one eigenvector per column), and by λ_i to its eigenvalues. The matrix of normalized eigenvectors is defined as

$$U_n = U \begin{pmatrix} \frac{1}{\sqrt{\lambda_1}} & & 0 \\ & \ddots & \\ 0 & & \frac{1}{\sqrt{\lambda_n}} \end{pmatrix} \quad (10)$$

Then the whitened natural images are $X_w = XU_n$. STC for natural images now performs a classical STC analysis on the whitened natural images, obtaining a set of relevant dimensions, V_w (one relevant dimension per column). Finally, the desired relevant dimensions of the cell, V (one relevant dimension per column), are $V = U_n V_w$.

The autocovariance matrix of natural images is nearly singular, so its last eigenvalues (λ_i , $i \gg 1$) will be very small and tend to amplify noise. To avoid this effect, a threshold is selected and, for each eigenvalue λ_j less than this threshold, the diagonal value $1/\sqrt{\lambda_j}$ in Equation 10 is set to zero. The results reported here correspond to using a threshold such that approximately 35% of the eigenvalues were greater than it, as in Touryan et al. (2005).

rSTA

Suppose the responses of a cell are generated by a static nonlinearity on the projection of input images along a single relevant dimension. In other words,

assume $y(x) = N(\alpha^T x)$, where $y(x)$ is the response of the cell to image x , α is a linear filter, and N is a static nonlinearity. If the input images are Gaussian white noise, the filter α can be estimated by cross-correlating the responses with the images, $\hat{\alpha} = C_{yw}$ (de Boer and Kuyper 1968). But, natural images are not Gaussian white noise (Field 1987; Ruderman and Bialek 1994; Simoncelli and Olshausen 2001). Nevertheless, if the response of the cell is a linear function of its inputs, the filter α can be estimated as $\hat{\alpha} = C_{wvw}^{-1} C_{yw}$, where C_{wvw} represents the autocorrelation of the inputs (Theunissen et al. 2001). The autocorrelation matrix, C_{wvw} , for natural stimuli is nearly singular; therefore, its true inverse tends to amplify noise. To avoid this problem, we regularized the autocorrelation matrix using the truncated singular value decomposition (Hansen 1987) and computed the pseudoinverse (Ben-Israel and Greville 1980) from this regularized matrix (Smyth et al. 2003). The computation of the truncated singular value decomposition uses a threshold to decide how many singular values to include in the regularized matrix. We selected the optimal threshold using k-fold cross-validation (Efron and Tibshirani 1993).

Polynomial predictive model

Given a set of M spatio-temporal filters $\{f_1, \dots, f_M\}$, each with a temporal extent of D delays, to predict the response of a cell at time t , we first computed the dot product between each of the M filters and the concatenation of the images presented at times $t, t-1, \dots, t-(D-1)$; i.e., we computed the vector $v_t(m) = f_m^T[x(t), \dots, x(t-D)]$, $1 \leq m \leq M$. Then, we used this vector as input to a second-order multi-dimensional polynomial (Equation 11) to generate the predictions at time t of the simulated (Figures 4 and 14d) and cortical (Figure 6f and g) complex cells. Also, we used this vector as input to a third-order multi-dimensional polynomial (Equation 12) to generate the predictions of the cortical simple cell (Figure 8f and g). The coefficients of a polynomial model were estimated by minimizing the mean-squared error between responses of the cell and predictions from polynomial models.

$$y(t) = k_0 + \sum_{m=1}^M k_1(m)v_t(m) + \sum_{m_1=1}^M \sum_{m_2=m_1}^M k_2(m_1, m_2)v_t(m_1)v_t(m_2) \quad (11)$$

$$y(t) = k_0 + \sum_{m=1}^M k_1(m)v_t(m) + \sum_{m_1=1}^M \sum_{m_2=m_1}^M k_2(m_1, m_2)v_t(m_1)v_t(m_2) + \sum_{m_1=1}^M \sum_{m_2=m_1}^M \sum_{m_3=m_2}^M k_3(m_1, m_2, m_3)v_t(m_1)v_t(m_2)v_t(m_3) \quad (12)$$

To predict responses from the purely spatial filters of the ePPR models without time interactions in Figures 11c and 12c (pink curves, Appendix D), we used the spatio-temporal polynomial model in Equation 13. Given the spatial filters $\{f_m^d\}$, $0 \leq d \leq D$, $1 \leq m \leq M_d$ we first computed the dot products between each of these filters and the image at delay d ; i.e., we computed the vectors $v_t^d(m) = (f_m^d)^T x(t-d)$, $0 \leq d \leq D$, $1 \leq m \leq M_d$. Then, we used these vectors as inputs to the polynomial model in Equation 13.

$$\begin{aligned}
y(t) = & k_0 + \sum_{d=1}^D \sum_{m=1}^{M_d} k_1^d(m) v_t^d(m) + \sum_{d=0}^D \sum_{m_1=1}^{M_d} \sum_{m_2=m_1}^{M_d} k_2^d(m_1, m_2) v_t^d(m_1) v_t^d(m_2) \\
& + \sum_{d_1=0}^{D-1} \sum_{d_2=d_1+1}^D \sum_{m_1=1}^{M_{d_1}} \sum_{m_2=1}^{M_{d_2}} k_2^{d_1, d_2}(m_1, m_2) v_t^{d_1}(m_1) v_t^{d_2}(m_2) \quad (13)
\end{aligned}$$

Testing for difference in the predictions

To compare the predictive power of two models, we used them to predict responses to the eight testing subsets, and computed Pearson correlation coefficients between these predictions and cell responses. In this way, we constructed two sets of eight correlation coefficients, one for each model^{VI}. In most cases, the variability of the correlation coefficients in each set was large, and the error bars (size two standard deviation) of the mean correlation coefficients of the two models intersected. However, it was frequently the case that for each testing subset the correlation coefficient of one of the models was larger than that of the other model. So, to compare the predictive power of the two models, for each testing subset we paired the correlation coefficients of the two models, and used a one-sided Wilcoxon signed-rank test (Hollander and Wolfe 1999) to check if one element of the pair was significantly larger than the other element.

Notes

- [I] A more standard definition of an LN model would use spatio-temporal filters, concatenating the i th filters at all delays to give the i th spatio-temporal filter. But our definition is more convenient to introduce of the extended Projection Pursuit Regression model below, while both definitions are equivalent. (page 36)
- [II] In the vector representation, the columns of an image are concatenated together to form a long vector. (page 40)
- [III] PPR and nSTC, as implemented here, can only estimate spatial filters; i.e., filters that operate on a single image presented to the cell prior to its response. For these methods, we estimate filters operating on the image presented to the cell at the same time bin as the response. This choice optimized the models predictions. (page 45)
- [IV] For filters estimated from 10,000 and 20,000 natural stimuli, ePPR and MID predictions were not statistically different (two-sided Wilcoxon signed-rank test $p > 0.01$). For filters estimated from 3,000 natural stimuli, predictions from MID filters were significantly better than those from ePPR filters (one-sided Wilcoxon signed-rank test $p < 0.01$). However, this difference is probably due to the fact that ePPR predictions were computed using only one filter, while MID predictions were computed using two filters. From 3,000 inputs, the ePPR the model selection procedure found only one filter, while MID used the number of filters found by the ePPR models selection procedure using 20,000 inputs. (page 54)
- [V] PPR models were estimated using the images presented to the cell at the time bin before the response of the cell. (page 54)
- [VI] For each condition (cell — stimuli type — spikes/image or number of inputs) we estimated five sets of ePPR filters, and computed a mean set of eight correlation coefficient, one for each testing subset, by averaging the correlation coefficients of the five sets of ePPR filters for the eight testing subsets; however, due to the large

computational requirements of MID, in most conditions we only estimated one set of MID filters, and used it to compute one set of eight correlation coefficients. (page 70)

Declaration of interest: The authors report no conflicts of interest. The authors alone are responsible for the content and writing of the paper.

