# Optimal signal representation in neural spiking population codes as a model for the formation of simple cell receptive fields.

Laurent Perrinet

# Optimal signal representation in neural spiking population codes as a model for the formation of simple cell receptive fields

Laurent U. Perrinet
Institut de Neurosciences Cognitives de la Méditerranée (INCM)
CNRS / University of Provence
13402 Marseille Cedex 20, France
e-mail: Laurent.Perrinet@incm.cnrs-mrs.fr

September 19, 2008

**Abstract**

The primary visual cortex is the central hub for the transmission of visual information to the rest of the central nervous system. We study here models explaining how this neural system becomes efficient through natural selection and neural development. By defining and then optimizing an efficiency cost, we may derive an adaptive model of the input to the primary visual cortex as an unsupervised learning algorithm. As an alternative to classical models, we focus here on the fact that visual information is carried from the sensory organs to the primary visual cortex by neuronal events, or spikes, in bundles of parallel fibers. In fact, taking advantage of the constraint that spikes may be considered as all-or-none binary events, we may build a generic cost for the efficiency of the visual representation as a measure of the $L_0$ norm sparseness of the code. However, this is a "hard" NP-complete problem and we propose a solution in a population of generic Integrate-and-Fire neurons. It relies both on a correlation-based inhibition using lateral interactions and on an homeostatic constraint by a spiking gain control mechanism, two key features of cortical processing. For comparison purposes, we applied this scheme to the learning of small patches taken from natural images and compared the results and efficiency with state-of-the-art algorithms. Results show that while the different coding algorithms gave similar efficiencies, the homeostasis provided an optimal balance which was crucial during the learning. This study provides a simpler yet more efficient algorithm for learning that is particularly well adapted to the architecture of neural computations. By providing the optimally independent components in a set of inputs, it suggests that this Sparse Spike Coding strategy may provide a generic computational module.

## Keywords

Neural population coding, Spike-event computation, Statistics of natural images, Independent Component Analysis, Sparse Spike Coding, Unsupervised learning, Sparse-Hebbian Learning

# 1 Introduction

The neural architecture on which our cognitive abilities are based is a dynamical, adaptive system which evolves to provide optimal solutions in our interactions with the environment. Vision is a useful illustration of these properties and in particular, models for the formation of orientation selective simple cell receptive fields in the primary visual cortex (V1) have attracted great attention as generic models of coding and learning in neural computations. Viewed from a functional approach, the system evolved so that relevant sensory information is transformed efficiently so as to allow in the end optimal decision making (Atick, 1992; Barlow, 2001). The most accepted explanation for the formation of these cells in V1 is that it optimizes the efficiency of the neural representation over images drawn from natural scenes, that is from behaviorally relevant scenes. This may be formalized as representing at best visual information while increasing the sparseness of the representation (Olshausen and Field, 1996). Similar approaches have been followed for natural images (Doi et al., 2007; Fyfe and Baddeley, 1995; Hamker and Wiltschut, 2007; Perrinet, 2004a; Rehn and Sommer, 2007; Zibulevsky and Pearlmutter, 2001) and sounds (Lewicki and Sejnowski, 2000; Smith and Lewicki, 2006) that were based on solving the inverse of a generative model of the signal. These correspond more generally to a coding part on a short time scale and a learning part on a longer time scale which corresponds to finding the independent components in the natural scenes (Bell and Sejnowski, 1997). However, all of these solutions relied on specific parameterizations and didn't explicitly demonstrated how their algorithm could be specifically adapted to the nature of neural computations. For instance, the coding was achieved by conjugate gradient (Olshausen and Field, 1996) or orthogonal matching pursuit (Rehn and Sommer, 2007) without explicitly addressing the problem of the existing representational constraints in the cortex. More specifically, they don't specifically take advantage of the parallel, dynamical and adaptive nature and architecture of neural computations that make them different from the ones occurring in a traditional sequential computer.

In that direction, a key aspect of neural computations is that most information between neurons is carried by *spikes*. Spikes (or Action Potentials) are simple pulses of the membrane potential whose shape seems to carry few information and which may travel robustly over long distances on axons[1]. In the human early visual system for instance, after flashing a visual stimulus, such as after

[1] Spikes have a shape of approximately 1 ms and are also present on dendrites since their presence is linked to the dynamical properties of the active ion channels on the neuron's membrane (Cessac and Samuelides, 2007). They are universally present in the central nervous system but also across species and phylogeny.

a saccade, a cascade of mechanisms will take place after the activation of the photoreceptors in the retina. A volley of spikes leaves the retina through the bundle of axons that forms the optic nerve to reach the lateral geniculate nuclei (after approximately 25 ms). There, a new processing takes place generating a new volley of spikes toward the primary visual cortex that is reached after approximately 35 ms (Bullier, 2001). The visual information that is "decoded" there is often considered to be "encoded" in the spikes' firing pattern of every fiber. As a consequence, neural computations are event-based and dynamical: information transfer is parallel while in classical solutions computations are sequential and non-interruptible. A goal of this work is to show how we may take advantage of spiking mechanisms to represent visual information in a dynamic, parallel and event-based fashion.
To achieve that agenda, we will first analytically formulate the problem of learning in the primary visual cortex. This mathematical part (Sec. 2) may be skipped at first by non-experts until Sec. 2.6 but is essential to understand quantitatively the results

1. We will first define the function of the neural coding in a statistical inference framework (see Sec. 2.1) and then define a possible spike coding algorithm of a flashed static image (see Sec. 2.2).

2. This will allow to quantify the efficiency of the spike coding in the neural assembly by introducing the $L_0$ norm as a measure of the sparseness of the spike code (see Sec. 2.3).

3. Based on previous results (Perrinet et al., 2002), we will define an efficient sparse spike coding and decoding scheme using correlation-based inhibition coupled with the spiking mechanism. Taking advantage of a biologically-inspired homeostatic spike gain control to ensure an optimal balance within the assembly, we will improve the performance of the previously proposed algorithm by optimizing the competition between neurons (see Sec. 2.4).

4. Finally, once these mechanisms are defined, we may easily derive an unsupervised learning algorithm (see Sec. 2.5) and finally derive a simple hebbian-type learning scheme on the sparse representation (see Sec. 2.6).

This means that to define the learning algorithm we will first have to define the framework, the coding and an efficiency cost. Then, we will compare the proposed algorithm with standard methods: SparseNet (Olshausen and Field, 1996) and Adaptive Matching Pursuit (Perrinet et al., 2003) and show the relative importance of the coding scheme and of the homeostasis on the resulting systems thanks to the quantitative measures of efficiency (see Sec. 3). We will conclude by comparing this method with previously proposed schemes and how these may be reconciled to improve our understanding of the neural code by drawing the link between structure (spikes in a distributed network) and function (efficient coding) and explore the significant parameters at work in these mechanisms.

# 2 Method : adaptive Sparse Spike Coding (aSSC)

## 2.1 Generative models and statistical inference

We saw that in low-level sensory areas, the goal of neural computations is to build efficient intermediate *representations* to allow efficient decision making (Barlow, 2001; Field, 1994). A "good" representation of the world should map at best the information from the physical signals which are relevant for the sensory area under study. Furthermore, it will be more efficient if it is easily transformable according to usual transforms. In visual areas for instance, any representation of a scene should be easily transformed for any translation or rotation of the scene[2], since these are common movements and that higher-level areas will need to take into account this information. As a consequence, it is easier to define first a synthesis model of the world and its transformations and then to build the representation by inverting this model. This synthesis model (also called the forward model) may be built using statistical observations or with prior assumptions on the physics of the generation of the signal. A *Linear Generative Model* (LGM) (Olshausen and Field, 1998) is a generic case where the signal may be thought as the linear combination of independent causes. Inverting the forward model corresponds in the terminology of signal processing to the coding process, since it transforms the signal (for instance the observed image) into a more abstract representation as a combination of components from the forward model (for instance the edges the image is formed from). This coding may then be used to understand the content of the signal relative to the (forward) synthesis model but also to validate on a longer term the coding algorithm solving the inverse problem. In fact, one strategy is to build learning processes which optimize the overall efficiency of the representations for a known coding algorithm. It is then expected that the comparison of different learning strategies will help us understand the processes underlying receptive field formation (here in the input layer 4 of V1) as a generic neural computation. For instance, some coding algorithms seem better than others and comparing their relative efficiency will highlight the reasons why some aspects of the neural architecture (parallel event-based computations, lateral interactions within the cortical area) were selected during evolution.

Formally, to define the LGM, we will use a "dictionary" of $N$ images represented by the matrix $\mathbf{A} = \{\mathbf{A}_j\}_{1 \leq j \leq N}$, each of these being defined by $\mathbf{A}_j = \{A_{ij}\}_{1 \leq i \leq M}$ over the set of sampling positions $i$ (that is the pixels in a simple image processing framework). Knowing $\mathbf{A}$ and the "sources" $\mathbf{s} = \{s_j\}_{1 \leq j \leq N}$, the signal $\mathbf{x} = \{x_i\}_{1 \leq i \leq M}$ is defined as

$$\mathbf{x} = \sum\nolimits_{1 \leq j \leq N} s_j.\mathbf{A}_j + \mathbf{n} = \mathbf{A}.\mathbf{s} + \mathbf{n} \tag{1}$$

where $\mathbf{n}$ is a decorrelated gaussian additive noise of variance $\sigma_n^2$. This noise model is achieved thanks to the preprocessing (which could be achieved in gen-

[2]We will restrict ourselves to static flashed images since they are behaviorally important for pattern recognition and because they will highlight the problem of the parallel spatial processing by populations of neurons.

eral by Principal Component Analysis) without loss of generality since the processing is invertible (Perrinet, 2004b) (see Fig. 5). The LGM is well adapted to natural scenes because transparency laws are linear for luminance values and thus the LGM describes well the synthesis in a local neighborhood of any natural image. The goal of any coding algorithm for the inverse problem is to find for an observed $\mathbf{x}$ the best set $\mathbf{s}$ of sources that generated the signal. Then, the goal of a learning algorithm is to adapt at best in the long term to the parameters of the LGM, that is to the matrix $\mathbf{A}$ and the statistics of $\mathbf{s}$. This dictionary $\mathbf{A}$ is possibly much larger than the dimension of the input space (that is when $N >> M$); the dictionary is then said to be *over-complete*. One advantage of over-complete dictionaries is that it's representational power is greater and that for instance if the dictionary is transform invariant then it is easy to build a transform invariant representation. On the other hand this leads to a combinatorial explosion for the inversion of the LGM and typically, there exist many solutions for one input. We will see in Sec. 2.3 how we may quantify the global efficiency of the coding and in Sec. 2.4 the solution that we propose, but let's first define how one may evaluate the likelihood of any source knowing an input $\mathbf{x}$.

In fact, having defined the forward model, we may now be interested in computing how well a particular instance of the signal (here an image) matches with the model. From (Perrinet, 2004b, 2007), we know that for a given signal $\mathbf{x}$, the log-probability $\log P(\{s_j\}|\mathbf{x},\mathbf{A})$ corresponding to a *single* source $s_j.\mathbf{A}_j$ for any $j$ is maximal for the projection coefficient defined by:

$$s_j^* =< \mathbf{x}, \frac{\mathbf{A}_j}{\|\mathbf{A}_j\|^2} > \stackrel{\text{def}}{=} \frac{\sum_{1\leq i\leq M} \mathbf{x}(i).\mathbf{A}_j(i)}{\sum_{1\leq i\leq M} \mathbf{A}_j(i)^2} \tag{2}$$

where $\stackrel{\text{def}}{=}$ means "equal by definition". This is based on the hypothesis that the signal is a realization of the LGM as it is defined in Eq. 1 and for which we assume no prior knowledge on the coefficients. The log-probability is then maximal globally for the source $j^*$ with maximal correlation coefficient $j^* = \text{ArgMax}_j \rho_j$ with

$$\begin{aligned} \rho_j &= < \frac{\mathbf{x}}{\|\mathbf{x}\|}, \frac{\mathbf{A}_j}{\|\mathbf{A}_j\|} > \\ &\stackrel{\text{def}}{=} \frac{\sum_{1\leq i\leq M} \mathbf{x}(i).\mathbf{A}_j(i)}{\sqrt{\sum_{1\leq i\leq M} \mathbf{A}_j(i)^2}.\sqrt{\sum_{1\leq i\leq M} \mathbf{x}(i)^2}} \end{aligned} \tag{3}$$

It should be noted that $\rho_j$ is the $M^{\text{th}}$-dimensional cosinus and that its absolute value is therefore bounded by 1. The value of $\text{ArcCos}(\rho_j)$ would therefore give the angle of $\mathbf{x}$ with the pattern $\mathbf{A}$ and in particular, the angle would be equal (modulo $2\pi$) to zero if and only if $\rho_j = 1$ (full correlation), $\pi$ if and only if $\rho_j = -1$ (full anti-correlation) and $\pm\pi/2$ if $\rho_j = 0$ (both vectors are orthogonal, there is no correlation). Also, the correlation coefficient is independent of the norm of the filters and we will assume without loss of generality in the rest that

these are normalized to unity (that is that $\forall j, \|\mathbf{A}_j\| = 1$).
In canonical models of neural modeling this corresponds to the linear dendritic integration over the receptive field, producing for a positive correlation a driving current leading to the hyper-polarization of the cell and possibly to spiking. This justifies the computation of the correlation in the perceptron model (Rosenblatt, 1960) as it provides a direct measure of the log-probability under the assumptions that we used (the LGM with Gaussian noise and uniform priors). Starting from this basic mechanism, one could compute for every signal a vector from the set of activities corresponding to how well the neurons corresponded to patterns in the image predefined in the weights matrices. This representation is certainly not explicit and could be implemented in various ways which are out of the scope of this paper (Ma et al., 2006). However, we will now explain how this information may be coded and decoded by a set of spiking neurons with a simple Linear/Non-Linear (L/N-L) architecture (Carandini et al., 1997, 2005).