References

- Adelson E, Bergen J. 1985. Spatiotemporal energy models for the perception of motion. *Journal of the Optical Society of America A* 2:284–299.
- Barlow H. 1953. Summation and inhibition in the frog's retina. *Journal of Physiology (London)* 119:69–88.
- Bellman R. 1961. *Adaptive control processes*. Princeton, NJ: Princeton University Press.
- Ben-Israel A, Greville T. 1980. *Generalized inverses: Theory and applications*. Huntington, NY: Krieger.
- Bertsekas D. 1999. *Nonlinear programming*. 2nd edn. Nashua, NH: Athena Scientific.
- Betsch B, Einhäuser W, König K, König P. 2004. The world from a cat's perspective – statistics of natural videos. *Biological Cybernetics* 90(1):41–50.
- Burnham K, Anderson D. 2002. *Model selection and multimodel inference: A practical information-theoretic approach*. Berlin, Germany: Springer.
- Chen X, Han F, Poo M, Dan Y. 2007. Excitatory and suppressive receptive field subunits in awake monkey primary visual cortex (V1). *PNAS* 104(48):19120–19125.
- Chichilnisky E. 2001. A simple white noise analysis of neuronal light response. *Network: Computation Neural Systems* 12:199–213.
- Christianson G, Sahani M, Linden J. 2008. The consequences of response nonlinearities for interpretation of spectrotemporal receptive fields. *The Journal of Neuroscience* 28(2):446–455.
- David S, Mesgarani N, Fritz J, Shamma S. 2009. Rapid synaptic depression explains nonlinear modulation of spectro-temporal tuning in primary auditory cortex by natural stimuli. *The Journal of Neuroscience* 29(11):3374–3386.
- David S, Vinje W, Gallant J. 2004. Natural stimulus statistics alter the receptive field structure of V1 neurons. *The Journal of Neuroscience* 24(31):6991–7006.
- de Boer E, Kuyper P. 1968. Triggered correlation. *IEEE Transactions on Biomedical Engineering* 15:169–179.
- de Ruyter van Steveninck R, Bialek W. 1988. Real-time performance of a movement-sensitive neuron in the blowfly visual system: Coding and information transfer in short spike sequences. *Proceedings Royal Society of London B* 234:379–414.
- DeAngelis G, Ohzawa I, Freeman R. 1993a. Spatiotemporal organization of simple-cell receptive fields in the cat's striate cortex I. General characteristics and postnatal development. *Journal of Neuroscience* 69:1091–1117.
- DeAngelis G, Ohzawa I, Freeman R. 1993b. Spatiotemporal organization of simple-cell receptive fields in the cat's striate cortex II. Linearity of temporal and spatial summation. *Journal of Neuroscience* 69:1118–1135.
- Diaconis P, Shahshahani M. 1984. On nonlinear functions of linear combinations. *SIAM Journal on Scientific Computing* 5(1):175–191.
- Donoho D, Johnstone I, Rousseeuw P, Stahel W. 1985. Discussion: Projection pursuit. *The Annals of Statistics* 13(2):496–500.
- Efron B, Tibshirani R. 1993. *An introduction to the bootstrap*. New York, NY: Chapman & Hall.
- Felsen G, Dan Y. 2005. A natural approach to studying vision. *Nature neuroscience* 8(12):1643–1646.
- Felsen G, Touryan J, Han F, Dan Y. 2005. Cortical sensitivity to visual features in natural scenes. *Public Library of Science Biology* 3(10):e342.
- Field D. 1987. Relations between the statistics of natural images and the response properties of cortical cells. *Journal of the Optical Society of America A* 4(12):2379–2394.
- Friedman J. 1984a. *Smart user's guide* (Tech. Rep. No. 1). Palo Alto, CA: Department of Statistics, Stanford University.
- Friedman J. 1984b. *A variable span scatterplot smoother* (Tech. Rep. No. 5). Palo Alto, CA: Department of Statistics, Stanford University.

- Friedman J, Stuetzle W. 1981. Projection pursuit regression. *Journal of the American Statistical Association* 76(376):817–823.
- Golub G, van Loan C. 1996. *Matrix computations*. Baltimore, MD: The Johns Hopkins University Press.
- Green P, Silverman B. 1994. *Nonparametric regression and generalized linear models*. Boca Raton, FL: CRC Press.
- Hansen P. 1987. The truncated SVD as a method for regularization. *BIT* 27(4):534–553.
- Hartline H. 1940. The receptive fields of optic nerve fibers. *American physiological society* 130(4):690–699.
- Heeger D. 1992. Normalization of cell responses in cat striate cortex. *Visual Neuroscience* 9:181–197.
- Hollander M, Wolfe D. 1999. *Nonparametric statistical methods*. 2nd edn. New York, NY: Wiley-Interscience.
- Huber P. 1985. Projection pursuit. *The Annals of Statistics* 13(2):435–475.
- Hwang J, Lay S, Maechler M, Martin R, Schimert J. 1994. Regression modelling in back-propagation and projection pursuit learning. *IEEE Transactions on Neural Networks* 5(3):342–353.
- Jones L. 1987. On a conjecture of Huber concerning the convergence of projection pursuit regression. *The Annals of Statistics* 15(2):880–882.
- Korenberg M, Hunter I. 1986. The identification of nonlinear biological systems: LNL cascade models. *Biological Cybernetics* 55:125–134.
- Kuffler S. 1953. Discharge patterns and functional organization of mammalian retina. *Journal of Neurophysiology* 16:37–68.
- Liberman M. 1982. The cochlear frequency map for the cat: Labeling auditory-nerve fibers of known characteristic frequency. *Journal of the acoustical society of America* 72:1441–1449.
- Marmarelis P, Marmarelis V. 1978. *Analysis of physiological systems: The white-noise approach*. New York, NY: Plenum Press.
- Marmarelis V. 1993. Identification of nonlinear biological systems using Laguerre expansions of kernels. *Annals of Biomedical Engineering* 21:573–589.
- Mechler F, Ringach D. 2002. On the classification of simple and complex cells. *Vision Research* 42(8):1017–1033.
- Movshon J, Thompson I, Tolhurst D. 1978a. Receptive field organization of complex cells in the cat's striate cortex. *Journal of Physiology* 383:79–99.
- Movshon J, Thompson I, Tolhurst D. 1978b. Spatial summation in the receptive fields of simple cells in the cat's striate cortex. *Journal of Physiology* 383:53–77.
- Nocedal J, Wright S. 2006. *Numerical optimization*. New York, NY: Springer.
- Paninski L. 2003. Convergence properties of three spike-triggered analysis techniques. *Network: Computation in Neural Systems* 14:437–464.
- Prenger R, Wu M-K, David S, Gallant J. 2004. Nonlinear V1 responses to natural scenes revealed by neural network analysis. *Neural Networks* 17:663–679.
- Rapela J, Mendel J, Grzywacz N. 2006. Estimating nonlinear receptive fields from natural images. *Journal of Vision* 6(4):441–474.
- Roosen C, Hastie T. 1994. Automatic smoothing spline projection pursuit. *Journal of Computational and Graphical Statistics* 3(3):235–248.
- Ruderman DL, Bialek W. 1994. Statistics of natural images: Scaling in the woods. *Physical Review Letters* 76(6):814–817.
- Rudin W. 1976. *Principles of mathematical analysis*. 3rd edn. Columbus, OH: McGraw-Hill.
- Rust N, Schwartz O, Movshon J, Simoncelli E. 2005. Spatiotemporal elements of macaque V1 receptive fields. *Neuron* 46:945–956.
- Sharpee T, Miller K, Stryker M. 2008. On the importance of static nonlinearity in estimating spatiotemporal neural filters with natural stimuli. *Journal of Neurophysiology* 99:2496–2509.
- Sharpee T, Rust N, Bialek W. 2004. Analyzing neural responses to natural signals: Maximally informative dimensions. *Neural computation* 16:223–250. (Code available at <http://cnl-t.salk.edu/Download/>.)
- Sharpee T, Sugihara H, Kurgansky A, Rebrik S, Stryker M, Miller K. 2006. Adaptive filtering enhances information transmission in visual cortex. *Nature* 439(7079):936–942.
- Simoncelli E, Olshausen B. 2001. Natural image statistics and neural representations. *Annual Review of Neuroscience* 24:1193–216.
- Simoncelli E, Paninski L, Pillow J, Schwartz O. 2004. Characterization of neural responses with stochastic stimuli. In: Gazzaniga M, editor. *The new cognitive neurosciences*. 3rd edn. Cambridge, MA: MIT Press. pp 327–338.

- Smyth D, Willmore B, Baker G, Thompson I, Tolhurst D. 2003. The receptive-field organization of simple cells in primary visual cortex of ferrets under natural scene stimulation. *The Journal of Neuroscience* 23:4746–4759.
- Spekreijse H. 1969. Rectification in the goldfish retina: Analysis by sinusoidal and auxiliary stimulation. *Vision Research* 9:1461–1472.
- Theunissen F, David S, Singh N, Hsu A, Vinje W, Gallant J. 2001. Estimating spatio-temporal receptive fields of auditory and visual neurons from their responses to natural stimuli. *Network: Computation in Neural Systems* 12:289–316.
- Theunissen F, Sen K, Doupe A. 2000. Spectral-temporal receptive fields of nonlinear auditory neurons obtained using natural sounds. *Journal of Neuroscience* 20(6):2315–2331.
- Touryan J, Felsen G, Dan Y. 2005. Spatial structure of complex cell receptive fields measured with natural images. *Neuron* 45:781–791.
- Touryan J, Lau B, Dan Y. 2002. Isolation of relevant visual features from random stimuli for cortical complex cells. *Journal of Neuroscience* 22(24):10811–10818.
- van Hateren J, Ruderman D. 1998. Independent component analysis of natural image sequences yields spatio-temporal filters similar to simple cells in primary visual cortex. *Proceedings Royal Society London B* 265:2315–2320.
- Weiss T. 1966. A model of the peripheral auditory system. *Kybernetik* 3:153–175.
- Wiener N. 1958. *Nonlinear problems in random theory*. New York: Wiley.
- Woolley S, Gill P, Theunissen F. 2006. Stimulus-dependent auditory tuning results in synchronous population coding of vocalization in the songbird midbrain. *Journal of Neuroscience* 26(9):2499–2512.
- Woolley S, Fremouw T, Hsu A, Theunissen F. 2005. Tuning for spectro-temporal modulations as a mechanism for auditory discrimination of natural sounds. *Nature Neuroscience* 8:1371–1379.

Appendix A: PPR

PPR algorithm

A detailed description of the PPR algorithm appears in Friedman (1984a). For completeness we provide an abridged description here. Equation 1 describes the PPR model whose parameters are optimized by minimizing Equation 2, as described algorithmically in Listing 1. PPR consists of a forward stepwise procedure, followed by a backward stepwise procedure. The parameters of the algorithm are the responses, y , the input images x , the length of the forward model, M_L , the length of the final PPR model, M_0 , and the degree of smoothness of the nonlinear functions, d .

Listing 1 PPR

Require: y, x, M_L, M_0, d

- 1: (model, r) \leftarrow FORWARD_STEPWISE(y, x, M_L, d) {Built forward model $\hat{y}(x) = \bar{y} + \sum_{m=0}^{M_L} \beta_m \phi_m(\alpha_m^T x)$ }
 - 2: model \leftarrow BACKWARD_STEPWISE(model, r, x, M_0, d) {Obtained model $\hat{y}(x) = \bar{y} + \sum_{m=0}^{M_0} \beta_m \phi_m(\alpha_m^T x)$ }
 - 3: **return** model
-

PPR forward stepwise procedure. For the forward procedure, an initial M_L -term model of the form given in Equation 1 is constructed. An algorithmic description of this procedure is given in Listing 2. It first defines the residuals r as the mean-subtracted responses. Then, it obtains an initial estimate of α by reverse correlation.

Next, it fits the first term of the model, as described below, obtaining the the approximation $y = \bar{y} + \beta_1 \phi_1(\alpha_1^T x)$. It next defines the new residuals $r = r - \beta_1 \phi_1(\alpha_1^T x)$ and fits to them the second term of the model. This gives the approximation $y = \bar{y} + \beta_1 \phi_1(\alpha_1^T x) + \beta_2 \phi_2(\alpha_2^T x)$. Continuing in this fashion, it arrives at the forward stepwise estimated model.

Listing 2 PPR: FORWARD_STEPWISE

Require: y, x, M_L, d

```

1:  $r \leftarrow y - \bar{y}$ 
2: for  $m \in 1$  to  $M_L$  do
3:    $\alpha \leftarrow \text{GET\_INITIAL\_ALPHA}(r, x)$ 
4:    $(\beta_m, \phi_m, \alpha_m) \leftarrow \text{FIT\_NEW\_TERM}(r, x, \alpha, d)$  {Find  $(\beta_m, \phi_m, \alpha_m)$  minimizing  $\mathcal{J} = \sum_{i=1}^N (r - \beta_m \phi_m(\alpha_m^T x))^2$ }
5:    $r = r - \beta_m \phi_m(\alpha_m^T x)$ 
6: end for {Built model  $\hat{y}(x) = \bar{y} + \sum_{m=0}^{M_L} \beta_m \phi_m(\alpha_m^T x)$ }
7: return (model =  $[(\beta_1, \phi_1, \alpha_1), \dots, (\beta_{M_L}, \phi_{M_L}, \alpha_{M_L})], r = r$ )

```

PPR backward stepwise procedure. The forward stepwise procedure is greedy and it is not guaranteed to converge to the optimal model. The terms in the model estimated by the forward stepwise procedure do not necessarily appear in decreasing order of importance. To estimate a model with the M_0 most important terms, a backward stepwise procedure is used. The strategy is to fit the forward model with a relatively large value of M_L , and then to search for models of size $M_{L-1}, M_{L-2}, \dots, M_0$ with optimal terms. The starting parameter values to search for the M -term model with optimal terms are the M most important terms of the model with $M+1$ terms. Term importance is measured by $|\beta_m|$ ($1 \leq m \leq M+1$). The starting parameters values to search for the $(M_L - 1)$ -term model with optimal terms are given by the forward stepwise model.

An algorithmic description of the backward stepwise procedure is given in Listing 3. The procedure repeatedly drops terms from the model, until the model contains only M_0 terms. After each term is dropped the residuals are adjusted according to the contributions of the dropped term, and all the terms of the model are refitted.

Listing 3 PPR: BACKWARD_STEPWISE

Require: model, r, x, M_0, d

```

1: for  $m \in M_L$  downto  $M_0$  do
2:    $((\beta, \phi, \alpha), \text{model}) \leftarrow \text{DROP\_LEAST\_IMPORTANT\_TERM}(\text{model})$ 
3:    $r \leftarrow r + \beta \phi(\alpha^T x)$ 
4:   model  $\leftarrow \text{REFIT\_MODEL}(\text{model}, r, x, d)$ 
5: end for
6: return model

```

The refit model procedure is described in Listing 4. It starts by refitting the β_m parameters in Equation 1, while fixing the remaining ϕ_m and α_m parameters. This is a linear problem, with regressors $\phi_m(\alpha_m^T x)$ and dependent variable $y(x)$, that can be solved using standard linear regression techniques. Then, following the PPR estimation strategy, terms are refitted one at a time. In doing so, a term is removed from the model, residuals are adjusted according to the contribution of the removed term, a new term is fitted to the adjusted residuals, as indicated below, the newly fitted term is added to the model, and the residuals are adjusted again according to the contributions of this term.

Listing 4 PPR: REFIT_MODEL

Require: model, r , x , d

```

1: model  $\leftarrow$  REFIT_MODEL_BETAS(model,  $r$ )
2: for  $m \in 1$  to  $M$  do
3:    $((\beta_m, \phi_m, \alpha_m), \text{model}) \leftarrow$  REMOVE_TERM_FROM_MODEL( $m$ , model)
4:    $r \leftarrow r + \beta_m \phi_m(\alpha_m^T x)$ 
5:    $(\beta, \phi, \alpha) \leftarrow$  FIT_NEW_TERM( $r$ ,  $x$ ,  $\alpha_m$ ,  $d$ )
6:   model  $\leftarrow$  ADD_TERM_TO_MODEL( $(\beta, \phi, \alpha)$ , model)
7:    $r \leftarrow r - \beta \phi(\alpha^T x)$ 
8: end for
9: return model
  
```

PPR fit new term procedure. When fitting a new term the sum of square error (SSE) in Equation 14 is minimized with respect to the smooth function ϕ , and the projection direction α . This is done iteratively. In each iteration α is first fixed and ϕ is adjusted to minimize the SSE. Then ϕ is fixed and α is adjusted to minimize the SSE. This iteration is repeated until the SSE stops decreasing. Finally, β is calculated and ϕ is normalized. The procedure is described algorithmically in Listing 5.

$$SSE = \sum_{i=1}^N (r_i - \phi(\alpha^T x_i))^2 \quad (14)$$

Listing 5 PPR: FIT_NEW_TERM

Require: r , x , α , d

```

1: repeat
2:   Fix  $\alpha$  and find  $\phi_{k+1}$  minimizing  $SSE(\alpha, \phi) = \sum_{i=1}^N (r_i - \phi(\alpha^T x_i))^2$  {Smooth the scatter plot  $(\alpha^T x_i, r_i)$  using a smoothing spline or Friedman's 'super smoother.'}
3:   Fix  $\phi_{k+1}$  and update  $\alpha$  along a Gauss-Newton descent direction  $\delta_k$  of  $SSE(\alpha, \phi_{k+1})$ :  $\alpha_{k+1} = \alpha_k + \delta_k$ .
  
```

- 4: **until** SSE stops decreasing
- 5: $\beta \leftarrow \sqrt{E\{\phi^2(\alpha^T x)\}}$
- 6: $\phi \leftarrow \phi/\beta$
- 7: **return** (β, ϕ, α)

To adjust ϕ , with α fixed, the procedure projects the input images onto α , $p_i = \alpha^T x_i$, builds a scatter plot with the pairs of projections and residuals, (p_i, r_i) , and smooths the scatter plot using a smoothing spline (Green and Silverman 1994), or Friedman's 'super smoother' (Friedman 1984b), giving the adjusted ϕ .

With ϕ fixed the SSE in Equation 14 can be expressed as the L^2 norm of a nonlinear function h depending on α , as shown in Equation 15. Then, finding the value of α minimizing the SSE reduces to a nonlinear least-squares problem, which PPR solves using the Gauss-Newton algorithm (Bertsekas 1999).

$$SSE = \sum_{i=1}^N (r_i - \phi(\alpha^T x_i))^2 = \sum_{i=1}^N |h_i(\alpha)|^2 = \|h(\alpha)\|_2^2 \quad (15)$$

where $h_i(\alpha) = r_i - \phi(\alpha^T x_i)$
and $h(\alpha) = (h_1(\alpha), \dots, h_N(\alpha))$

Briefly, the Gauss-Newton algorithm produces a sequence $\alpha_1, \alpha_2, \dots$ such that $\|h(\alpha_k)\|_2^2$ converges to a minimum. Given α_k , we obtain α_{k+1} using Equation 16. The value of δ_k is selected to minimize $\|h(\alpha_{k+1})\|_2^2$. In doing so the Gauss-Newton algorithm approximates $h(\alpha_{k+1})$ using its first-order Taylor expansion, $\hat{h}(\alpha_{k+1}) = h(\alpha_k) + \nabla^T h(\alpha_k) \delta_k$, and estimates δ_k satisfying Equation 17. From the first-order necessary conditions for a minimum of Equation 17, δ_k is the solution of the linear system of equations in Equation 18.

$$\alpha_{k+1} = \alpha_k + \delta_k \quad (16)$$

$$\begin{aligned} \delta_k &= \arg \min_{\delta} \frac{1}{2} \|h(\alpha_k) + \nabla^T h(\alpha_k) \delta\|_2^2 \\ &= \arg \min_{\delta} \frac{1}{2} [\|h(\alpha_k)\|_2^2 + 2\nabla h(\alpha_k) h(\alpha_k) \delta + \delta^T \nabla h(\alpha_k) \nabla^T h(\alpha_k) \delta] \end{aligned} \quad (17)$$

$$\nabla h(\alpha_k) \nabla^T h(\alpha_k) \delta_k = -\nabla h(\alpha_k) h(\alpha_k) \quad (18)$$

In PPR α_k is not updated until $\|\hat{h}(\alpha_k)\|_2^2$ converges to a minimum. Instead, α_k is updated only once and then the algorithm proceeds with the next iteration.