## 2.2 Spike coding and decoding of a transient signal in a population of neurons

In fact, before defining a possible efficiency cost based on the LGM, it is necessary to define how information may be coded using the binary event-based representations underlying neural spiking activity. As a matter of fact, neurons are intrinsically dynamical systems and we will take advantage of this property to transform the signal into a volley of spikes. For the large class of Integrate-and-Fire neurons which is relevant for neurons in the input layer of the primary visual cortex, we may use the fact that the larger the driving excitation, the larger the firing frequency and dually the shorter the latency of spiking (Perrinet et al., 2004). More precisely, let's consider a population of $N$ neurons as an information channel for which we wish to code and then decode a vector $\mathbf{s} = \{s_j\}_{1 \leq j \leq N}$ only by transmitting spikes. If for instance we write $s_j = |\rho_j|$, for any $j$ of this vector, higher values will represent more salient features than lower values. Classically, one would map each value to an excitation value which corresponds through a monotonously increasing function to a spiking latency or frequency, which can then be decoded by the corresponding inverse function. The dynamic propagation ordered by latency propagating "important" information first. By using Integrate-and-Fire neurons (Lapicque, 1907), we may associate a single neuron to every value from the vector we want to transmit. For the linear Leaky-IF, if we associate a driving current to each value $s_j$ (with $0 \leq j \leq N$), it will elicit spikes with latencies (Perrinet et al., 2004) $\lambda_j(s_j)$ where $\lambda_j$ is a monotonously decreasing function of $s_j$ corresponding to the transformation of the linear value into a latency. By this architecture, since the latency is monotonously decreasing, one implements a simple ArgMax operator in the form of the the ordered list of output spikes.
However a first problem arises when we consider the set of different excitation vectors globally. In fact, if the probability distribution function (pdf) of the input activation is not uniform, then the average spiking activity of the neurons will be *a priori* different. In the competitive network formed with the

neural cells, this is in disagreement with the general fact that spikes are similar and should therefore carry similar information to the different efferent neurons their axons are connected to. While the impact of each spike on a receiving neuron is variable (this being measured by the synaptic weight as the force of the post-synaptic current) spikes are binary all-or-none events. To maximize the representational information of possible spike patterns, it is necessary that neurons of the same class in one assembly should build up a distributed representation where activity is on average uniformly distributed. Equivalently, the distribution of coefficients represents exactly the *a priori* knowledge in Bayesian terminology, so that this equalization is justified by the hypothesis underlying Eq. 2. Another dual explanation is that spikes have similar metabolic costs and that the system should balance the use of the different neurons so as to minimize the average metabolic use by the system[3]. To optimize the efficiency of the ArgMax operator, one has therefore to ensure that one optimizes the entropy of the index of output spikes and therefore of the driving current. This may be ensured by modifying the different functions $\lambda_j$ so that for all $j$, the distributions of $\lambda_j(s_j)$ are similar and that the overall distribution corresponding to $\lambda$ has a shape adapted to the neural dynamics. The second point —finding an optimal latency envelop— will be out of scope of this paper, and we will without loss of generality only try to find functions $f_j$ (with $\lambda_j = \lambda \circ f_j$) such that the variables $z_j = f_j(s_j)$ are uniformly distributed between 0 and 1.

A standard method to achieve this homeostasis is to map the input vector $\{s_j\}$ through a point non-linearity[4] which equalizes the probability of the output (Atick, 1992). This method is similar to histogram equalization in image processing and provides an output with maximum entropy for a bounded output: it therefore optimizes the coding efficiency of the representation in terms of compression (van Hateren, 1993) or dually the minimization of intrinsic noise (Srinivasan et al., 1982). It may be easily derived from the probability $P$ of variable $s_j$ by choosing the non-linearity as the cumulative function

$$f_j(s_j^*) = P(s_j^*) \stackrel{\text{def}}{=} \int_{-\infty}^{s_j^*} dP(s_j) \tag{4}$$

where the symbol $dP(x) = p_X(x)dx$ will here denote in general the probability distribution function (pdf) for random variable $X$. The equalization process has been observed in a variety of species and is for instance perfectly illustrated in salamander's retina (Laughlin, 1981). It may evolve dynamically to slowly adapt to varying changes in luminance values, such as when the light diminishes at dawn but also to some more elaborated scheme within a map (Hosoya et al., 2005). As in theoretically "ideal democracy" all neurons are "equal" and when changing the neurons parameters such as with a learning, this process has to be dynamically updated over a time scale similar at least larger than the learning time scale so as to still achieve optimum balance. As a consequence, since

[3]However, this argument is a consequence through evolution from the first, since the goal of neural computations is primarily to be an efficient processor before being an economic one.

[4]That is to a set of scalar non-linearities applied independently to every single element of the vector.

for all $j$, the pdf of $z_j = f_j(s_j)$ is uniform and that sources are independent, the vector $\{z_j\}$ may be considered as a random vector drawn from an uniform distribution in $[0,1]$. Knowing the different spike generation mechanisms which are similar in that class of neurons, every vector $\{s_j\}$ will thus generate a list of spikes $\{j(1), j(2), \ldots\}$ (with corresponding latencies) where no information is carried *a priori* in the latency pattern but all is in the relative timing across neurons. Finally, the signal is transformed in this particular implementation of a L-NL architecture (see Fig. 1). The difference is first that the non-linearity is determined by the statistics of the input. Second, the neurons are not generated according to independent Poisson point processes but according to the deterministic (possibly noisy) equations of the generalized Integrate-and-Fire neurons, since there is *a priori* no reason to lose information.

We coded the signal in a spike volley, but how can this spike list be "decoded", especially if it is conducted over some distance and therefore with an additional latency? In the case of transient signals, since we coded the vector $\mathbf{s} = \{s_j\}$ using the homeostatic constraint from Eq. 4, we may retrieve the analog values from the order of firing neurons in the spike list. In fact, knowing the "address" of the fiber $j(1)$ corresponding to the first spike to arrive at the receiver end, we may infer that it has been produced by a value in the highest quantile of $P(s_{j(1)})$ on the emitting side. We may therefore decode the corresponding value with the best estimate $\hat{s}_{j(1)} = f^{-1}_{j(1)}(\frac{1}{N})$ where $N$ is the total number of neurons. This is also true for the following spikes and if we write as $r_{j(k)} = 1 - z_{j(k)} = \frac{k}{N}$ the relative rank of the spike (that is neuron $j(k)$ fired at rank $k$), we can reconstruct the corresponding value as

$$\hat{s}_{j(k)} = f^{-1}_{j(k)}(1 - \frac{k}{N}) \tag{5}$$

This corresponds to a generalized rank coding scheme (Perrinet, 1999; Perrinet et al., 2001). First, it loses the information on the absolute latency of the spike train which is giving the maximal value of the input vector. This has the particular advantage of making this code invariant to contrast (up to a fixed delay due to the precision loss induced by noise). Second, when normalized by the maximal value, it is a first order approximation of the vector which is especially relevant for over-complete representations where the information contained in the rank vector is greater than the information contained in the particular quantization of the image[5].

This code therefore focuses on the particular sequence of neurons that were chosen and loses the particular information that may be coded in the pattern of individual inter-spike intervals in the assembly. A model accounting for the exact spiking mechanism would correct this information loss, but this would be at the cost of introducing new parameters (hence new information), while it

[5] To give an order of the upper bound for the information in a ranked spike list is thanks to Stirling's approximation of order $\log_2(N!) = O(N.\log(N))$, that is more than 2700 bits for 324 neurons. However, we are generally unable to detect quantization errors on an image consisting of more 256 gray levels, that is for 8 bits per pixel.

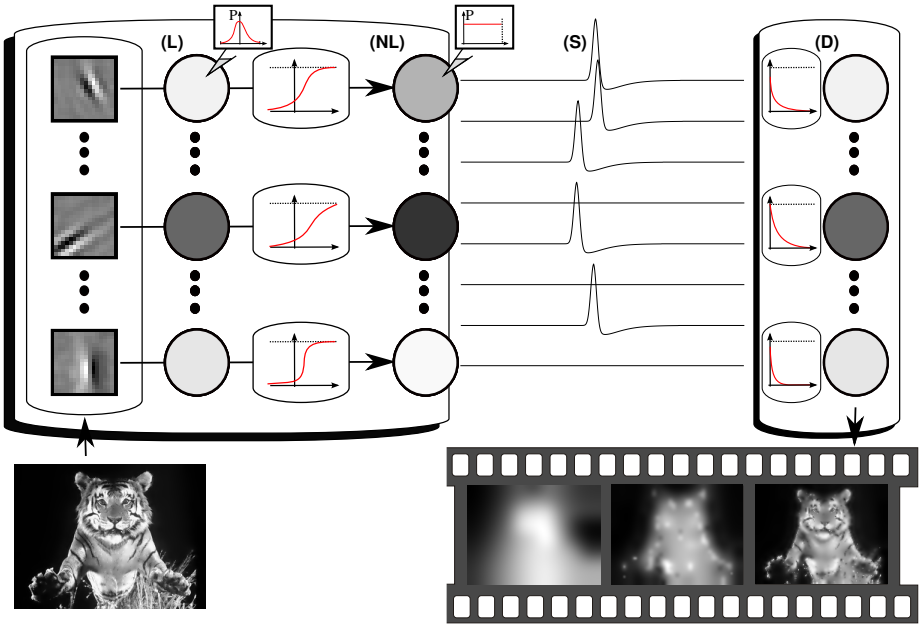


Figure 1: **Spike coding channel using homeostatic gain control.** We show here how a bundle of L-NL neurons (Carandini et al., 1997, 2005) tuned by a simple homeostatic mechanism allow to transfer a transient information, such as an image, using spikes. **(L)** The signal to be coded, for instance the match $s_j$ of an image patch (the tiger on the left bottom) with a set of filters (edge-like images), may be considered as a stochastic vector defined by the probability distribution function (pdf) of the values $s_j$ to be represented. **(NL)** By using the cumulative function as a point non-linearity $f_j$, one ensures that the probability of $z_j = f_j(s_j)$ is uniform, that is that the entropy is maximal. This non-linearity in the L-NL neuron implements a homeostasis that is controlled only by the time constant with which the cumulative probability function $f_j$ is computed (typically $10^4$ image patches in our case). **(S)** Any instance of the signal may then be coded by a volley of spikes: a higher value corresponds to a shorter latency and a higher frequency. **(D)** Inversely, for any spike events vector, one may estimate the value from the firing frequency, the latency. We may simply use the ordering of the spikes since the rank provides an estimate of the quantile in the probability distribution function thanks to the equalization. Using the inverse of $f_j$ one retrieves the value in feature space so that this volley of spikes is decoded (or directly transformed) thanks to the relative timing of the spikes using the modulation (see Eq. 5). This builds a robust information channel where information is solely carried by spikes as binary events. Given this model, the goal of this work is to find the most efficient architecture to code natural images and in particular to define a coding cost and to derive a learning algorithm. To draw a quantitative comparison with state-of-the-art algorithms Lewicki and Sejnowski (2000); Rehn and Sommer (2007); Smith and Lewicki (2006), we will use the framefork used in SparseNet (Olshausen and Field, 1998).

seems that this information would have a low impact relative to the total information (Panzeri et al., 1999). More generally, one could use different mappings for the transformation of the $z$ value into the spike volley which can be more adapted to continuous flows, but this scheme corresponds to an extreme case (a transient signal) which is useful to stress on the dynamical part of the coding by emphasizing solely on the first spike per neuron (van Rullen and Thorpe, 2001) and is mathematically more tractable. On a practical note, we similarly use the fact that the inverse of $f_j$ may be computed from the mean over trials of the function of the absolute functions as a function of the rank. In particular, one may show that the coding error is proportional to the variability of the sorted coefficients (Perrinet et al., 2004), the rest of the information being the information coded in the time intervals between two successive spikes. Thus, the efficiency of information transmission will directly depend on the validity of the hypothesis of independence of the choice of components and therefore on the statistical model build by the LGM.
It should be also noted that no explicit reconstruction is *necessary* (in the mathematical sense of the term) on the receiver side as we do here, since the goal of the receiver could only be to manipulate information on for instance some subset of the spike list (that is on some receptive field covering a subpart of the population). In particular one may imagine that we may add some arbitrary global point linearity to the $z$ values in order to threshold low values or to quantize values (for instance set all values to 1 only for the first 10% of the spikes). However, this full reconstruction scheme is a general framework for information transmission, and we may then imagine that if for instance we pool information over a limited receptive field, the information needed (the ranks in the sub-spikelist) will still be available to the receiver directly without having to compute the full set (in fact, since the pdf of $z$ is uniform, the pdf of a subset of components of $z$ is also uniform). Finally, we defined a simple spike coding algorithm to transmit information robustly with events. However, it is not yet clear how we may quantitatively estimate its efficiency.