Selection of PPR hyperparameters

Number of terms in the forward (M_L) and final (M_0) model. As described above, the PPR algorithm is controlled by parameters M_L , the number of terms in the forward model, and M_0 , the number of terms in the final model. To estimate these parameters we fitted a PPR model using a very large value for M_L , usually $M_L = 10$, and with $M_0 = 1$. The PPR algorithm returns the goodness of fit (SSE, Equation 2,

for the training dataset) of the backfitted models having between M_0 and M_L terms. A plot of these goodness of fit values, as a function of the number of terms in the PPR model, usually has an L shape. Before a given point, as we increase the number of terms in PPR models, the goodness of fit improves considerably; but, after this point, increasing the number of terms improves the goodness of fit only marginally. So, we selected the number of terms at which the SSE stops decreasing considerably as the number of terms, M_O , for the final PPR model. To allow refitting of the terms in the final PPR model, we set the number of terms in the forward model to $M_L = M_0 + 3$.

Degree of smoothness of the nonlinear functions (d). To estimate the smooth functions we used super smoother (Friedman 1984b) with automatic span selection.

Appendix B: ePPR

ePPR algorithm

Equation 6 describes the ePPR model whose parameters are optimized by minimizing Equation 7. To overcome the curse of dimensionality, ePPR retains the optimization strategy of PPR. There are three main differences between the PPR and ePPR estimation algorithms. First, ePPR extends PPR to become spatio-temporal. For the model without time interactions, after fitting the responses to images presented at the same time bin as the responses, residuals are fitted to images presented at previous time bins. For models with time interactions, images at several delays are concatenated to form spatio-temporal inputs, and spatial ePPR models are then estimated using these spatio-temporal inputs; i.e., the ePPR estimation procedure is invoked with a delay $D^L = 0$ for the forward model (see below). Second, to avoid problems caused by correlations in natural images, ePPR uses a Trust Region method, instead of a Gauss Newton method, to solve the nonlinear least-squares problem in Equation 20. Third, to obtain smooth projection directions, ePPR penalizes the criterion used in PPR to fit a new term (Equation 3), as shown in Equation 19. The first difference requires minor changes in the FORWARD_STEPWISE, BACKWARD_STEPWISE, and REFIT_MODEL procedures for the estimation of models without time interactions, and requires no change in the PPR algorithm for the estimation of models with time interactions. The last two differences only require changing the FIT_NEW_TERM procedure.

The ePPR estimation algorithm is described in Listing 6. It consists of a forward stepwise procedure, followed by a backward stepwise procedure, and a model selection procedure. The parameters of the algorithm are the responses, y , the input images, x , the number of terms at each delay d for the forward model, $M_d^L, 0 \leq d \leq D^L$, the regularization parameter for the α 's, λ , and the number of degrees of freedom for the ϕ 's, d .

Listing 6 ePPR

Require: $y, x, \{M_0^L, \dots, M_{D^L}^L\}, \lambda, d$

1: (model, r) \leftarrow FORWARD_STEPWISE($y, x, D^L, \{M_0^L, \dots, M_{D^L}^L\}, \lambda, d$) {Built forward model $\hat{y}(i) = \bar{y} + \sum_{d=0}^{D^L} \sum_{m=0}^{M_d^L} \beta_{m,d} \phi_{m,d}(\alpha_{m,d}^T x_{i-d})$ }

```

2: models ← BACKWARD_STEPWISE( $r, x, \lambda, d$ )
   {Obtained models of the form  $\hat{y}(x) = \bar{y} + \sum_{d=0}^D \sum_{m=0}^{M_d} \beta_{m,d} \phi_{m,d}(\alpha_{m,d}^T x_{i-d})$ , having
   between 1 term and the maximum number of terms in the forward model.}
3: model ← SELECT_BEST_MODEL(models)
4: return model

```

ePPR forward stepwise procedure. The forward procedure estimates an ePPR model with terms at delays $d=0, \dots, D^L$ and containing M_d^L terms at delay d . An algorithmic description of this procedure is given in Listing 7. It first defines the residuals r as the mean-subtracted responses. Then it fits to these residuals M_0^L terms operating on images presented at the same time bin as the responses. These M_0^L terms are fitted with the ePPR FIT_NEW_TERM procedure described below. Next, the input images are shifted in time, so that the image presented at time $i-1$ is displaced to time i , and a new set of M_1^L terms operating on these shifted images is fitted to the response residuals. In this way the response at time i has been approximated by M_0^L terms operating on the image presented at time i , plus additional M_1^L terms operating on the image presented at time $i-1$. After shifting the image D^L times, the forward ePPR model is constructed.

Listing 7 ePPR: FORWARD_STEPWISE

```

Require:  $y, x, \{M_0^L, \dots, M_{D^L}^L\}, \lambda, d$ 
1:  $r \leftarrow y - \bar{y}$ 
2: for  $d \in 0$  to  $D^L$  do
3:   for  $m \in 1$  to  $M_d^L$  do
4:      $\alpha \leftarrow$  GET_INITIAL_ALPHA( $r, x$ )
5:      $(\beta_{m,d}, \phi_{m,d}, \alpha_{m,d}) \leftarrow$  FIT_NEW_TERM( $r, x, \lambda, d$ )
6:      $r = r - \beta_{m,d} \phi_{m,d}(\alpha_{m,d}^T x)$ 
7:   end for
8:    $x_i \leftarrow x_{i-1}$ 
9: end for {Built model  $\hat{y}(i) = \bar{y} + \sum_{d=0}^{D^L} \sum_{m=1}^{M_d^L} \beta_{m,d} \phi_{m,d}(\alpha_{m,d}^T x_{i-d})$ }
10: return model =  $[(\beta_{1,0}, \phi_{1,0}, \alpha_{1,0}), \dots, (\beta_{M_{D^L}^L, D^L}, \phi_{M_{D^L}^L, D^L}, \alpha_{M_{D^L}^L, D^L})]$ 

```

ePPR backward stepwise procedure. The ePPR backward stepwise procedure builds a list containing models having between one and $\sum_{d=0}^{D^L} M_d^L$ terms. The procedure is described in Listing 8. It is similar to the same procedure in PPR (Listing 3). The input model for this procedure is the output of the ePPR FORWARD_STEPWISE procedure. The procedure operates iteratively. Suppose that at the beginning of one iteration model contains the M terms that best fit the response, then model is saved in the models list, the least important term is dropped from model, the residuals are adjusted according to the contribution of the dropped term, and the parameters of the dropped model are refitted to the adjusted residuals. Then, by the end of the iteration, model approximates a model with $M-1$ terms that best fits the response. In the next iteration model will be saved in the list of models

and a model with the best $M - 2$ terms fitting the response will be estimated. By the end of the loop, the list `models` will contain $\sum_{d=0}^{D_L} M_d^L$ models, where the M th model in this list will be the M -term model that best approximates the response.

Listing 8 ePPR: BACKWARD_STEPWISE

Require: `model`, r , x , λ , d

```

1: models  $\leftarrow$  [ ]
2: for  $m = \text{nTerms}(\text{model})$  downto 2 do
3:   models  $\leftarrow$  [models, model]
4:    $((\beta, \phi, \alpha), \text{model}) \leftarrow \text{DROP\_LEAST\_IMPORTANT\_TERM}(\text{model})$ 
5:    $r \leftarrow r + \beta\phi(\alpha^T x)$ 
6:   model  $\leftarrow \text{REFIT\_MODEL}(\text{model}, r, x, \lambda, d)$ 
7: end for
8: models  $\leftarrow$  [model, models]
9: return models

```

The ePPR `REFIT_MODEL` procedure is described in Listing 9. It is similar to the same procedure in PPR (Listing 4), but adds a few statements to refit terms using appropriately delayed images. The procedure begins by refitting the β parameters. Then it enters a loop where each term is removed from the model, the images are shifted according to the delay of the removed term, residuals are adjusted according to the contribution of the removed term, a new term is fitted to the adjusted residuals, this new term is added to the back of the model, and the residuals are adjusted according to the contribution of the new term.

Listing 9 ePPR: REFIT_MODEL

Require: `model`, r , x , λ , d

```

1: model  $\leftarrow \text{REFIT\_MODEL\_BETAS}(\text{model})$ 
2: noTerms  $\leftarrow \text{noTerms}(\text{model})$ 
3: for  $m \in 1$  to noTerms do
4:    $(\beta_m, \phi_m, \alpha_m, \text{delay}, \text{model}) \leftarrow \text{REMOVE\_FIRST\_TERM\_FROM\_MODEL}(\text{model})$ 
5:    $xs_i \leftarrow x_{i-\text{delay}}$ 
6:    $r \leftarrow r + \beta_m\phi_m(\alpha_m^T xs)$ 
7:    $(\beta, \phi, \alpha) \leftarrow \text{FIT\_NEW\_TERM}(r, xs, \alpha_m, \lambda, d)$ 
8:   model  $\leftarrow \text{ADD\_TERM\_TO\_BACK\_OF\_MODEL}((\beta, \phi, \alpha), \text{model})$ 
9:    $r \leftarrow r - \beta\phi(\alpha^T xs)$ 
10: end for
11: return model

```

ePPR fit new term procedure. To avoid estimates that overfit noise in the responses, the ePPR estimation algorithm penalizes estimates, \hat{y} in Equation 6, containing non-smooth projection directions. This is accomplished by adding penalty terms to

the PPR estimation criterion in Equation 2, as shown in Equation 7. To account for these penalty terms the objective function used in PPR to estimate a new term, Equation 3, is expanded to that used in ePPR, Equation 19. The ePPR new term procedure optimizes the criterion \mathcal{J} in this equation with respect to the projection α and the smooth function ϕ . This is performed iteratively. In each iteration α is first fixed and ϕ is adjusted to minimize \mathcal{J} . Then ϕ is fixed and α is adjusted to minimize \mathcal{J} . This iteration is repeated until the reduction in \mathcal{J} for two consecutive iterations falls below a convergence value. Finally, β is calculated and ϕ is normalized. The procedure is described algorithmically in Listing 10.

$$\mathcal{J}(\alpha, \phi) = \sum_{i=1}^N (r_i - \phi(\alpha^T x_i))^2 + \lambda \|L\alpha\|^2 \quad (19)$$

Note that, when α is fixed, the second term in Equation 19 is a constant. So, to adjust ϕ , with α fixed, it suffices to minimize the first term in Equation 19. This is done in the same way as in PPR. The procedure projects α onto the input images, $p_i = \alpha^T x_i$, builds a scatter plot with the pairs of projections and residuals, (p_i, r_i) , and sets ϕ as the smoothing spline that best approximates, the points in the scatter plot. The degrees of freedom d , controls the smoothness of the estimated splines. Appendix B.2 describes the procedure we used to select this parameter.

With ϕ fixed, Equation 19 can be expressed as the L^2 norm of a nonlinear function \tilde{h} depending on α , as shown in Equation 20. Then, finding the value of α that minimize Equation 20 reduces to a nonlinear least-squares problem. In ePPR this problem is solved using a Trust Region method (Nocedal and Wright 2006).

Listing 10 ePPR: FIT_NEW_TERM

Require: r, x, α, λ, d

1: **repeat**

2: Fix α and find ϕ_{k+1} minimizing $\tilde{\mathcal{J}}(\alpha, \phi) = \sum_{i=1}^N (r_i - \phi(\alpha^T x_i))^2$ {Smooth the scatterplot $(\alpha^T x_i, r_i)$ using a smoothing spline with d degrees of freedom}

3: Fix ϕ_{k+1} and update α along a Trust Region descent direction δ_k of $\mathcal{J}(\alpha, \phi_{k+1}) = \sum_{i=1}^N (r_i - \phi(\alpha^T x_i))^2 + \lambda \|L\alpha\|^2 : \alpha_{k+1} = \alpha_k + \delta_k$.

4: **until** \mathcal{J} stops decreasing

5: $\beta \leftarrow \sqrt{E\{\phi^2(\alpha^T x)\}}$

6: $\phi \leftarrow \phi/\beta$

7: **return** (β, ϕ, α)

$$\mathcal{J}(\alpha, \phi) = \sum_{i=1}^N (r_i - \phi(\alpha^T x_i))^2 + \lambda \|L\alpha\|^2 = \sum_{i=1}^N |h_i(\alpha)|^2 + \sum_{j=1}^m |\sqrt{\lambda} L\alpha_j|^2 = \|\tilde{h}(\alpha)\|_2^2 \quad (20)$$

where $h_i(\alpha) = r_i - \phi(\alpha^T x_i)$ and $\tilde{h}(\alpha) = (h_1(\alpha), \dots, h_N(\alpha), \sqrt{\lambda}(L\alpha)_1, \dots, \sqrt{\lambda}(L\alpha)_m)$

As in PPR, α_k is not updated until $\|\hat{h}(\alpha_k)\|_2^2$ converges to a minimum. Instead, α_k is updated only once and then the algorithm proceeds with the next iteration.

ePPR select best model procedure. The backward-stepwise procedure (Listing 8) returns a set of models having between one and $\sum_{d=0}^{D^L} M_d^L$ terms. The final ePPR estimate is selected from this set. Several strategies have been proposed for selecting the best model from a set of candidate models (Burnham and Anderson 2002). Here we use cross-validation, as described in Section “Methods: ePPR model selection procedure”.

Selection of ePPR hyperparameters

Below we describe the ePPR hyperparameters, and the procedure we used to select their values. The values used to estimate the models in this paper are shown in Section “Methods: ePPR hyper-parameters”.

Number of delays (D^L) and number of terms per delay (M_d^L) for the forward model. As described above, ePPR builds a forward model with a maximum delay of D^L and with M_d^L terms at each delay d , $0 \leq d \leq D^L$. We set the parameters D^L and M_d^L so that the forward model contained enough terms per delay, and enough delays, to characterize the responses of the cell, as indicated below.

Starting from the forward model, the backward stepwise procedure returns a list of models, where the m th model in this list is the ePPR model with m terms that best predicts the responses of the cell. Then, the model selection procedure selects from this list the final ePPR model as that model with the minimum number of terms that best predicts the responses of the cell. We set the values of D^L and M_d^L large enough so that the model chosen by the model selection procedure contains at most $M_d^L - 1$ terms at every delay d and the maximum delay is at most $D^L - 1$. This guarantees that at least 1 term at every delay, and all the terms at delay D^L , were irrelevant for the model maximizing predictive power. Thus, the forward model contained more terms per delay, and more delays, than were needed by the model maximizing predictive power.

Regularization parameters for the filters (λ). To avoid estimates overfitting noise in the responses we penalized estimates having non-smooth projection directions, by adding penalty terms to the ePPR optimization criterion in Equation 7. Each of these penalty terms has the form $\lambda \|L\alpha_{m,d}\|^2$. We chose λ by cross validation. We estimated ePPR models with different values of λ and chose the value of λ maximizing predictions to data not used in fitting the model parameters.

Degrees of freedom for the splines (d). The smoothness of the splines used to fit the nonlinear functions $\phi_{m,d}$ is controlled by the degrees of freedom parameter, d . The standard procedure is to estimate the value of this parameter from training data using standard or generalized cross validation (Green and Silverman 1994). However, for the large levels of noise in neural responses, these methods performed poorly. Better results were obtained by choosing d from validation data using cross-validation. For the models estimated with the reference level of noise for the simulated cell, and for the models estimated with the largest amount of data for the cortical cells, we verified that setting $d = 5$ provided reasonable results. So, to reduce

computations, for all estimations we fixed the degrees of freedom for the splines to $d=5$.

Convergence values for fitting new terms (addTermsCV and refitCV). As described above, ePPR fits a new term by repeatedly minimizing the criterion J in Equation 19, first with respect to α and then with respect to ϕ . These minimizations continue until the reduction in J for two consecutive iterations falls below a convergence value. The ePPR algorithm provides two convergence value hyperparameters, addTermsCV and refitCV, the former is used when adding a new term to the forward model, and the latter is used when refitting an ePPR term. For all the models estimated in this article we used addTermsCV = 0.01 and refitCV = 0.001.

Trust region hyperparameters (r0, rmax, and iterlim). At each iteration, the Trust Region method minimizes a target function on a restricted ‘trust’ region where the function behaves well. The size of this trust region is changed adaptively through the minimization (Nocedal and Wright 2006). The r0 and rmax hyperparameters give the initial and maximal size of the trust region. The hyperparameters iterlim controls the maximum number of iterations in the Trust Region method. For all the models estimated in this article we used r0 = 1, rmax = 1,000, and iterlim = 1,000.

Appendix C: Proofs

Proposition 1: *Let $x \in \mathbb{R}^p$ and $f(x)$ be a polynomial. Then*

$$f(x) = \bar{y} + \sum_{m=1}^{M_0} \beta_m \phi_m(\alpha_m^T x) \quad (21)$$

for constants \bar{y} , β_m , vectors $\alpha_m \in \mathbb{R}^p$, and univariate functions ϕ_m .

Proof: Any polynomial $f(x)$ can be written in the form $f(x) = \sum_{i=0}^n f_m(x)$ where $f_m(x)$ is an homogeneous polynomial of degree m^{AI} . Proposition 2 proves that any homogeneous polynomial $f_m(x)$ can be written in the form of Equation 21. Then, because the sum of expressions of the form of Equation 21 is another expression of the form of Equation 21, the proof is complete. \square

Proposition 2 is a more detailed and simplified version of a result by Diaconis and Sahshahani (1984, Proposition 1). We provide this proof here for completeness, and because it is very elegant. Our contribution to the result from Diaconis and Sahshahani (1984) appears in Lemma 2, that replaces the algebraic independence argument, at the end of the original proof, by a simpler argument using only concepts from linear algebra.

Proposition 2: *Let $r = \binom{m+p-1}{m}$ be the number of distinct monomials of degree m . Then there exist r distinct directions a^1, a^2, \dots, a^r in \mathbb{R}^p such that any homogeneous polynomial f of degree m can be written as $f(x) = \sum_{j=1}^r \alpha_j (x^T a^j)^m$ for some real numbers α_j .*

Proof: The space of homogeneous polynomial of degree m is an r -dimensional vector space over the real numbers. Due to this dimensionality, to prove the proposition, it suffices to show that we can find r directions $\{a^j\}_{j=1}^r$, such that the polynomials $\{(x^T a^j)^m\}_{j=1}^r$ are linearly independent.