## 2.3 Definition of the efficiency of Spike coding

Now that we defined the key concepts of the spike coding algorithm, we should be able to derive a generic cost function that will allow us to quantify the efficiency of different coding algorithms but also to derive a learning algorithm for the spike coding algorithm defined above. In fact, unless the dictionary $\mathbf{A}$ is orthogonal, when choosing one component over an other (for instance the one that maximizes Eq. 4), any choice may influence the choice of the other components. For instance, if we chose the successive neurons with maximum linear correlation values, the resulting representation will be proportionally more redundant when the dictionary gets more over-complete so that the linear correlation values in general do not correspond to the coefficients of the LGM. For every signal $\mathbf{x}$, one may state as in Occam's razor that given two solutions of similar quality, the best is the one with lowest representational complexity. Globally, we may introduce an "Occam factor" to compare the efficiency of dif-

ferent representations (MacKay, 2003, Ch. 28.3). This factor may be expressed as the Kolmogorov-Chaitin complexity and can be formalized in a probabilistic framework by using the bound given by Shannon's coding theorem as the average measure of the information to code the input given the solutions (the coding sequences) and the model's parameters (Lewicki and Sejnowski, 2000). A goal is therefore to maximize the evidence of the model that is to minimize this information or similarly minimize the description length (Rissanen, 1978). Furthermore, in the context of dynamical coding by spikes, this coding is progressive and there will be a dynamical compromise between the precision and the complexity of the representation.
Using the same notation as in Sec. 2.1 and Olshausen and Field (1998), the total representational cost is $\mathcal{C} \stackrel{\text{def}}{=} E(-\log P(\mathbf{s}|\mathbf{x},\mathbf{A}))$, where $E(.)$ denotes averaging over multiple trials (that is different image patches $\mathbf{x}$) and $\mathbf{s}$ is the coding sequence. For one coding sequence, this cost may thus be written as the sum of its likelihood probability knowing the set of sources added to the coding length of the set of sources:

$$\mathcal{C}(\mathbf{s}|\mathbf{x},\mathbf{A}) = \log Z + \frac{1}{2.\sigma_n^2}\|\mathbf{x} - \sum_j \hat{s}_j.\mathbf{A}_j\|^2 - \log P(\mathbf{s}|\mathbf{A}) \tag{6}$$

where $Z$ is the partition function (that we will omit in the sequel) and $\hat{\mathbf{s}}$ is the decoded sequence. The efficiency cost will be measured in bits if the logarithm is of base 2 (as will be assumed without loss of generality in the following). For any coding $\mathbf{s}$, the first term corresponds to the information from the image which was not retrieved by the whole coding/decoding algorithm (reconstruction cost) and that can be encoded at best using entropic coding pixel by pixel (it's the log-likelihood in Bayesian terminology). The second term is the representation cost: it quantifies the efficiency of the representation as the description length of the coefficients and is equal to the entropic coding of $\mathbf{s}$ knowing its probability distribution function (it is the log-prior).
We will assume independence of the coefficients and therefore $\log P(\mathbf{s}|\mathbf{A}) = \sum_j \log P(\mathbf{s}_j|\mathbf{A})$. Moreover, based on a parameterization of the coefficients' prior and on the assumption of a perfect reconstruction (that is $\hat{\mathbf{s}} = \mathbf{s}$), this yields the sparseness cost defined in Olshausen and Field (1998):

$$\mathcal{C}_1(\mathbf{s}|\mathbf{x},\mathbf{A}) = \frac{1}{2.\sigma_n^2}\|\mathbf{x} - \sum_j s_j.\mathbf{A}_j\|^2 + \beta\sum_j \log(1 - \frac{\mathbf{s}^2}{\sigma^2}) \tag{7}$$

where $\beta$ is the steepness of the prior and $\sigma$ is the prior scaling (see Figure 13.2 from (Olshausen, 2002)). This cost is related to the classical cost with the $L_1$ norm but represents a more kurtotic probability distribution function for the prior than the Laplacian prior corresponding to the $L_1$ norm. This cost therefore links the efficiency to the sparseness of the code by parametrizing *a priori* the sparseness of the coefficients, the "occam factor" measuring the global evidence of the model knowing this parametrized prior.
This liberty in the definition of the sparseness leads to a wide variety of proposed solutions to defining and optimizing this cost or *sparse coding* (for a

review, see (Pece, 2002)) such as numerical optimization (Lee et al., 2007; Olshausen and Field, 1998), non negative matrix factorization (Lee and Seung, 1999; Ranzato et al., 2007) or by using Matching Pursuit (Perrinet et al., 2003; Rehn and Sommer, 2007; Smith and Lewicki, 2006). Note that this cost will efficiently quantify the representation efficiency if the pdf of the coefficients is indeed well fitted to the parameterization. Moreover, this variety may exist in biological systems by varying the optimal sparseness levels via regulation mechanisms (Assisi et al., 2007). However, this parameterization is not known *a priori* and must be tuned accordingly to fit the model to the statistics of natural images and be further validated. This is the reason why we did build a non-parametric measure by taking advantage of the fact that thanks to the homeostasis, the probability of firing of every fiber is uniform across the population. In fact, spikes are *a priori* equally likely to be generated on any of the $N$ neurons (see Sec. 2.2), so that the probability of the origin of any new spike is simply $\frac{1}{N}$. Therefore, differently to the SparseNet algorithm, the model for the statistics of the LGM assumes that spikes are independent all-or-none events and carry a binary representation as was presented above for the Spike Coding algorithm (see Sec. 2.2). This explicitly defines the information content of a spike volley as an ordered list of spikes where the whole information is coded in the "addresses" of the different spikes in the list. Using a dictionary of $N$ neurons, the cost *per spike* may then be defined as $\log_2(N)$ bits per spike, so that we propose for the coding cost of a spike list to be simply:

$$\mathcal{C}_0(\mathbf{s}|\mathbf{x},\mathbf{A}) = \frac{1}{2\sigma_n^2}.\|\mathbf{x} - \sum_j \hat{s}_j.\mathbf{A}_j\|^2 + \log_2(N).\|\mathbf{s}\|_0 \tag{8}$$

where $\|\mathbf{s}\|_0$ is the length of the retrieved solution (or also the $L_0$ norm). Again, this cost is only valid if the probability of every spike is a priori uniform and therefore the cost used in (Rehn and Sommer, 2007) should include a correction term in the $L_0$ norm to account for this non-unifomity. It also explicitly rates the economy of consumed metabolic resources as is used in (Rehn and Sommer, 2007), but we may again consider this only as a consequence of the optimization. Note first that for any spike coding solution, this cost function is dynamic since the number of spikes increases in time. Note also that it links efficiency to sparseness for a known representation error solely on the basis that the spiking representation is binary and event-based. More generally, such a sparse representation is the best solution to allow a good discriminability between different patterns and is similar with information criterions such as the AIC (Akaike, 1974) or distortion rate (Mallat, 1998, p. 488). For instance, as a model of the input layer of the primary visual cortex, optimizing the coding according to Eq. 8 will provide the best representation to segregate different orientations for instance by representing the ridge of edges in images instead of representing the linear correlation as defined by Eq. 4. In a nutshell, sparser representation make "more peaked" cross-correlograms. This also means that the selectivity of neurons is therefore sharper than the response predicted by linear filters, such as is observed in orientation tuning (Troyer et al., 1998). Resolving the

coding problem with the $L_0$ norm is therefore a matter of getting the best $\mathbf{s}$ in the sense of Eq. 8 knowing $\mathbf{x}$, that is $\text{ArgMin}_s(\mathcal{C}_0(\mathbf{s}|\mathbf{x},\mathbf{A}))$. However, even when assuming perfect quantization, this is a non-convex optimization problem which is *NP-complete* with respect to the dimension $N$ of the dictionary (Mallat, 1998, p. 409). Instead of going back to an orthogonal basis to circumvent this difficulty, biological neural systems seem to use sparsity as a generic tool for efficient signal representation (Baudot et al., 2004; Deweese and Zador, 2003; Vinje and Gallant, 2000). We will present here a solution to this problem inspired by the architecture and dynamics of the primary visual cortex.

## 2.4 Sparse Spike Coding

We saw that overcoming the inefficiency of the L-NL architecture by optimizing the choice according to Eq. 8 leads then to a combinatorial explosion. To solve this NP-complete problem to model realistic representations such as when modeling the primary visual cortex, one may implement a solution designed after the richly laterally connected architecture of cortical layers. In fact, an important part of cortical areas consists of a lateral network propagating information in parallel between neurons. We will here propose that the NP-problem can be approximately solved by using a cross-correlation based inhibition between neurons. In fact, as was first proposed in *Sparse Spike Coding* (SSC) (Perrinet et al., 2002), one could use a greedy algorithm on the $L_0$ norm cost defined in Eq. 8 to recursively compute the sparse coefficients. The sparse coding part is basically equivalent to the Matching Pursuit (MP) algorithm (Mallat and Zhang, 1993) and the sparse coefficients will be transmitted, quantified and decoded in the same way as in the Spike Coding scheme for the linear correlation coefficients (see Sec. 2.2).

More specifically, let's first define Competition-Optimized Matching Pursuit (COMP) by introducing non-linearities in the choice step of MP corresponding to the optimized gain control functions defined in Eq. 4 (Perrinet, 2008). Like Matching Pursuit, it is based on an initialization and two repetitive steps. At initialization, given the signal $\mathbf{x}$, we set up the input vector as the absolute value of the linear correlation coefficient, that is for all $j$ to $s_j^{(0)} = |\rho_j|$ (and we will write $\mathbf{x}^{(0)} = \mathbf{x}$, $\rho_j^{(0)} = \rho_j$). These values $\mathbf{s}^{(0)}$ are proportional to the *a posteriori* probability knowing $\mathbf{x}$ (see Eq. 4). Then we repeat the two following steps that we index by their rank $k$ (initialized at 0). First, we are searching for the best*single* source corresponding to the maximum *a posteriori* (MAP) knowing $\mathbf{x}^{(k)}$ which is proportional to $s_j^{(k)} = |\rho_j^{(k)}|$ but modulate this choice by the point non-linearity $f_j$. This Matching step is thus defined by:

$$j^{(k)} = \text{ArgMax}_j[f_j(s_j^{(k)})] \tag{9}$$

In a second Pursuit step, the information is fed-back to correlated sources through:

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} - \rho^{(k)}.\mathbf{A}_{j^{(k)}} \tag{10}$$

where we note $\rho^{(k)} = \rho^{(k)}_{j^{(k)}}$ the maximal scalar projection $< \mathbf{x}^{(k)}, \mathbf{A}_{j^{(k)}} >$ (see Eq. 2). Equivalently, from the linearity of the scalar product (see Eq. 4), we may instead propagate information laterally:

$$\rho^{(k+1)}_j = \rho^{(k)}_j - \rho^{(k)} < \mathbf{A}_{(k)}, \mathbf{A}_j > \tag{11}$$

As described in (Perrinet, 2004b), while the Matching step is efficiently performed by the LIF neurons driven by the (NL) layer (see Fig. 1), the pursuit step could be implemented in a cortical area by a correlation-based inhibition. In Fig. 1, it will correspond to a lateral interaction within the linear (L) neuronal population. This type of inhibition is typical of fast-spiking interneurons though there is no direct evidence of an activity-based synaptic topology. COMP shares many properties with MP, such as the monotonous decrease of the error or the exponential convergence of the coding. The algorithm is then iterated with Eq. 9 until some stopping criteria is reached.
Sparse Spike Coding (SSC) is then defined as the spike coding algorithm that transforms an image $\mathbf{x}$ into a list of spikes $\{j^{(k)}\}$ defined as the ordered list of the addresses of neurons that fired in the Matching step (and to the corresponding values $\{\rho^{(k)}\}$) as was described in the Spike Coding scheme (see Sec. 2.2). This scheme thus extends the algorithm based on the Matching Pursuit (MP) algorithm proposed in (Perrinet et al., 2002) by linking it to a statistical model which tunes optimally the matching step. In fact, all choices are statistically equally probable thanks to the adaptive point linearity so that one maximizes the entropy of every match and therefore the computational power of the ArgMax operator. Think *a contrario* to a totally unbalanced network where the match will be always a given neuron: the spikes are totally predictable and the information carried by the spike list then drops to zero. In practice, the $f_j$ functions are initialized for all neurons to the identity function (COMP is then a simple MP algorithm) and then Eq. 4 is evaluated after the end of every coding sweep using an online stochastic algorithm with a "learning" parameter corresponding to a smooth average which effect was controlled (see Fig. 4 and Annex. 5.5). As a matter of fact, this algorithm is circular since the choice of $\mathbf{s}$ is non-linear and depends on the choice of $f_j$. However, thanks to the exponential convergence of MP, for any set of components, the $f_j$ will converge to the correct non-linear functions as defined by Eq. 4.

## 2.5 Introducing Hebbian Learning in SSC

We may now finally derive the unsupervised learning model by optimizing the efficiency of SSC with respect to the efficiency cost defined in Eq. 8. In fact, on a longer time scale, the efficiency of the system may be optimized by slowly adapting the dictionary $\mathbf{A}$ as in SparseNet thanks to the sparse solution given by the coding algorithm. More generally, knowing the coding algorithm, an unsupervised learning solving the efficiency cost is related to the goal of Independent Component Analysis (ICA), since knowing that the signal is created by a mixture of components as in Eq. 1, it tries to find the best set of components

so that the representation is well described by the fewer number of components. As in SparseNet, this can be seen as a joint optimization of the reconstruction and the sparseness. We may implement this for every image at any coding step $k$ after Eq. 9 (that is after a spike of address $j^{(k)}$ is emitted) since we have an evaluation of the log-likelihood by the distance of the residual image to the selected filter, that is to $\|\mathbf{x}^{(k)} - \rho^{(k)}.\mathbf{A}_{j^{(k)}}\|^2$, the rest of the signal being regarded as a perturbation which will cancel out on average. Using the gradient descent approach used in (Olshausen and Field, 1998), we similarly infer that we may slowly modify the winning weight vector corresponding to the winning filter $\rho^{(k)}.\mathbf{A}_{j^{(k)}}$ by taking it closer to $\mathbf{x}^{(k)}$,

$$\begin{aligned}\frac{\partial \mathcal{C}(s_{j^{(k)}}|\mathbf{x}^{(k)}, \mathbf{A})}{\partial \mathbf{A}_{j^{(k)}}} &= \frac{1}{2\sigma_n^2}.\frac{\partial \|\mathbf{x}^{(k)} - \rho^{(k)}.\mathbf{A}_{j^{(k)}}\|^2}{\partial \mathbf{A}_{j^{(k)}}} \\ &= \frac{1}{2\sigma_n^2}.\rho^{(k)}(\mathbf{x}^{(k)} - \rho^{(k)}.\mathbf{A}_{j^{(k)}})\end{aligned}$$

that is:

$$\mathbf{A}_{j^{(k)}} = \mathbf{A}_{j^{(k)}} + \eta\rho^{(k)}(\mathbf{x}^{(k)} - \rho^{(k)}.\mathbf{A}_{j^{(k)}}) \tag{12}$$

where $\eta$ is the learning rate, which is inversely proportional to the time scale of the features being learned.A more rigorous mathematical approach for the learning is to consider a rotation of $\mathbf{A}_{j^{(k)}}$ toward $\mathbf{x}^{(k)}$ using a Jacobi Matrix rotation so that all component vectors stay on the unit sphere. In practice, Eq. 12 for small learning rates $\eta$ followed by a normalization is a good approximation of this high-dimensional (linear) transform. Basically, it will decrease the angle between the filter and the feature and therefore increase the correlation: since the energy decreases at every step in COMP by the energy of the selected coefficient (see (Perrinet, 2004b, Sec. 2.1.3)), it will ensure that the decrease is quicker and thus that the representation gets sparser. Similarly to Eq. 17 in (Olshausen and Field, 1998) or to Eq. 2 in (Smith and Lewicki, 2006), the relation is linear: it is an "Hebbian" rule (Hebb, 1949) in the classical sense since it will enhance the weight of neurons of correlated pre- and post-synaptic neurons. However, the novelty of this formulation is to apply this formulation to the sparse representation at every single spike. In fact, note that if the coding step is not the initialization, $\mathbf{x}^{(k)}$ is the residual image (thus a non-linear transform of the initial $\mathbf{x}$ computed using Eq. 10) and the $\rho_j^{(k)}$ are the linear correlations with $\mathbf{x}^{(k)}$. The biological correlate of this adaptation could be a "slow" anterograde signal going backwards this feed-forward pass (Harris, 2008). An improvement is to "back-propagate" in early spikes of the following choices to correct the gradient descent instead of regarding the residuals as a perturbation. This corresponds to Spike Coding with Orthogonal Matching Pursuit as was implemented (without quantization) in the learning algorithm used in (Rehn and Sommer, 2007). However, this would imply a more complex neural architecture and it did not drastically change the efficiency of the system. Without homeostasis, this algorithm (as well as SparseNet) is unstable. In fact, since we start with random filters, it is is more likely that any salient feature would be selected at first and will modify the first winning filter. There

is therefore a higher probability that the same neuron will be selected with a higher probability in subsequent learning steps, causing a diverging non uniformity in the balance of the learning across neurons. Whereas SparseNet uses the norm of the filters to control the variance of the coefficients across neurons, the SSC matching criteria (see Eq. 9) is intrinsically independent to the norm of the filters. While we implemented in AMP the homeostasis used by (Olshausen and Field, 1998), if we use the homeostatic regulation (which has a similar time-scale than the learning), the probability of choosing any neuron in COMP remains uniform and ensures the convergence of the learning algorithm (see Annex. 5.4). The homeostasis will therefore optimize the balance between the neurons, assuring that the internal representation driving the spiking neurons may always be considered as a uniformly distributed random vector. Note then that on a long time scale, if two filters at some point during the learning correspond to filters with different selectivity (thus, they have different pdf, the more selective being more kurtotic), they are still selected with the same probability thanks to the non-linearity. However, since on a batch of natural scenes, the filters corresponding to the less selective neurons are less likely to be present and from the relative "boost" induced by the non-linearity and which acts as a cortical gain control, these filters are more likely to change. It is easier to imagine this property by drawing the Voronoi diagram on the unit $M$-dimensional sphere corresponding to the possible positions of the filters, the dictionary being represented by $N$ points on this sphere. For the diagram corresponding to the dictionary each centroid will be attracted toward the mean vector of inputs corresponding to its Voronoi cell, as in the K-means algorithm. Since we force the probability of selecting centroids to be uniform, the equilibrium of the learning is when the population of filters, that is the dictionary, forms an uniform tiling of filter space. As a consequence, the $f_j$ functions become equal at convergence and the matching step becomes similar to the one in MP and in particular if one perturbs a filter (by setting it to a random vector for instance, that is a point on the unit $M$-dimensional sphere) the algorithm will change the whole dictionary so that it settles to a new stable point (see Annex. 5.4). Finally, we find the counter-intuitive result that in aSSC, the homeostasis is more important during the learning period and may be ignored when synapses don't evolve anymore.

## 2.6 Adaptive Sparse Spike Coding (aSSC)

In summary, the whole learning algorithm is given by the following nested loops:

1. Initialize the components $\mathbf{A}$ to random values on the unit $N$-dimensional sphere and set the point non-linear gain function to unity ($f_j(s) = s$ for all $j$),
2. repeat:
   (a) draw a signal $\mathbf{x}$ from the database,
   (b) compute $\rho_j$ for all $j$ using Eq. 4, $k = 0$

(c) while $\|\mathbf{x}^{(k)}\|^2$ is above a certain precision threshold do sparse spike coding (SSC):

i. emit a spike by selecting the best match $j^{(k)}$ with Eq. 9,

ii. modify information correlated to $j^{(k)}$ by computing $\rho_j^{(k+1)}$ for all $j$ using Eq. 11,

iii. slowly modify $\mathbf{A}_{j^{(k)}}$ using Eq. 12,

iv. slowly update the $f_j$ for all $j$ using Eq. 4

v. increment the rank $k$

When convergence is achieved, one could simply make a coding by using step (b) to initialize and then (c) and optionally for the pure spike coding evaluate the coefficient using Eq. 5 in step 2-(c)-ii. One advantage is that the greedy algorithm may adapt to quantization errors (Perrinet et al., 2004, Fig.10). The decoding of the spike list $\{j^{(1)}, j^{(2)} \dots j^{(k)} \dots\}$ is then simply:

1. Initialize $\hat{\mathbf{x}}$ to a zero image and the rank $k$ to zero,

2. while we have spikes do :

   (a) retrieve the address $j^{(k)}$ of the spike and the corresponding value $\hat{s}^{(k)}$ of the coefficient using Eq. 5,

   (b) add $\hat{s}^{(k)}.\mathbf{A}_{j^{(k)}}$ to $\hat{\mathbf{x}}$,

   (c) increment the rank $k$,

Note that this pseudo-code is given in a traditional sequential way where all steps are given the one after the other. This corresponds to our actual simulations (see Annex. 5.1) but only corresponds to an event based description of the dynamical processes occurring in the system. In fact, in a dynamical implementation, all processes are done in parallel and events just correspond to the times where the integration reached a threshold (Perrinet, 2004b).

# 3 Results on natural images

## 3.1 Receptive field formation: comparison with SparseNet

We first compared this novel aSSC algorithm with the SparseNet algorithm. In fact, this algorithm as other similar schemes only differs by the coding method used to obtain the sparse representation and by the homeostasis scheme. In particular, we focused herein in the validation and quantitative comparison of both algorithms in terms of efficiency on the task at hand that we defined[6]. We used a similar context and architecture as the experiments described in (Olshausen and Field, 1998) and used in particular the same database of inputs as the SparseNet algorithm and restrict ourselves to study the selection

[6]See Annex. 5.1 for the table of parameters, details of the experimental setup and to a link to reproduce the results.

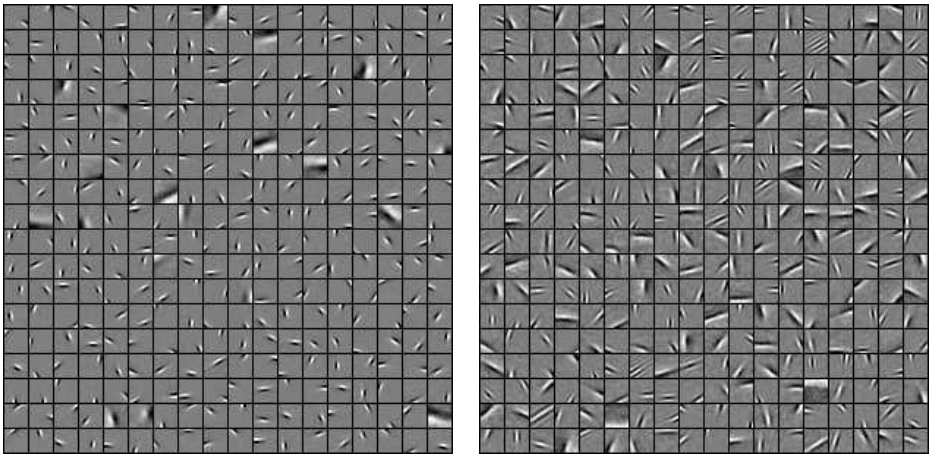

Figure 2: **Results of the proposed aSSC scheme compared to** SparseNet**.** Starting with random filters, we compare here the results of the learning scheme with 324 filters at convergence (20000 steps) using **(Left)** the classical conjugate gradient function method as is used in (Olshausen and Field, 1998) with **(Right)** the Sparse Spike Coding method. Filters of the same size as the imagelets ($16 \times 16$) are presented in a matrix (separated with a black border). Note that their position in the matrix is as in ICA arbitrary (invariant up to any permutation). Results replicate the original results of (Olshausen and Field, 1998) and are similar for both methods: both dictionary consist of gabor-like filters which are similar to the receptive fields of simple cells in the primary visual cortex. Edges appear in these conditions to be the independent components of natural images. However, the distribution of the quality of the edges (in particular their mean frequency, length, width) appears to be different and the question remains as how we may compare the efficiency of the different architectures quantitatively.

of optimal filters on imagelets (that is small patches from natural images)[7]. In particular, these images are static, grayscale and filtered according to the same parameters to allow a one-to-one comparison of the different algorithms.