Let $m_i(x)$, $1 \leq i \leq r$, be an enumeration of all the monomials in homogeneous polynomials of degree m . For each monomial $m_i(x)$, let D_i be the associated differential operator (e.g., if $m_i(x) = x_1^2 x_2 x_3$, $D_i = \partial^4 / \partial^2 x_1 \partial x_2 \partial x_3$). Lemma 1 proves that $D_i(x^T a^j)^m = m! m_i(a^j)$.

Suppose that for any $\{a^j\}_{j=1}^r$ the polynomials $\{(x^T a^j)^m\}_{j=1}^r$ were linearly dependent. This happens if and only if, for any $\{a^j\}_{j=1}^r$, the equation $\sum_{j=1}^r c_j (x^T a^j)^m = 0$ admits a non-trivial solution. Then, applying D_i to both sides of the previous equality, we get that, for any $\{a^j\}_{j=1}^r$, the system of equations $\sum_{j=1}^r m_i(a^j) c_j = 0$, $1 \leq i \leq r$, admits a non-trivial solution. Then, for any $\{a^j\}_{j=1}^r$, the determinant of the matrix associated with this system of equations must be zero; i.e., $\det(A) = 0$, with $A_{i,j} = m_i(a_j)$. As a function of the coefficients of the directions, $\{a^j\}$, $1 \leq j \leq r$, $1 \leq l \leq p$, $\det(A)$ is a non-zero polynomial with coefficients in the field of the rationals. Thus, the evaluation of the non-zero polynomial $\det(A)$ should equal zero for any possible assignment of directions $\{a^j\}_{j=1}^r$. But this is not possible according to the contrapositive of Lemma 2^{AII}. This contradiction arose because we supposed that that for any $\{a^j\}_{j=1}^r$ the polynomials $\{(x^T a^j)^m\}_{j=1}^r$ were linearly dependent. Therefore, there must exist a set of directions $\{a^j\}_{j=1}^r$ such that the polynomials $\{(x^T a^j)^m\}_{j=1}^r$ are linearly independent, and the proposition is proved. \square

Lemma 1: Let $x \in \mathbb{R}^p$ and $m_i(x) = x_1^{n_1} x_2^{n_2} \dots x_p^{n_p}$, with $m = \sum_{j=1}^p n_j$. Associate with $m_i(x)$ the differential operator $D_i = \frac{\partial^m}{\partial x_1^{n_1} \partial x_2^{n_2} \dots \partial x_p^{n_p}}$. Then $D_i(x^T a^j)^m = m! m_i(a^j)$.

Proof: By the multinomial theorem

$$(x^T a^j)^m = \sum_{k_1, k_2, \dots, k_p} \frac{m!}{k_1! k_2! \dots k_p!} (x_1 a_1^j)^{k_1} (x_2 a_2^j)^{k_2} \dots (x_p a_p^j)^{k_p} \tag{22}$$

where the summation is taken over all non-negative integers k_1 through k_p such that $\sum_{j=1}^p k_j = m$. Applying D_i to both sides of Equation 22 we obtain

$$D_i(x^T a^j)^m = \sum_{k_1, k_2, \dots, k_p} m! (a_1^j)^{k_1} (a_2^j)^{k_2} \dots (a_p^j)^{k_p} D_i \frac{x_1^{k_1} x_2^{k_2} \dots x_p^{k_p}}{k_1! k_2! \dots k_p!}$$

But

$$D_i \frac{x_1^{k_1} x_2^{k_2} \dots x_p^{k_p}}{k_1! k_2! \dots k_p!} = \begin{cases} 1 & \text{if } k_1 = n_1, k_2 = n_2, \dots, k_p = n_p \\ 0 & \text{otherwise} \end{cases}$$

So

$$D_i(x^T a^j)^m = m! (a_1^j)^{n_1} (a_2^j)^{n_2} \dots (a_p^j)^{n_p} = m! m_i(a^j)$$

\square

Lemma 2: Let k be an infinite field, and F a polynomial with n variables and coefficients in k . If $F(b_1, b_2, \dots, b_n) = 0$, for all $\{b_j\}_{j=1}^n \subset \mathbb{R}$; then $F = 0$.

Proof: By induction on n .

$n = 1$: any non-zero and one-dimensional polynomial, with coefficients on an infinite field, has a finite number of roots. Then, if $F(b_1) = 0$ for all $b_1 \in \mathbb{R}$, F is a one-dimensional polynomial with an infinite number of roots. Thus, F must be the zero polynomial.

$n \rightarrow n + 1$: if F is a polynomial of degree m with coefficients in k and variables x_1, \dots, x_n, x_{n+1} , then we can write $F = \sum_{i=0}^m F_i x_{n+1}^i$, where F_i is a polynomial with coefficients in k and variables x_1, \dots, x_n . For example, if $n = 1$ and $m = 2$, $F = c_0 + c_{11}x_1 + c_{12}x_2 + c_{211}x_1^2 + c_{212}x_1x_2 + c_{222}x_2^2 = (c_0 + c_{11}x_1 + c_{211}x_1^2) + (c_{12} + c_{212}x_1)x_2 + c_{222}x_2^2 = F_0 + F_1x_2 + F_2x_2^2$, with $F_0 = c_0 + c_{11}x_1 + c_{211}x_1^2$, $F_1 = c_{12} + c_{212}x_1$, and $F_2 = c_{222}$.

Suppose $F(b_1, b_2, \dots, b_{n+1}) = 0$, for all $b_1, b_2, \dots, b_{n+1} \in \mathbb{R}$. Take any $\tilde{b}_1, \tilde{b}_2, \dots, \tilde{b}_n \in \mathbb{R}$ and define $G(x_{n+1}) = F(\tilde{b}_1, \tilde{b}_2, \dots, \tilde{b}_n, x_{n+1})$. Then, by hypothesis, $G(b_{n+1}) = F(\tilde{b}_1, \tilde{b}_2, \dots, \tilde{b}_n, b_{n+1}) = 0$ for all $b_{n+1} \in \mathbb{R}$. So, by the inductive hypothesis, $G = 0$, which means $F_i(\tilde{b}_1, \tilde{b}_2, \dots, \tilde{b}_n) = 0$, for $0 \leq i \leq m$. Because $\tilde{b}_1, \tilde{b}_2, \dots, \tilde{b}_n$ are arbitrary elements in \mathbb{R} , we have that $F_i(b_1, b_2, \dots, b_n) = 0$, for $0 \leq i \leq m$ and for any $b_1, b_2, \dots, b_n \in \mathbb{R}$. So by the inductive hypothesis $F_i = 0$, $0 \leq i \leq m$. Thus, $F = \sum_{i=0}^m F_i x_{n+1}^i = 0$. \square

Appendix D: ePPR models without time interactions

Figures 11a and b show the parameters of an example ePPR model without time interactions estimated from the same data as in Figures 2a and b. Nonlinear interactions between pixels of images at different delays are relevant to characterize the responses of the simulated complex cell, but these nonlinear interactions cannot be accounted by ePPR models without time interactions. Despite this, the estimated filters (Figure 11a) well approximate the true filters of the simulated model (Figure 2a). Also, the estimated nonlinear functions (Figure 11b) correctly recovered the facilitatory/suppressive nature of the associated filters. Figure 11c plots the predictions of ePPR models with (red curve) and without (pink curve) time interactions. For comparison, it re-plots, from Figure 4, the predictions of second-order multi-dimensional polynomials using ePPR filters with time interactions (orange curve). At 5.62 and 0.56 spikes/image predictions of the ePPR models with time interactions were significantly better than those of the polynomial models (Wilcoxon signed-rank test, $p < 0.01$), and at all noise levels predictions of the ePPR model with time interactions were significantly better than those ePPR models without time interactions ($p < 0.01$).

Above we proved that there exists an ePPR model that can approximate, to an arbitrary degree of precision, any continuous function with inputs in $[0, 1]^p$, as the simulated cell. However, Figure 11c shows that, even at the lowest noise level (5.62 spikes/image), ePPR models do not perfectly approximate the responses of the simulated cell. This problem could be an example where the ePPR estimation algorithm did not converge to its optimal approximation, as noted in Section “Extended Projection Pursuit Regression”.