Here, we show the results for $16 \times 16$ patches (so that $M = 256$) from the whitened images and we chose to learn $N = 324$ filters which are replicated as ON and OFF filters (but see Annex. 5.2). Results show the emergence of edge-like filters (see Fig. 2) for a wide range of parameters (see Annex. 5.5 for an analysis of the robustness of the methods to variations of the parameters). Studying the evolution of one single filter during the learning shows that it firsts

[7] These results should be taken as a lower bound for the efficiency of adaptive Sparse Spike Coding, since the sparseness in imagelets is only local, but sparseness is also spatial in natural images (Perrinet et al., 2004). For instance, it is highly probable in natural images that large parts of the space —such as the sky— are flat. See (Perrinet, 2007, Sec. 3.3.4) for an extension to whole images.

represent any salient feature (such as a sharp corner or edge) and that if it contains multiple edges only the most salient edge remains later in the learning. This is due to the competition between filters, the algorithm ensuring that independent features should not be mixed since this will result in a larger $L_0$-norm. When looking at very long learning times, the solution is not fixed (for both algorithms) and edges may smoothly drift from one orientation to another while the cost still remains stable. This is due to the fact that there are many solutions to the problem and that there is no constraint such as topological links between filters to decrease the degree of liberty of solutions. Thus, as in ICA results are to be understood as a whole, and if for instance two filters are swapped in the dictionary, the efficiency stays the same.
However, it is not clear by the sole shape of the filters alone which solution is most efficient. In fact, rather than the shape of the components individually, it is the distribution of the assembly of components in the image space that will yield different efficiencies. Such an analysis was performed with a qualitative analysis of the filters' shape, by fitting them with Gabor filters (Lewicki and Sejnowski, 2000). A recent study compares the distribution of the parameters of the Gabor filters with neurophysiological experiments (Rehn and Sommer, 2007). They did indeed show that their learning scheme, which is also based on a Matching pursuit algorithm, did better match than SparseNet the distribution of some parameters of Gabor filters over the set of filter observed in the macaque's primary visual cortex. However, if this similarity is certainly necessary, it is not sufficient to understand the effect of each parameter and more generally to the emergence of edge-like receptive fields. We will rather try to evaluate *quantitatively* the relative efficiency of the different learning schemes to extract what aspect is the most relevant.

### 3.2 Quantifying efficiency compared to SparseNet

To address this question, we compared the quality of both methods by computing the mean efficiency of the coding as the learning converged. Using $5.10^4$ imagelets drawn from the natural image database, we performed the progressive coding of the images using both methods first for random vectors and then for the filters learned by each method. First, we quantified at the end of the coding the distribution of coefficients for the different cases. To allow a comparison of the coefficients, we normalized the coefficients by the energy of the imagelets (since thanks to Eq. 4 and Eq. 2, we have $\rho_j = s_j.\frac{\|\mathbf{A}_j\|}{\|\mathbf{x}\|}$) and by the norm of the filters to retrieve coefficients such that

$$\frac{\mathbf{x}}{\|\mathbf{x}\|} = \sum\nolimits_{1\leq j\leq N} \rho_j.\frac{\mathbf{A}_j}{\|\mathbf{A}_j\|} \tag{13}$$

these coefficients are similar to the measure of the linear correlation coefficient (see Eq. 4). These non-linear correlation coefficients do match the linear correlation coefficients when the dictionary which was selected at each coding is quasi-incoherent, that is that every selected filter is perpendicular to the residual of the coding (Gribonval and Vandergheynst, 2006). This is not true in general

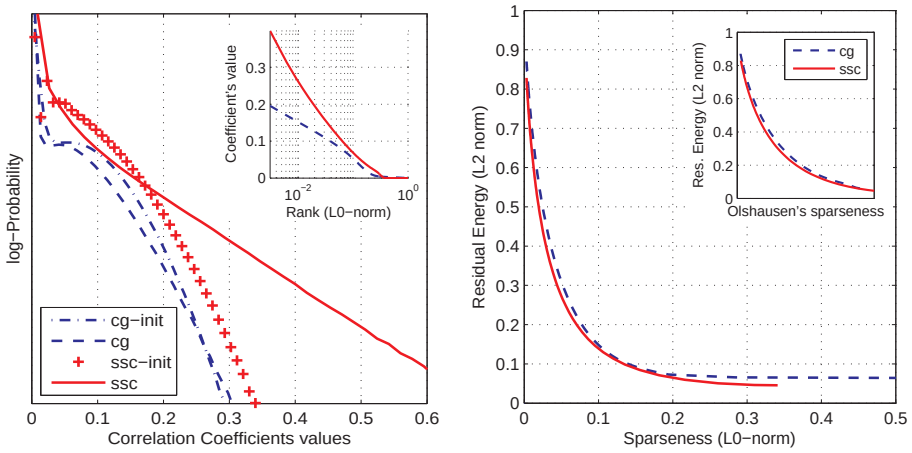


Figure 3: **Coding efficiency of adaptive Sparse Spike Coding.** We evaluated the quality of the algorithm with two different coding strategies by comparing the coding efficiency of the sparse spike coding ('ssc') method with the classical conjugate gradient function ('cgf') method as is used in (Olshausen and Field, 1998). For the coding of a set of $5.10^4$ image patches drawn from a database of natural images, we plot *(Left)* the distribution of both methods before and after the convergence of the learning phase. At initialization, the distributions are more gaussian (curves 'cg-init' and 'ssc-init') while they get more kurtotic: both algorithms yield at convergence sparse distributions of the coefficients. We also plot *(Right)* the mean final residual error ($L_2$ norm) as a function of the relative number of active (or non-zero) coefficients (that is the normalized $L_0$ norm and the coding step for SSC) which provides an estimate of the mean coding efficiency for the image patches. Best results are those providing a lower error for a given sparsity or a lower sparseness (better compression) for the same error. In fact, the efficiency cost measures Occam's razor applied to coding: it states that for a given $L_2$ norm, a lower complexity (that is a lower $L_0$ norm) is more efficient (graphically, an horizontal line would cross from left the best solution first). It should be noted that it is also similar for the cost based on the $L_1$ norm, a result which may reflect that the $L_0$ norm defines a stronger sparseness constraint. Moreover, one should take into account the fact that in the proposed algorithm the coding is constrained to be binary while it is analogous in SparseNet or in Lewicki and Sejnowski (2000); Rehn and Sommer (2007); Smith and Lewicki (2006) and there is no explicitly defined quantization. Also, while in these schemes this aspect of the coding is not studied or is expected to be the fixed point of a recurrent system, we take here advantage of the spiking representation to optimize the representation and the speed of information transmission by a single spike volley.

and the $\rho_j$ correspond here rather to a measure of the quality (measured as a log-probability) of the match with the signal of each from the sparse set of selected sources. When plotting the histogram of the non-linear correlation coefficients, one sees that distributions are approximately gaussians with the initial random filters —a result that one could have predicted from the central limit theorem— but that these become very kurtotic after the convergence of the learning (see Fig. 3-Left). The measure of the kurtosis of the resulting code words proved to be very sensitive and a poor indicator of the global efficiency, in particular for code words at the beginning of the coding, when many coefficients are still strictly zero. In particular, it seemed inaccurate to compare the kurtosis for systems with different over-completeness factors as in (Rehn and Sommer, 2007). Both final distributions seemed to fit well the bivariate model introduced in (Olshausen and Millman, 2000) where coefficients are $L_0$ sparse and the non-zero coefficients follow a laplacian pdf. However, the SSC algorithm provided the most kurtotic distribution of the coefficients (with values around 60 versus 20 for SparseNet). Dually, plotting the decrease of the sorted coefficients as a function of their rank again showed that coefficients for SSC were first higher and then decreased quicker (see Fig. 3-Left Inset), following the link between both curves in Fig. 3-Left from Eq. 5. In a second analysis, we compared the efficiency of both methods while varying the number of active coefficients (the $L_0$ norm), that is the number of spikes during the progressive coding for SSC (Eq. 8). To compare this method with the conjugate gradient, a first pass of the latter method was assigning for a fixed number of active coefficients the best neurons while a second pass optimized the coefficients for this set of "active" vectors (see Fig. 3, Right). This method was also used in (Rehn and Sommer, 2007) and proved to be a fair method to compare both methods. At the same time, one could yield different mean residual error with different mean sparseness of the coefficients, as defined in Eq. 7 (see Fig. 3, Right Inset).
Controlling with a wide range of parameters and a variety of methods yielded similar qualitative results (such as changing the learning rate or the parameters of the conjugate gradient, see Annex. 5.5) proving that the hebbian learning converged robustly as long as the coding algorithm provided a good sparse representation of the input. As a result, it appeared in a robust manner that the greedy solution to the hard problem (that is SSC) is as efficient for the optimized cost but also to the cost defined in the relaxed problem (see Fig. 3). Moreover, it should be noted that this non-parametric method is controlled by less parameters (which were here optimized to give best operating point, see Annex. 5.1) and we should stress again that the SSC method simply uses a feed-forward pass with lateral interactions, while the Conjugate Gradient could only be implemented as the fixed point of a recurrent network. Therefore, applying a "higher-level" Occam razor confirms that for a similar overall coding efficiency, aSSC is better since it is of lower *structural* complexity[8]. Since the coding

[8] Accounting for a quantitative measure of the structural complexity of the different methods is for instance measured by the minimal length of a code that would implement them (for instance the number of characters of the program in memory). It would therefore depend on the machine on which it is implemented, and one would of course see a clear advantage of

used in aSSC is rather sub-optimal (MP) compared to other methods such as (Rehn and Sommer, 2007), we conclude that this improvement is mainly due to how we tuned the algorithm according to the efficiency cost defined in Eq. 8 and in particular to the homeostasis mechanism ensuring that all neurons fire equally.

### 3.3 Efficiency compared to Adaptive Matching Pursuit

The choice of the homeostatic regulation was based on the cost function and the hypothesis that led to it. In fact, by forcing that all neurons should be chosen with equal probability, we impose a strong constraint for the neural assembly (all neurons should be "equal") and this may hinder the global efficiency of the system. On the other hand, when choosing a more relaxed system (such as normalizing the filters or using the homeostatic rule defined in SparseNet) we obtain qualitatively different filters whose efficiency would depend on a different cost function. To resolve this ambiguity, we therefore compared the efficiency for the aSSC scheme that we presented above (see Fig. 2, Right) with a system where we just imposed the components to stay on the unit sphere, that is setting the homeostatic learning time to infinity. This last algorithm is exactly the Adaptive Matching Pursuit (AMP) algorithm that was studied previously (Perrinet et al., 2003) and which is similar to other strategies such as (Rehn and Sommer, 2007; Smith and Lewicki, 2006).
In fact, the homeostasis constraint in the AMP algorithm is relaxed and the filters will correspond to features of more various saliencies. In particular, we observe the emergence of both broader Gabor filters which better match textures and of checkerboard-like patterns (see the result after convergence at Fig. 4, Left). Because of their lower generality, these 'textural' filters will be more likely to be selected with lower correlation coefficients. They correspond more to the Fourier filters that one may obtain by PCA (see for instance (Fyfe and Baddeley, 1995)) or the simple Hebbian rule on linear coefficients and that are still optimal to code arbitrary imagelets such as noise (Li, 2006). The aSSC algorithm (see Sec. 2.6) ensures with the homeostasis constraint that all filters will be selected equally by the definition of the homeostasis in Eq. 4. In particular, the point non-linearity from Eq. 9 plays the role of a gain control. Compared to AMP, textured elements will be relatively "boosted" during the learning compared to the correlation coefficient computed on a more generic "edge" component. This explains that they would end up being less probable and why at the convergence of the learning there is no textured filters in Fig. 2, Right. As a conclusion, as was stated formally in Sec. 2.5, the homeostasis efficiently constrains the dictionary to better match the *a priori* pdf of natural scenes in the $M$-dimensional image space.
We may then compare quantitatively the efficiency of these two approaches. When not using the quantization step using the inverse $f_j$ function (see Eq. 5), the AMP yields a better final result since it represents more efficiently the noisy

aSSC on parallel machines.