The parameters of an example ePPR model without time interactions estimated from responses to natural stimuli of the cortical complex cell are shown in Figures 12a and b. At each delay, the filters without time interactions match well the

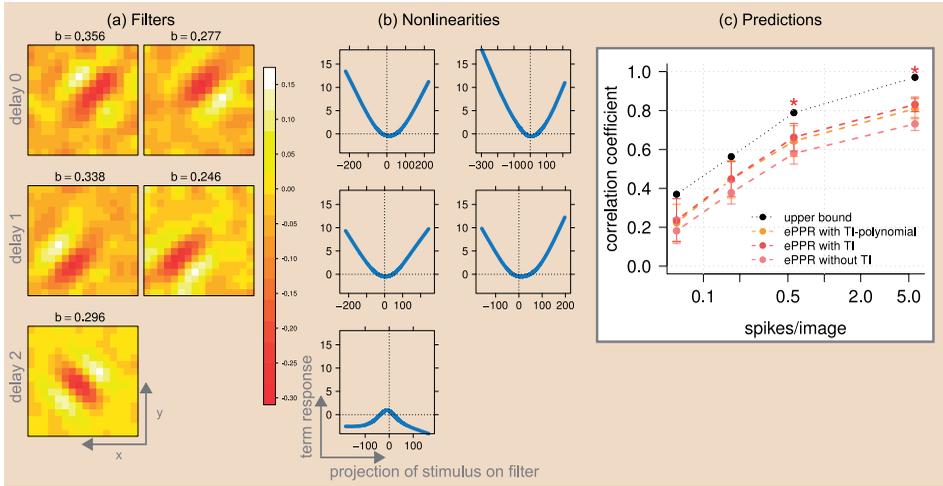


Figure 11. Simulated cell: ePPR models without time interactions estimated from responses to natural stimuli. (a, b): filters (a) and nonlinear functions (b) of an example model estimated from responses to natural stimuli with the reference noise level (0.56 spikes/image, as in Figure 2). The titles in (a) are the corresponding β coefficients. (c) predictive power of ePPR models with and without time interactions compared to that of a polynomial model. Orange curve: predictions for a second-order multi-dimensional polynomial constructed with ePPR filters with time interactions (re-plotted from Figure 4a). Light red curve: predictions from ePPR models with time interactions. Pink curve: predictions from ePPR models without time interactions. Black curve: upper bound on correlation coefficients. Light red asterisks mark number of spikes/image at which predictions of the ePPR models with time interactions were significantly better than those of the polynomial models. Despite the mismatch between the simulated model in Equation 8, that incorporates time interactions between pixels of images at different delays, and the ePPR model without time interactions, that cannot model these interactions, the estimated filters (Figure 11a) well approximate the true filters (Figure 2a), and the estimated nonlinear functions correctly recovered the facilitatory/suppressive nature of the associated filters.

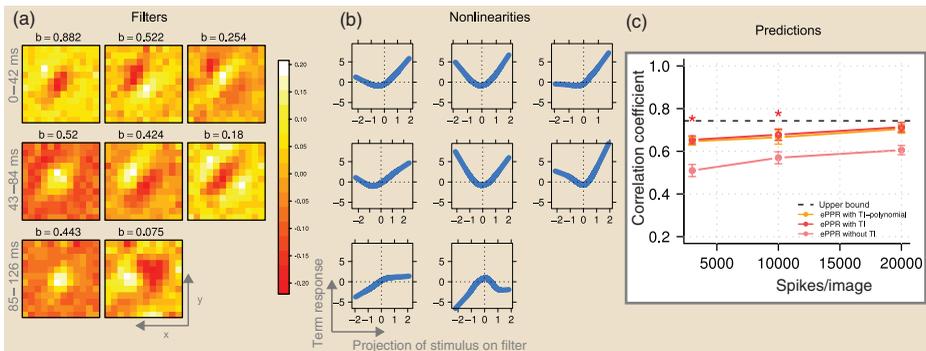


Figure 12. Complex cell: ePPR models without time interactions estimated from responses to natural stimuli. Same format as Figure 11. The filters and nonlinear function of this model are similar to those of the model with time interactions (Figure 6). The ePPR model with time interactions predicts significantly better than the ePPR without time interactions, demonstrating the relevance of nonlinear interactions between pixels of images at different delay for the response of this complex cell.

corresponding frames of the filters with time interactions (Figure 6a). For instance, the filters at delay 0–42 ms of the model without time interactions (first row in Figure 12a), match well the first frames of the three most important filters of the model with time interactions (first frames of the three leftmost filters in Figure 6a)^{AIII}. The nonlinear functions of the model without time interactions correspond well with those of the model with time interactions. For example, all the nonlinear functions at delays zero and one of the model without time interactions are facilitatory, and those at delay two are suppressive (Figure 12b). In agreement, the nonlinear functions of the four most important terms of the model with time interactions, whose filters have most structure at delays zero and one, are facilitatory, and the nonlinear function of the least important term, whose filter has most structure at delay two, is suppressive (Figure 6b). As for the simulated cell, Figure 12c plots the predictions of ePPR models with (red curve) and without (pink curve) time interactions, as well as those of a polynomial model using the filters of ePPR models with time interactions (orange curve, re-plotted from Figure 4a). ePPR models with time interactions predict significantly better than ePPR models without time interactions (red curve above pink curve). This shows that nonlinear interactions between pixels of images at different delays are relevant to predict the responses of this complex cell. Also, ePPR models with time interactions yield better or equal predictions than polynomial models.

The ePPR model without time interactions estimated from responses of the simulated LNL cell is shown in Section “Simulated cell: Linear-nonlinear-linear model”, and that estimated from responses of the simulated complex cell with temporally correlated inputs is shown in Appendix F.

Appendix E: Varying the amount of divisive inhibition

Due to the divisive normalization, the simulated model in Equation 8 cannot be represented exactly by the ePPR model in Equation 6. However, as shown in Figure 2, ePPR produced good approximations. To check if this positive result only holds for the particular amount of inhibition used in our simulations, we estimated ePPR models from responses with different amounts of inhibition.

We call the denominator in Equation 8 the inhibitory factor. The solid line in Figure 13a, re-plotted from Figure 2d, shows the principal angles between the true filters and those of ePPR models estimated from simulated responses with a mean inhibitory factor of 4.26; i.e., where the denominator in Equation 8 reduced the numerator, on the average, 4.26 times. The dashed and dotted lines plot the principal angles for ePPR models estimated from simulated responses with a mean inhibitory factor of 20.58 and 40.17, respectively.

When the mean inhibitory factor is increased by a factor of five, from 4.26 to 20.58, only the first principal angle increases marginally. And when the mean inhibitory factor is increased by a factor of ten, from 4.26 to 40.17, the third principal angle also increases substantially. This happens because two of the five ePPR models estimated from responses with a mean inhibitory factor of 40.17 did not recover the inhibitory filter^{AIV}. To visualize the impact of increasing the amount

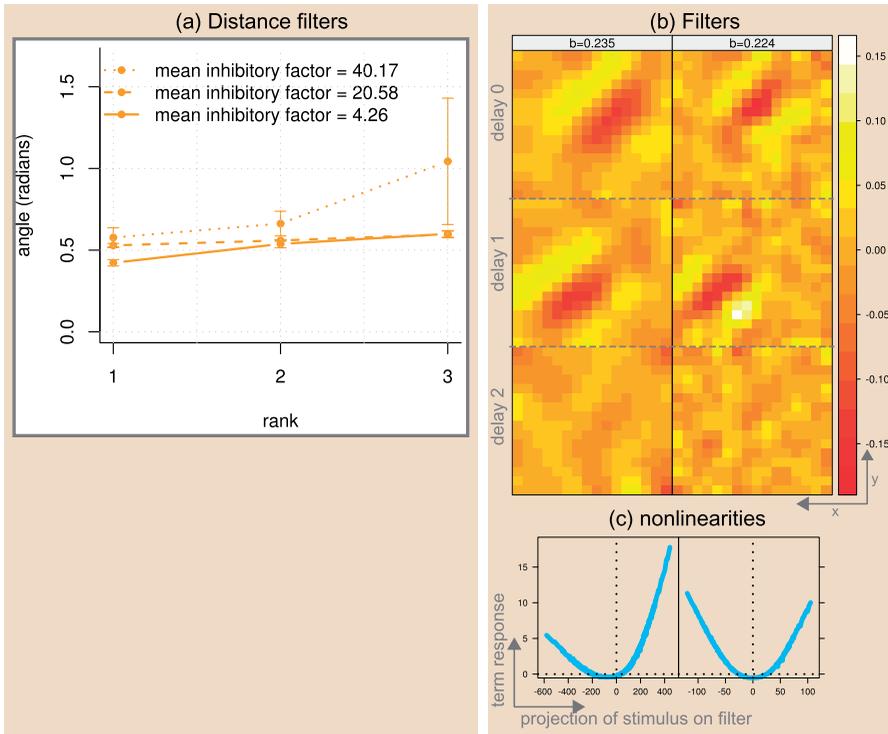


Figure 13. Varying the amount of divisive inhibition. (a): principal angles between the true filters of the simulated model (Figure 2a) and those of ePPR models with time interactions estimated from responses with varying amount of inhibition. (b, c): filters (b) and nonlinear functions (c) of the ePPR model estimated from responses with the largest inhibition and whose filters were most different from the true filters. Even though ePPR models cannot represent exactly the simulated model with divisive inhibition, they yielded good approximations for a broad range of inhibition strengths.

of inhibition on ePPR estimates, Figure 13b and c show the filters and nonlinear functions of the model estimated from responses with the largest inhibition and whose filters were most different from the true filters (according to their principal angles). These estimates capture the most important features of the true excitatory filters in Figure 2a.

Note that, as the mean inhibitory factor increased from 4.26 to 20.58 to 40.17, the mean of the simulated responses (Equation 8) decreased from 0.56 to 0.33 to 0.26 spikes per frame. Then, because the noise is Poisson, as inhibition increased the signal to noise ratio in the responses decreased. Hence, the degradation in the quality of the ePPR estimates as the strength of the inhibition is increased does not only reflect the mismatch between the simulated and ePPR models, but is also due to an increasing noise level.

Overall, Figure 13 shows that, although ePPR models cannot represent exactly the simulated model with divisive inhibition, they yield good approximations for a large range of inhibition strengths.