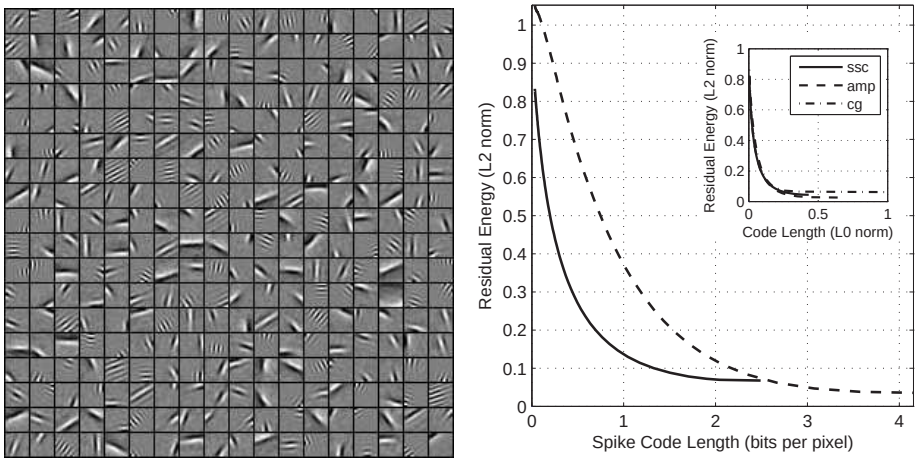


Figure 4: **Homeostasis implements efficient spike quantization during learning.** *(Left)* When relaxing the homeostatic constraint in the aSSC (see Sec. 2.6) larning algorithm to the one implemented in Adaptive Matching Pursuit (AMP), the algorithm converges to a set of filters which contains some less localized filters and to some high-frequency Gabors which correspond to more "textural" features as in (Perrinet et al., 2003). One may wonder if these filters are inefficient and capturing noise or if they rather correspond to inherent features of natural images in this LGM model (see Fig. 3). *(Right)* In fact, the AMP solution gives a slightly better result than aSSC in terms of residual energy as a function of pure $L_0$ sparseness (see inset) and is for that purpose of similar efficiency than conjugate gradient. However, when defining the efficiency in terms of the residual energy as a function of the description length of the spiking code word, then the proposed model is more efficient than AMP because of the quantization errors inherent to the higher variability of coded coefficients. Thus, including homeostasis improved the efficiency of adaptive Sparse Spike Coding both for the coding and the learning. It should be noted that homeostasis is important in aSSC during learning but that from the inherent equalization it is not useful in coding (see Sec. 2.5).

aspects of the signal. On average, this strategy provides a slightly better initial decrease of the residual energy: the components of the signal are better represented for a similar number of neurons. However, it is weaker when the $L_0$-sparseness is greater than $\sim 10\%$ of the dimension $M$, at which point noise dominates the signal (see Fig. 4, Right, inset). On the other hand, when using the quantization and therefore when rating the efficiency of the full spike coding / decoding system, the AMP approach will display a greater variability and there will be a greater quantization error[9]. Results show that in average, the loss in information transmission makes the AMP solution obviously less efficient than the aSSC approach (see Fig. 4, Right). This is due to the higher variability of coding coefficients in AMP and therefore to the greater quantization error induced from the reconstruction using Eq. 5. As a conclusion, both solutions have advantages, the efficiency depending on a definition of the utility function and how we assigned the distribution of resources to achieve this goal. In a nutshell, for the rapid spike coding of a transient signal an homeostatic approach as implemented in aSSC is significantly more adequate while on a longer term for a spike frequency representation, the more relaxed system in AMP may be sufficient.

# 4 Discussion

Using the tools of statistical inference and information theory, we derived quantitative costs for the efficiency of different representation models for low-level sensory areas. We then designed aSSC as a coding and learning solution which heavily relied on basic aspects of the neural architecture, namely the parallel event-based nature of the code. Applied on patches from natural scenes, we proved here that aSSC is superior to the SparseNet architecture in terms of the global efficiency of information transmission. This reflects the fact that in this local receptive fields the LGM describes well the images since they are for instance no occlusions. Similar approaches have been taken that could be grouped under the name of Sparse-Hebbian Learning (SHL) (Fyfe and Baddeley, 1995; Lewicki and Sejnowski, 2000; Olshausen and Field, 1996; Perrinet, 2004a; Rehn and Sommer, 2007; Smith and Lewicki, 2006; Zibulevsky and Pearlmutter, 2001). A common advantage of these strategies is that Hebbian learning may account for the formation of receptive fields if applied on a sparse representation and that the coding algorithm used to obtain this sparseness was of secondary importance. These algorithms may be variants of conjugate gradient, of Matching Pursuit or more generally based on correlation-based inhibition (for a review, see (Pece, 2002)). A more radical solution based on neurophysiological evidence and not based on a generative model was proposed by (Hamker and Wiltschut, 2007), but was in the end also interpretable as an optimization scheme and therefore to the definition of a cost through a generative model of the signal to code. Thus, these SHL schemes are all similar optimization algorithms, gradually improving

[9] This result did not change qualitatively when using an entropic cost in bits per spike, the AMP requiring necessarily less bits per pixel since in aSSC the coefficients' pdf is uniform.

the efficiency using a stochastic algorithm on the database of signals. With a correct tuning of parameters, all of these unsupervised learning algorithms will show the emergence of edge-like filters thanks to the correlation-based inhibition such as may be observed to be necessary for the formation of elongated receptive fields (Bolz and Gilbert, 1989). However, a major advantage of our formulation is the fact that it tightly couples the efficiency measure to the coding representation. In fact, we explicitly link here the sparseness constraint with the efficiency of inverting the generative problem. Also, we can predict that applying the homeostasis constraint on the other SHL schemes will result in schemes with more efficient spike coding capacity. In particular, the efficiency is based on the spiking nature of neural information while other algorithms relied on a firing-frequency representations. In these schemes based on an analog representation, the problem of coding and decoding of the values was not explicitly addressed and in most of the cases the decoding solution was achieved as the fixed point solution of a recurrent network. This solution therefore requires at each coding step to settle to a fixed point and is therefore incompatible with the rapidity of cortical processing (Keysers et al., 2000). Moreover, a crucial feature of our solution is that the output of the coding algorithm gives non-linear results. For instance, for a mixture of images, the output to the sum of two images is not necessarily the sum of both individual output. Note that as in (Troyer et al., 1998), the response selectivity to rotated oriented lines will be sharper than the linear response but also that it will be relatively contrast-invariant (Perrinet, 2005). This provides an alternative to the debate between forward and recurrent models for the origin of selectivity by offering a functional reasoning behind the emergence of orientation selectivity. In particular, we predict that it will exhibit a similar non-linearity in the spiking response without the need of explicitly adding after the first stage of matching a parametrized non-linear gain control that matches physiological recordings (Carandini et al., 1997, 2005). As a consequence, by taking advantage of the parallel architecture of the cortex, the aSSC architecture provides a new and simple interpretation for the receptive fields of neurons which in this view self organize optimally with neighboring neurons and can therefore only be understood as a whole in an assembly.

The work presented here is part of a larger program aiming at assessing qualitatively the functional efficiency of different modeling solutions to computational neuroscience problems and at providing computational applications. Using constraints from neuroscience, we have built a solution to the LGM inverse problem which we proved to be more efficient than the Matching Pursuit algorithm by using these quantitative tools. We proved that this aSSC algorithm was an efficient unsupervised algorithm that competes with standard methods for searching independent components in signals and that the coding was an efficient coder for binary messages with over-complete dictionaries. In fact, by including an adaptive homeostasis mechanism, we optimized the efficiency of the representation (the COMP algorithm) and proved that image patches could be efficiently coded by the binary event-based representation. We proved also that this homeostasis played a significant role in these results but also that counter-intuitively that textured filters could also be good candidates for optimal coding

in V1 if the goal was set by a different coding cost. Its efficiency makes it a good candidate for future technologies of information processing. In particular, it compares favorably with compression methods such as JPEG (Fischer et al., 2007). Moreover, a major advantage is that it provides a progressive dynamical result while the conjugate gradient method had to be recomputed for any different number of coefficients. In fact, the most relevant information is propagated first and the reconstruction may be interrupted at any time. All these models were implemented with the intention of providing reproducible research and are freely available and we encourage to modify them (see Annex. 5.1). The main advantage of this type of formulation is that it uses a simple set of operations: computing the correlation, applying the point non-linearity from a Look-Up Table, choosing the ArgMax, doing a subtraction, retrieving a value from a Look-Up-Table (see Sec. 2.6). In particular in this parallel algorithm, the complexity of the ArgMax operator does not depend on the dimension of the vector as in classical solutions and the transfer of this technology to parallel architectures will provide a supra-linear gain of performance.
To conclude, we proved that using a spiking event-based representation could be an advantage for the modeling of efficient neural networks and not only a constraint for biological plausibility. Thus, this may explain on a functional level why spikes have been selected during evolution as an efficient signal quanta for long range, rapid communication. However, a major limitation of this algorithm is the use of transient signals and of relatively abstract neurons. This choice was made on purpose to stress the importance of the transient network's dynamics versus traditional strategies using spike frequency representations. It shows that solutions using spike coding/decoding may be built and that they prove to be of better efficiency than traditional solutions. A solution of SSC for continuous flows was proposed under the term Causal Sparse Spike Coding in (Perrinet, 2007, Sec. 3.4), but some new problems arise (for instance the dynamical compromise between speed and precision) that were beyond the scope of this paper. Another extension of the algorithm is to not use the implicit ON-OFF symmetry of filters which introduces the constraint that if a filter exists, then the symmetric filter exists, that is that we rate the efficiency of a match by the absolute value of the correlation coefficient. The relaxed condition proved to be more efficient, suggesting that the symmetry that is observed is more a general effect and that since neurons are not linear only integrators with a rectifier, more efficient solutions may exist (see Annex. 5.2). This simple architecture provided also a rich range of other novel experiments, such as introducing topological relations between filters or by using a representation with some build-in invariances, such as translation and scaling in a gaussian pyramid such as in (Bednar et al., 2004; Hyvärinen et al., 2001). This last example provided a multi-scale analysis algorithm where the set of filters that were learned were a dictionary of mother wavelets of the multi-scale analysis, hence the name of SparseLet Analysis (Perrinet, 2007, Sec. 3.3.4). Another interesting perspective is to study the evolution of the efficiency of the algorithm with the complexity of the representation: when increasing the over-completeness, one observes the emergence of different classes of filters, such as different positions

and edges at first and then a similar edge with different phases. Exploring the results for different dimensions of the dictionary may give an evaluation of the optimal complexity of the LGM to describe images in terms of a trade-off between accuracy and generality (see Annex. 5.3). Pushing this experiment to the extreme (that is when the over-completeness equals the size of the dictionary of signals), one would get a dictionary where every single signal from the database would be represented, the so-called grand-mother neurons. More generally, the architecture of the connections between cortical areas highly suggests that neural information is distributed, that this distribution is organized according to a hierarchical but also recursive architecture. An important feature is the generalization of the representation to common transformations (for an image a translation, a different non-uniform lighting, an occlusion,...) robustly to noise and interferences. This calls for the extension of this kind of approach to a more integrated multi-scale approach were events could be a more general bit of information, from a synaptic quanta, a spike (such as studied here), a burst in a cortical column or a coherent activation in a cortical area.

**Acknowledgments**

The author thanks the team at the Redwood Neuroscience Institute for stimulating discussions. Special thanks to Artemis Kosta for essential comments on this work.
This work was supported by a grant form the French Research Council (ANR "NatStats") and by EC IP project FP6-015879, "FACETS".

# 5 Supporting information

## 5.1 Annex: Computational implementation

The whole collection of simulation scripts were written with the intention of controlling and comparing the convergence of the algorithms and the relative effect of the different parameters. All scripts to reproduce the figures and supplementary material are available upon request on the author's website (see `http://incm.cnrs-mrs.fr/LaurentPerrinet/SparseHebbianLearning`). Version 1.5, rev. 666 and experiment 20080502T174325 are used for this paper, and other figures regarding control experiments may be found there. In particular, all figures except Fig. 1 were produced directly by the scripts without any editing: script `experiment_learn.m` for Fig. 2, script `experiment_nonhomeo.m` for Fig. 4 and script `experiment_code_efficiency.m` for Fig. 3. The original parameters of SparseNet were used for the CGF algorithm. Supplementary Figures may be consulted on the author's website (see `http://incm.cnrs-mrs.fr/LaurentPerrinet/SparseHebb`

Table 1: **Parameters used in the simulations.** Note that aSSC uses 4 parameters while CGF uses 9 parameters. Note also that the learning rate depends on the dimension of the dictionary. In fact, in higher dimensions, the non-linear correlation coefficients will be necessarily lower and thus the learning slower (see Eq. 12). These parameters were determined from the original parameters from (Olshausen and Field, 1998) and then by sequentially optimizing using the robustness analysis (see Annex. 5.5).

| Description | Value |
|---|---|
| experiment | $20080417T175704$ |
| dimension of imagelets (CGF & aSSC) | 256 |
| dimension of dictionary (CGF & aSSC) | 324 |
| number of learning steps (CGF & aSSC) | 32001 |
| learning rate (CGF) | 1 |
| learning rate (aSSC) | 0.1 |
| batch size (CGF & aSSC) | 100 |
| noise variance (CGF) | 0.017 |
| noise variance (aSSC) | 0.008 |
| homeostasis' learning rate (SSC) | 0.001 |
| homeostasis' learning rate (CGF) | 0.0025 |
| homeostasis smoothing rate (CGF) $\alpha$ | 0.02 |
| desired variance (CGF) $VAR - GOAL$ | 0.1 |
| prior steepness (CGF) *beta* | 0.2 |
| prior scaling (CGF) *sigma* | 0.1 |
| tolerance (CGF) *tol* | 0.0031 |

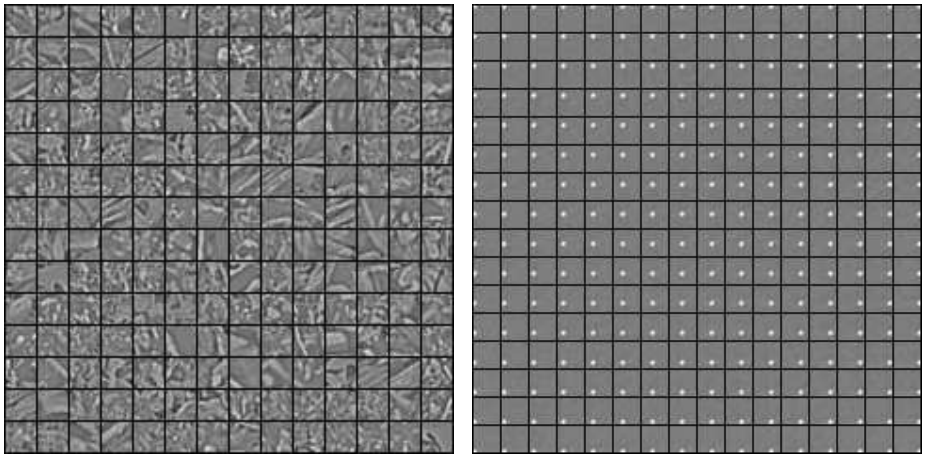

Figure 5: **Control of the statistics of the inputs**. *(Left)* One set of 324 $16 \times 16$ imagelets drawn from the database provided with SparseNet. We use the same presentation as Fig. 2. In order to have comparable patches, the images had to be fairly homogeneous (and therefore textured) since it is likely in natural images to draw a patch from a flat area (such as the sky) in which case the signal was poor and the convergence of the learning was slower. This was controlled by learning using different sets of images (not shown here). *(Right)* We show here a $16 \times 16$ matrix of $16 \times 16$ correlation values representing the covariance matrix for every pixel depicted in every box by showing the point-wise cross-correlation. This shows in every box the luminance's cross-correlation between 2 points: it is low (gray) compared to auto-correlation (white) when increasing the distance between both points to more than one pixel, validating the whitening hypothesis for the image's preprocessing. See script `experiment_stats_images.m` to reproduce the figure.

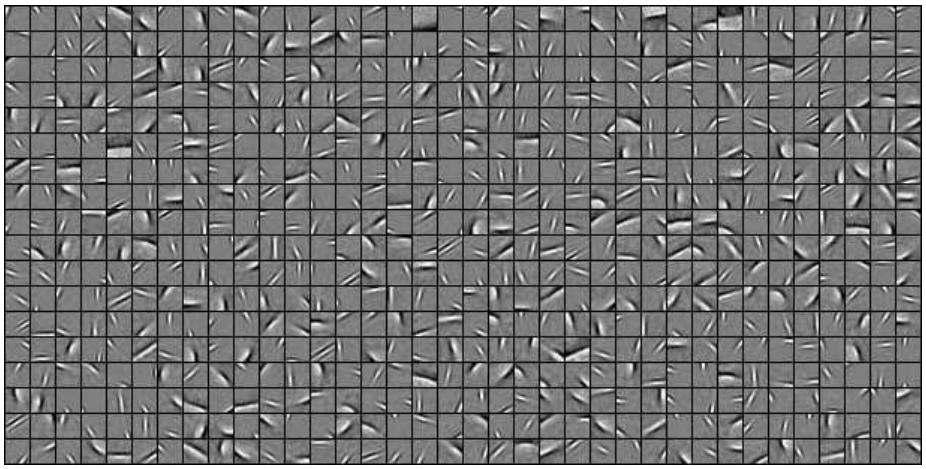

Figure 6: **Learning solution with non-negative coefficients**. When releasing the symmetry constraint, the learning algorithm converged to a similar set of filters. However, the convergence was quicker and proved to be of higher efficiency when the information for the polarity (that is, one bit) was rather used to double the size of the dictionary (efficiency measure not shown here). This suggested that the assumption of symmetry of the sign of the coefficients is not strictly true for the LGM and that a non-negative representation as implemented by (Lee and Seung, 1999) may be more efficient. See script `experiment_symmetric.m` to reproduce the figure.

## 5.2 Annex: Releasing the constraint of symmetry of filters

To compare our algorithm with SparseNet, we similarly assumed that in the dictionary, filters were symmetric (see Sec. 3.1). In fact, inspired by biology, receptive fields often coexist with opposite polarities (the so-called ON/OFF symmetry). This implied a constraint in the generative model that when looking for a match, the correlation could be positive or negative and therefore that the best match should be chosen as the greatest absolute value in Eq. 9. If we rather choose a dictionary of double the size and that we choose only the greatest values (that is not applying the absolute operator) we will obtain a system were each spike would have the same informational cost (the additional bit replacing the polarity bit from the symmetric case). We therefore look similarly to the non-negative representation without any further modification of the algorithm. The solution to the problem when releasing the symmetry constraint looked qualitatively similar but proved to be of slightly higher efficiency (see Fig. 6).

## 5.3 Annex: Over-completeness

We analyzed the effect of increasing the size of the dictionary, that is of increasing the complexity of the representation, on the qualitative receptive fields maps

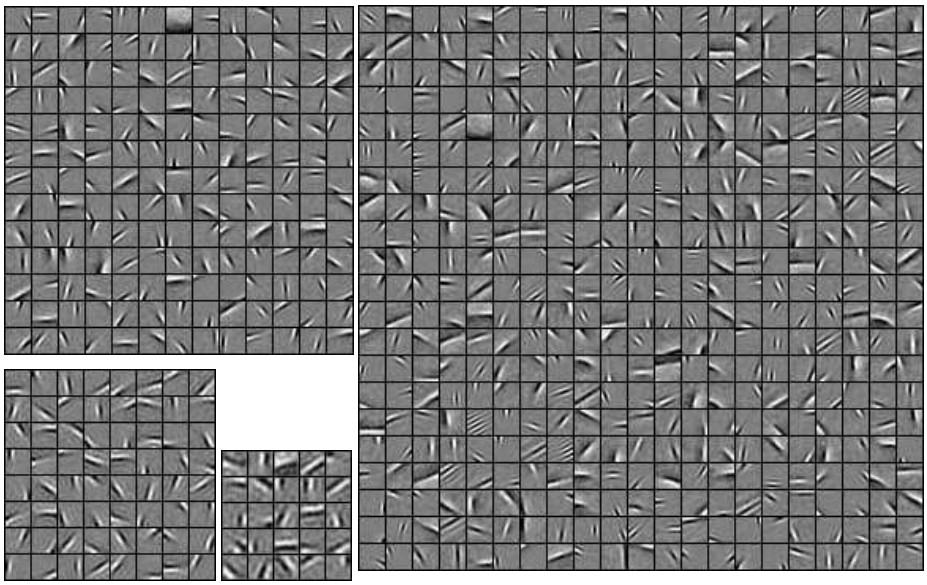

Figure 7: **Learning solution with an increasing over-completeness**. When increasing the number of coefficients from $8 \times 8$ (center,bottom), $13 \times 13$ (left,bottom), $21 \times 21$ (left, top) to $34 \times 34$ one sees that the complexity of the features represented by the filters progressively increases. However, only by using the biggest map, could one yield textured filters and "end-stopping" cells. This could be a result of a lack of convergence (we used the same number of steps for all experiments), but also the sign to a transition from representing edges and their different transforms (apparently from the lower-dimension map to the more complex position, scale, spatial frequency and phase). An interesting perspective is to do a cluster analysis on the distribution of the parameters of the best fit Gabors to observe bifurcations as a function of the complexity. See script `experiment_stability_oc.m` to reproduce the figure.

and on the efficiency. When increasing the over-completeness, one observes the emergence of different classes of filters, such as different positions and edges at first and then a similar edge with different phases (see Fig. 7). Exploring the results for different dimensions of the dictionary gave an evaluation of the optimal complexity of the LGM to describe imagelets in terms of a trade-off between accuracy and generality for the dimension that we use in this study (not shown).

## 5.4 Annex: Robustness to a perturbation

As an adaptive algorithm, we checked that the system returned to a similar macroscopic state after a perturbation. To illustrate that, we perturbed one filter (by re-initializing it to a random filter) and ran again the algorithm. The

first effect was that the corresponding gain function changed since the correlation coefficients values dropped for that particular neuron. As a consequence, the homeostatic constraint relatively "boosted" the correlation values of this neuron relative to the other neurons so that the choice of choosing any neuron was still uniform. After a few steps, the filter retrieved and edge-like shape which was often close to the feature prior to the perturbation, since this feature was momentarily "absent" from the representation dictionary. See script `experiment_perturb.m` to reproduce this experiment.

### 5.5 Annex: Robustness of the methods

We included in our computational framework the ability of exploring the evolution of the efficiency of one model when changing one single parameter around the operating point that was chosen over the experiences (see table in Annex. 5.1). This "perturbation analysis" allowed to trace if the chosen parameters were giving locally the best efficiency so that the comparison of two algorithms was valid. It also allows to identify the parameters which are the most relevant in the sense that small variations will induce bigger changes of efficiency. This was in particular true for the SparseNet algorithm. It showed in particular that SparseNet was more sensitive to parameters (including learning rate, homeostasis parameter) than our solution and that the parameters for tuning the parametric model (in particular the parameters $\beta$ and $\sigma$) were of particular importance. See scripts `experiment_stability_eta.m`, `experiment_stability_homeo.m` and to control SparseNet `experiment_stability_cgf.m` to reproduce the experiments.

## References


H. Akaike. A new look at the statistical model identification. *IEEE Transactions on Automatic Control*, 19:716–23, 1974.

Collins Assisi, Mark Stopfer, Gilles Laurent, and Maxim Bazhenov. Adaptive regulation of sparseness by feedforward inhibition. *Nature Neuroscience*, 10 (9):1176–1184, 2007. ISSN 1097-6256 (Print). doi: 10.1038/nn1947.

Joseph J. Atick. Could Information Theory Provide an Ecological Theory of Sensory Processing? *Network: Computation in Neural Systems*, 3(2):213–52, 1992. URL `http://ib.cnea.gov.ar/~redneu/atick92.pdf`.

Horace B. Barlow. Redundancy reduction revisited. *Network: Computation in Neural Systems*, 12:241—25, 2001.

P. Baudot, M. Levy, C. Monier, F. Chavane, A. René, N. Huguet, O. Marre, M. Pananceau, I. Kopysova, and Y. Fregnac. Time-coding, low noise vm attractors, and trial-by-trial spiking reproducibility during natural scene viewing in v1 cortex. In Society for Neuroscience Abstracts., editor, *34th Annual*

*Meeting of the Society for Neuroscience, San Diego, USA.*, pages 948–12, 2004.

James A Bednar, Amol Kelkar, and Risto Miikkulainen. Scaling self-organizing maps to model large cortical networks. *Neuroinformatics*, 2(3):275–302, 2004.

Anthony J. Bell and Terrence J. Sejnowski. The 'Independent Components' of Natural Scenes are Edge Filters. *Vision Research*, 37(23):3327–38, 1997.

J Bolz and CD Gilbert. The role of horizontal connections in generating long receptive fields in the cat visual cortex. *European Journal of Neuroscience*, 1 (3):263–8, 1989.

Jean Bullier. Integrated model of visual processing. *Brain Research Reviews*, 36: 96–107, 2001. URL `http://dx.doi.org/10.1016/S0165-0173(01)00085-6`.

Matteo Carandini, J. Heeger, and Anthony Movshon. Linearity and normalization in simple cells of the macaque primary visual cortex. *Journal of Neuroscience*, 17(21):8621—44, November 1997.

Matteo Carandini, Jonathan B. Demb, Valerio Mante, David J. Tolhurst, Yang Dan, Bruno A. Olshausen, Jack L. Gallant, and Nicole C. Rust. Do we know what the early visual system does? *Journal of Neuroscience*, 25 (46):10577–97, Nov 2005. doi: 10.1523/JNEUROSCI.3726-05.2005. URL `http://dx.doi.org/10.1523/JNEUROSCI.3726-05.2005`.