Appendix F: Temporally correlated inputs

The simulated models and the real simple and complex cells were probed with stimuli with natural spatial statistics, but that were temporally uncorrelated. To study the effect of temporal correlations on ePPR, we simulated responses using the model in Equation 8 to the Natural movie ensemble (see “Methods: Stimulus ensembles” and “Methods: Simulated responses”). Figures 14a and b show the filters and nonlinear functions of an estimated ePPR model with time interactions. The filters are very similar to the true filters of the model (Figure 2a), and the nonlinear functions correctly indicate the excitatory/inhibitory nature of the corresponding filters. For comparison, Figure 14c shows that MID recovered good estimates of the excitatory filters, but, as in all MID estimates of the simulated cell, it failed to recover the inhibitory filter. Also, the predictive power of a second-

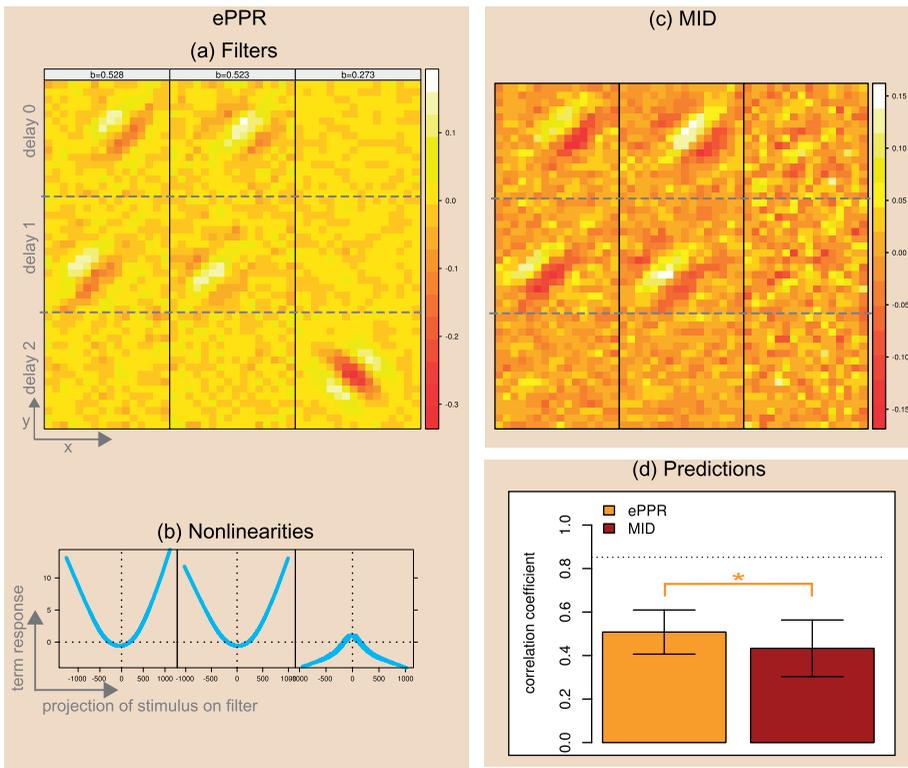


Figure 14. Temporally correlated inputs. (a, b): filters (a) and nonlinear functions (b) of an ePPR model with time interactions. (c): MID filters. (d): correlation coefficient between second-order polynomial models predictions and simulated cell responses; the orange and red bars correspond to polynomials constructed with ePPR and MID filters, respectively, and the black dotted line is an upper bound on the predictive power of any model. The ePPR filters are very similar to the true filters of the model, and the nonlinear functions correctly indicate the excitatory/inhibitory nature of the corresponding filters. MID recovered good estimates of the excitatory filters, but failed to recover the inhibitory filter. The predictive power of a second-order polynomial using ePPR filters was significantly better than that of a second-order polynomial using MID filters.

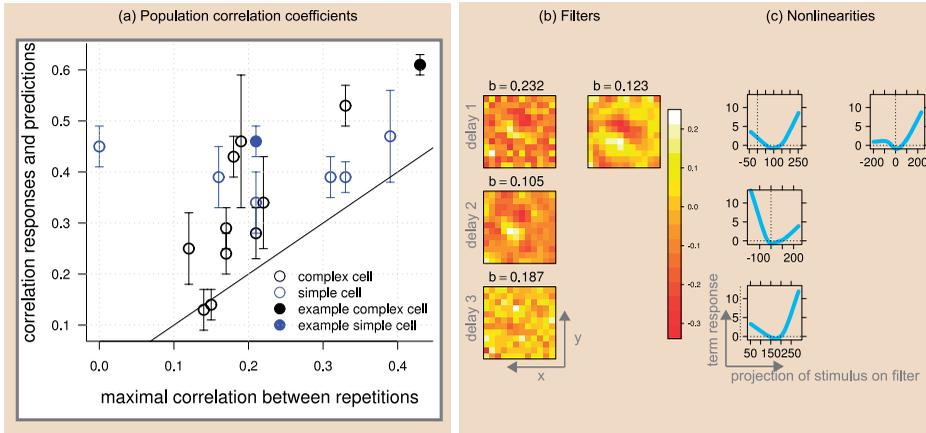


Figure 15. Population results: (a): correlation coefficients between responses of cells to natural stimuli and predictions from ePPR models without time interactions, as a function of the maximal correlation coefficient between pairs of responses to repetitions of the stimuli. (b, c): filters (b) and nonlinear functions (c) of the ePPR model without time interactions achieving the lowest correlation coefficient. Black circles: complex cells. Blue circles: simple cells. Solid circles: example cells shown in this article. Only cells for which the maximal correlation coefficient between pairs of responses was greater than 0.1 are shown. The example complex cell is the one for which we obtained best correlation coefficients, but similar correlation coefficients, and qualitatively similar ePPR estimates, were obtained for other complex cells. For example, the filters and nonlinear functions of the ePPR model achieving the lowest correlation coefficient are qualitatively similar to those of the ePPR model of the example complex cell (Figures 12a and b).

order polynomial using ePPR filters was significantly better than that of a second-order polynomial using MID filters ($p < 0.01$, Figure 14d).

In a previous evaluation we simulated response of the model in Equation 8 to a natural movie with a very high correlation between adjacent frames (the un-shuffled *Natural sequence* described in Section “Methods: Stimulus ensembles”). For these responses, MID filters were better than ePPR ones. Thus, to characterize cells from their responses to stimuli with very high temporal correlations, MID might be preferable to ePPR. However, due to eye and head movements, cells in the visual system are not normally exposed to stimuli with the very high temporal correlations of the previously used natural movie. The movie used in Figure 14 (Natural movie ensemble, Section “Methods: Stimulus ensembles”), containing realistic head movements, is a better approximation to the inputs that stimulate cells in natural environments.

Appendix G: Population plot

Figure 15a plots the correlation coefficients between responses of a cell to natural stimuli and predictions of ePPR models without time interactions, as a function of the maximal correlation coefficient between pairs of responses to repetitions of the stimuli^{AV}. Only cells for which the maximal correlation coefficient between pairs of

responses was greater than 0.1 are shown. From these cells, the example complex cell shown in this article is the one for which we obtained best correlation coefficients (filled black circle), but similar correlation coefficients, and qualitatively similar ePPR estimates, were obtained for other complex cells. For example, Figure 15b and c show the filters and nonlinear functions, respectively, of ePPR model achieving the lowest correlation coefficient, illustrating that, even in this case, ePPR models of complex cells were qualitatively similar to that of the example complex cell.

The maximal correlation coefficient between pairs of responses of a cell to repetitions of the stimuli is inversely proportional to the noise level in the responses of the cell. As expected, Figure 15 shows that the predictive power of ePPR models increases as the noise level in the responses of cells decreases.

The solid line shows the points where the correlation between responses of cells and predictions from models equal the maximal correlation between repetitions. Because we used the mean response to several repetitions of the stimuli as the response of the cells, the correlation coefficients between responses of cells and predictions from ePPR models can be, and in almost all cases is, larger than the maximal correlation between individual repetitions.

Notes

- [AI] An homogeneous polynomial of degree m is a polynomial whose monomials with nonzero coefficients all have the same total degree m . For example, $x_1^7 + x_1x_2^4x_3^2 + x_2^2x_3^5$ is an homogeneous polynomial of degree $m=7$; the sum of the exponents in each term is always seven. (page 82)
- [AII] The contrapositive of Lemma 2 says that if F is a non-zero polynomial, with n variables and coefficients in a infinite field k , then exist $\{b_j\}_{j=1}^n \subset \mathbb{R}$ such that $F(b_1, b_2, \dots, b_n)$ is not zero. Then, because $\det(A)$ is a non-zero polynomial with rp variables and coefficients in the infinite field of the rationals, there must exists $\{a_l^j\} \subset \mathbb{R}$, $1 \leq j \leq r$, $1 \leq l \leq p$ such that $\det(A) \neq 0$. (page 83)
- [AIII] However, this match is not perfect, at delay 43–84ms the ePPR model with time interactions recovered four filters with good structure (Figure 6a), while the model without time interactions recovered three filters (Figure 12a, Appendix D). (page 86)
- [AIV] The ePPR estimation algorithm returns a collection of models having between one term and the number of terms of the forward model (Appendix B). From this collection we select the optimal model using a cross-validation procedure. Because in order to compute three principal angles we need three estimated filters, for the two ePPR estimations where the optimal model contained two terms, we used the suboptimal model with three terms returned by the ePPR algorithm to compute the three principal angles. (page 86)
- [AV] The simple cell plotted as having a maximal correlation between repetition of zero was probed with only one repetition of the stimuli, so we could not compute the maximal correlation coefficient between repetitions. (page 89)