B. Cessac and M. Samuelides. *Topics in Dynamical Neural Networks: From Large Scale Neural Networks to Motor Control and Vision*, chapter From neuron to neural networks dynamics, pages 7–88. Volume 142 of *The European Physical Journal (Special Topics)*, Cessac et al. (2007), mar 2007. doi: 10.1140/epjst/e2007-00061-7. URL `http://lanl.arxiv.org/pdf/nlin/0609038`.

Bruno Cessac, Emmanuel Daucé, Laurent U. Perrinet, and Manuel Samuelides. *Topics in Dynamical Neural Networks: From Large Scale Neural Networks to Motor Control and Vision*, volume 142 of *The European Physical Journal (Special Topics)*. Springer Berlin / Heidelberg, mar 2007. doi: 10.1140/epjst/e2007-00061-7. URL `http://www.springerlink.com/content/q00921n9886h/?p=03c19c7c204d4fa78b850f88b97da2f7&pi=0`.

M. Deweese and A. Zador. Binary coding in auditory cortex. *Journal of Neuroscience*, 23(21), August 2003.

Eizaburo Doi, Doru C. Balcan, and Michael S. Lewicki. Robust coding over noisy overcomplete channels. *IEEE Transactions in Image Processing*, 16(2): 442–52, 2007. URL `http://www.hubmed.org/display.cgi?uids=17269637`.

David J. Field. What is the goal of sensory coding? *Neural Computation*, 6(4): 559–601, 1994.

Sylvain Fischer, Rafael Redondo, Laurent U. Perrinet, and Gabriel Cristóbal. Sparse approximation of images inspired from the functional architecture of the primary visual areas. *EURASIP Journal on Advances in Signal Processing*, pages Article ID 90727, 16 pages, 2007. doi: doi:10.1155/2007/90727. URL `http://www.hindawi.com/GetArticle.aspx?doi=10.1155/2007/90727&e=cta`. Special issue on Image Perception.

C. Fyfe and R. Baddeley. Finding compact and sparse- distributed representations of visual images. *Network: Computation in Neural Systems*, 6:333–44, 1995. URL `citeseer.nj.nec.com/fyfe95finding.html`.

R. Gribonval and P. Vandergheynst. On the exponential convergence of matching pursuits in quasi-incoherent dictionaries. *IEEE Transactions in Information Theory*, pages 255– –61, jan 2006. doi: 10.1109/TIT.2005.860474.

F H Hamker and J Wiltschut. Hebbian learning in a model with dynamic rate-coded neurons: an alternative to the generative model approach for learning receptive fields from natural scenes. *Network: Computation in Neural Systems*, 18(3):249–266, Sep 2007. doi: 10.1080/09548980701661210. URL `http://www.hubmed.org/fulltext.cgi?uids=17926194`.

KD Harris. Stability of the fittest: organizing learning through retroaxonal signals. *Trends in Neurosciences*, 31(3):130–136, 2008. doi: 10.1016/j.tins.2007.12.002. URL `http://andromeda.rutgers.edu/~kdharris/backprop.pdf`.

Donald O. Hebb. *The organization of behavior: A neuropsychological theory*. Wiley, New York, 1949.

Toshihiko Hosoya, Stephen A Baccus, and Markus Meister. Dynamic predictive coding by the retina. *Nature*, 436(7047):71–7, Jul 2005. doi: 10.1038/nature03689. URL `http://dx.doi.org/10.1038/nature03689`.

A. Hyvärinen, P. O. Hoyer, and M. Inki. Topographic Independent Component Analysis. *Neural Computation*, 13(7):1527–58, 2001.

C. Keysers, D. Xiao, Peter Földiák, and D.I. Perret. The speed of sight. *Journal of Cognitive Neuroscience*, (in press), 2000.

L. Lapicque. Recherches quantitatives sur l'excitation électrique des nerfs traitée comme une polarisation. *Journal of Physiology (Paris)*, 9:620–35, 1907.

S. B. Laughlin. A simple coding procedure enhances a neuron's information capacity. *Z. Naturforsch.*, 9–10(36):910–2, 1981.

D. D. Lee and H. S. Seung. Learning the Parts of Objects by Non-Negative Matrix Factorization. *Nature*, 401:788–91, 1999.

Honglak Lee, Alexis Battle, Rajat Raina, and Andrew Ng. Efficient sparse coding algorithms. In B. Schölkopf, J. Platt, and T. Hoffman, editors, *Advances in Neural Information Processing Systems 19*, pages 801–808. MIT Press, Cambridge, MA, 2007.

Michael S. Lewicki and Terrence J. Sejnowski. Learning Overcomplete Representations. *Neural Computation*, 12(2):337–65, 2000. URL `citeseer.nj.nec.com/lewicki98learning.html`.

Zhaoping Li. Theoretical understanding of the early visual processes by data compression and data selection. *Network: Computation in Neural Systems*, 17(4):301–334, 2006.

Wei Ji Ma, Jeffrey M. Beck, Peter E. Latham, and Alex Pouget. Bayesian inference with probabilistic population codes. *Nature Neuroscience*, 2006.

David J. C. MacKay. *Information Theory, Inference, and Learning Algorithms*. Cambridge University Press, 2003. URL `http://www.inference.phy.cam.ac.uk/mackay/itila/`.

Stéphane Mallat. *A wavelet tour of signal processing*. Academic Press, 1998.

Stéphane Mallat and Zhifeng Zhang. Matching Pursuit with Time-Frequency Dictionaries. *IEEE Transactions on Signal Processing*, 41(12):3397–3414, 1993.

Bruno A. Olshausen. *Probabilistic Models of the Brain: Perception and Neural Function*, chapter Sparse Codes and Spikes, pages 257–72. MIT Press, 2002.

Bruno A. Olshausen and David J. Field. Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature*, 381(6583): 607–9, jun 1996.

Bruno A. Olshausen and David J. Field. Sparse Coding with an Overcomplete Basis Set: A Strategy Employed by V1? *Vision Research*, 37:3311–25, 1998.

Bruno A. Olshausen and K. J. Millman. Learning sparse codes with a mixture-of-gaussians prior. In Michael I. Jordan, Michael J. Kearns, and Sara A. Solla, editors, *Advances in neural information processing systems*, volume 12, pages 887–93. The MIT Press, Cambridge, MA, 2000. URL `citeseer.nj.nec.com/olshausen00learning.html`.

Stefano Panzeri, Alessandro Treves, Simon Schultz, and Edmund T. Rolls. On Decoding the Responses of a Population of Neurons from Short Time Windows. *Neural Computation*, 11(7):1553–1577, 1999.

Arthur E. C. Pece. The problem of sparse image coding. *Journal of Mathematical Imaging and Vision*, 17:89–108, 2002.

Laurent Perrinet. Adaptive sparse spike coding : applications of neuroscience to the compression of natural images. In Gabriel Cristóbal Frédéric Truchetet Peter Schelkens, Touradj Ebrahimi, editor, *Optical and Digital Image Processing Conference 7000 - Proceedings of SPIE Volume 7000, 7 - 11 April 2008*, pages 15 – S4, 2008. URL http://incm.cnrs-mrs.fr/LaurentPerrinet/Publications/Perrinet08spie.

Laurent U. Perrinet. Finding Independent Components using spikes : a natural result of hebbian learning in a sparse spike coding scheme. *Natural Computing*, 3(2):159–75, January 2004a. doi: 10.1023/B:NACO.0000027753.27593.a7. URL http://incm.cnrs-mrs.fr/LaurentPerrinet/Publications/Perrinet04nc.

Laurent U. Perrinet. Feature detection using spikes : the greedy approach. *Journal of Physiology (Paris)*, 98(4-6):530–9, July-November 2004b. doi: 10.1016/j.jphysparis.2005.09.012. URL http://hal.archives-ouvertes.fr/hal-00110801/en/.

Laurent U. Perrinet. Efficient Source Detection Using Integrate-and-Fire Neurons. In W. Duch et al., editor, *ICANN 2005, LNCS 3696*, volume 3696 of *Lecture Notes in Computer Science*, pages 167–72, Berlin Heidelberg, 2005. Springer. doi: 10.1007/11550822_27. URL http://incm.cnrs-mrs.fr/LaurentPerrinet/Publications/Perrinet05icann.

Laurent U. Perrinet. *Topics in Dynamical Neural Networks: From Large Scale Neural Networks to Motor Control and Vision*, chapter Dynamical Neural Networks: modeling low-level vision at short latencies, pages 163–225. Volume 142 of *The European Physical Journal (Special Topics)*, Cessac et al. (2007), mar 2007. doi: 10.1140/epjst/e2007-00061-7. URL http://incm.cnrs-mrs.fr/LaurentPerrinet/Publications/Perrinet06.

Laurent U. Perrinet. Apprentissage hebbien d'un reseau de neurones asynchrone a codage par rang. Technical report, Rapport de stage du DEA de Sciences Cognitives, CERT, Toulouse, France, 1999. URL http://www.risc.cnrs.fr/detail_memt.php?ID=280.

Laurent U. Perrinet, Arnaud Delorme, Simon Thorpe, and Manuel Samuelides. Network of integrate-and-fire neurons using Rank Order Coding A: how to implement spike timing dependant plasticity. *Neurocomputing*, 38–40(1–4): 817–22, 2001.

Laurent U. Perrinet, Manuel Samuelides, and Simon Thorpe. Sparse spike coding in an asynchronous feed-forward multi-layer neural network using Matching Pursuit. *Neurocomputing*, 57C:125–34, 2002. URL http://incm.cnrs-mrs.fr/LaurentPerrinet/Publications/Perrinet02sparse. Special issue: New Aspects in Neurocomputing: 10th European Symposium on Artificial Neural Networks 2002 - Edited by T. Villmann.

Laurent U. Perrinet, Manuel Samuelides, and Simon Thorpe. Emergence of filters from natural scenes in a sparse spike coding scheme. *Neurocomputing*, 58–60(C):821–6, 2003. URL `http://incm.cnrs-mrs.fr/perrinet/publi/perrinet03.pdf`. Special issue: Computational Neuroscience: Trends in Research 2004 - Edited by E. De Schutter.

Laurent U. Perrinet, Manuel Samuelides, and Simon Thorpe. Coding static natural images using spiking event times : do neurons cooperate? *IEEE Transactions on Neural Networks, Special Issue on 'Temporal Coding for Neural Information Processing'*, 15(5):1164–75, September 2004. ISSN 1045-9227. doi: 10.1109/TNN.2004.833303. URL `http://hal.archives-ouvertes.fr/hal-00110803/en/`.

M. Ranzato, C.S. Poultney, S. Chopra, and Y. LeCun. Efficient learning of sparse overcomplete representations with an energy-based model. In Scholkopf et al., editor, *Advances in neural information processing systems*, volume 19. The MIT Press, Cambridge, MA, 2007.

M Rehn and FT Sommer. A model that uses few active neurones to code visual input predicts the diverse shapes of cortical receptive fields. *Journal of Computational Neuroscience*, 22(2):135–46, 2007.

J. Rissanen. Modeling by shortest data description. *Automatica*, 14:465–471, 1978.

Franck Rosenblatt. Perceptron simulation experiments. *Proceedings of the I. R. E.*, 20:167–192, 1960.

E. Smith and M. S Lewicki. Efficient auditory coding. *Nature*, 439(7079):978–82, 2006.

M.V. Srinivasan, S.B. Laughlin, and A Dubs. Predictive coding: A fresh view of inhibition in the retina. *Proc. R. Soc. London Ser.B*, 216:427–59, 1982.

T. W. Troyer, A. Krukowski, N. J. Priebe, and K. D. Miller. Contrast-invariant orientation tuning in cat visual cortex: Feedforward tuning and correlation-based intracortical connectivity. *Journal of Neuroscience*, 18:5927, 1998.

J.H. van Hateren. Spatiotemporal contrast sensitivity of early vision. *Vision Research*, 33:257–67, 1993.

Rufin van Rullen and Simon J. Thorpe. Rate Coding Versus Temporal Order Coding: What the Retina Ganglion Cells Tell the Visual Cortex. *Neural Computation*, 13(6):1255–83, 2001.

William E. Vinje and Jack L. Gallant. Sparse Coding and Decorrelation in Primary Visual Cortex During Natural Vision. *Science*, 287:1273–1276, 2000.

Michael Zibulevsky and Barak A. Pearlmutter. Blind Source Separation by Sparse Decomposition in a Signal Dictionary. *Neural Computation*, 13(4):863–82, 2001. URL `citeseer.nj.nec.com/article/zibulevsky99blind.html`.