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

Laurent Perrinet

Optimal signal representation in neural spiking population codes: 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

March 19, 2008

Abstract

Taking advantage of the constraints of spiking representations, we derive an unsupervised learning algorithm which we prove to efficiently code natural images and apply it to a model of the input to the primary visual cortex. In fact, spikes carry temporal event-based information in bundles of parallel fibers and may be considered as all-or-none binary events. This property may be used to formulate the efficiency of a representation problem as finding the L_0-norm sparsest representation, a "hard" NP-complete problem. We propose a solution for a bundle of Integrate-and-Fire neurons which improves previous results based on an Adaptive Matching Pursuit scheme by explicitly implementing an homeostatic constraint in the choice function by a spiking gain control mechanism in the neural population. For comparison purposes, we applied this scheme to the learning of small images taken from natural images as in SparseNet and compared the results and efficiency of this last algorithm with Matching Pursuit and the proposed algorithm. Results show that the different coding algorithm give similar efficiencies while the homeostasis provided an optimal balance which was crucial during the learning. This study provides a simpler and more efficient algorithm for learning independent components in a set of inputs such as natural images suggesting that this Sparse Spike Coding strategy may provide a generic computational module that help us understanding the efficiency of the Primary Visual Cortex.

Keywords

Neural population coding, spike-event computation, correlation-based inhibition, Sparse Spike Coding, Adaptive Matching Pursuit, 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. In particular, models for the formation of simple cell receptive fields in the primary visual cortex (V1) have attracted great attention as a model of learning applied to vision and more generally as a generic model of coding and representation in neural computations. Based on the functional approach that the system should evolve to be efficient, that is for low-level sensory areas that information is transformed efficiently [1, 2], the most accepted explanation for the formation of orientation selective simple cells in V1 is that it optimizes the sparseness of the representation of images drawn from natural scenes, that is from behaviorally relevant scenes [3]. Similar approaches have been followed for natural images [4, 5, 6, 7, 8, 9] and sounds [10, 11] that were based on solving the inverse of a generative model of the signal. However, all of these solutions relied on specific parameterizations and didn't explicitly demonstrated how their algorithm could be specifically adapted to neural computations. For instance, the coding was achieved by conjugate gradient [3] or orthogonal matching pursuit [7] without explicitly addressing the problem of the existing representational constraints in the cortex. More specificaly, they don't specifically take advantage of the nature and architecture of neural computations that make them different from the one occuring in a traditional sequential computer.

In that direction, a key aspect of neural communication 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 presenting a brief visual stimulus 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 approx. 35 ms [13]. 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 the efficient spike coding of a flashed static image and derive a measure attached to the performance of information transmission in the neural assembly by intro-

[1] Spikes have a shape of around 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 [12]. They are universally present in the central nervous system but also across species and phylogeny.

ducing the L_0-norm as a measure of the sparseness of the spike code. Based on previous results [14], 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 computational balance within the assembly, we will improve the performance of the previously proposed algorithm and derive a simple hebbian-type learning scheme on the sparse representation. We will finally compare the proposed algorithm with standard methods: SPARSENET [3] and Adaptive Matching Pursuit [15] and show the relative importance of the coding scheme and of the homeostasis on the resulting systems thanks to the quantitative measures of efficiency. We will conclude by comparing this method with previously proposed schemes and how this 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.

1.1 A generative model of signal synthesis

In low-level sensory areas, the goal of neural computations is to build efficient intermediate *representations* to allow efficient decision making [1, 16]. 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, 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) [17] 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 chosen 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 σ_n^2. This noise model is achieved thanks to the preprocessing (which could be achieved in general by Principal Component Analysis) without loss of generality since the processing is invertible [18] (see Fig. 5). The LGM is well adapted to natural scenes because transparency laws are linear for luminances 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 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 solution for one input. We will see in Sec. 1.3 how we may quantify the global efficiency of the coding, 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 [18, 19], 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 $s_j.\mathbf{A}_j$ knowing it is a realization of the LGM as it is defined in Eq. 1 (and for which we assume no prior knowledge) is maximal for the projection coefficient defined by:

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

where $\overset{\text{def}}{=}$ means "equal by definition". The log-likelihood $\log P(\{s_j\}|\mathbf{x},\mathbf{A})$ is then maximum for the source j^* with maximal correlation coefficient $j^* = \text{ArgMax}_j \rho_j$ with

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

It should be noted that ρ_j is the M^{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 2π) to zero if and only if $\rho_j = 1$ (full correlation), π 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, it is independent to the norm of the filters and we assume without loss of generality in the rest that these are normalized to unity. 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 [20] as it provides a direct measure of the log-probability under the assumptions that we used (the LGM with Gaussian noise). Starting from this basic mechanism, one could compute for every signal a set of activities corresponding to how well the neurons corresponded to patterns in the image predefined in the weights matrices. However, we should now explain how this information may be coded and decoded by a set of spiking neurons.

1.2 Spike coding and decoding of a transient signal in a population of neurons

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 pyramidal neurons, we may use the fact that the larger the driving excitation, the larger the firing frequency and dually the shorter the latency of spiking [21]. More precisely, let's consider a population of N pyramidal neurons as an information channel for which we wish to code and then decode a vector $\{\rho_j\}_{1\leq j\leq N}$ only by transmitting a spiking pattern over the assembly. Classically, one would map each value to an excitation value which corresponds through a monotonous increasing function to a spiking latency or frequency, which can then be decoded by the corresponding inverse function. 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 systematically different. In the competitive network formed with the pyramidal cells, this is in disagreement with the 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 system where activity is uniformly distributed. 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[2].

[2] 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.

A standard method to achieve this homeostasis is to map the input vector $\{\rho_j\}$ trough a point non-linearity[3] which provides a uniform probability for the output [2]. 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 [22] or dually the minimization of intrinsic noise [23]. It may be easily derived from the probability P of variable ρ_j (bounded in absolute value by 1) by choosing the non-linearity as the cumulative function

$$f_j(\rho_j) = \int_{-1}^{\rho_j} dP(\rho) \tag{4}$$

where the symbol $dP(x) = P_X(x)dx$ will here denote in general the probability distribution function (pdf) for the random variable X. This process has been observed in a variety of species and is for instance perfectly illustrated in the salamander [24] (see Fig. 1). It may evolve dynamically to slowly adapt to varying changes in luminances, such as when the light diminishes at dawn but also to some more elaborated scheme within a map [25]. As in "ideal democracies" where all neurons are "equal", this process has to be dynamically updated over some characteristic period so as to achieve optimum balance. As a consequence, since for all j, the pdf of $z_j = f_j(\rho_j)$ is uniform and that sources are independent, it 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 $\{\rho_j\}$ 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.

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 $\{\rho_j\}$ 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, we know in particular that for the first spike to arrive at the receiver end, knowing that it corresponds to fiber $j(1)$, has been produced by a value in the highest quantile of $\rho_{j(1)}$ on the emitting side. We may therefore decode the corresponding value with the best estimate $\hat{\rho}_{j(1)} = f_{j(1)}^{-1}(1)$. This is also true for the following spikes and if we write as $z_{j(k)} = \frac{k}{N}$ the relative rank of the spike (that is neuron $j(k)$ fired at rank k), we can reconstruct the corresponding value as

$$\hat{\rho}_{j(k)} = f_{j(k)}^{-1}(1 - \rho_{j(k)}) \tag{5}$$

This corresponds to a generalized rank coding scheme [28, 29] (see Fig. 1, Top Right). 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

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

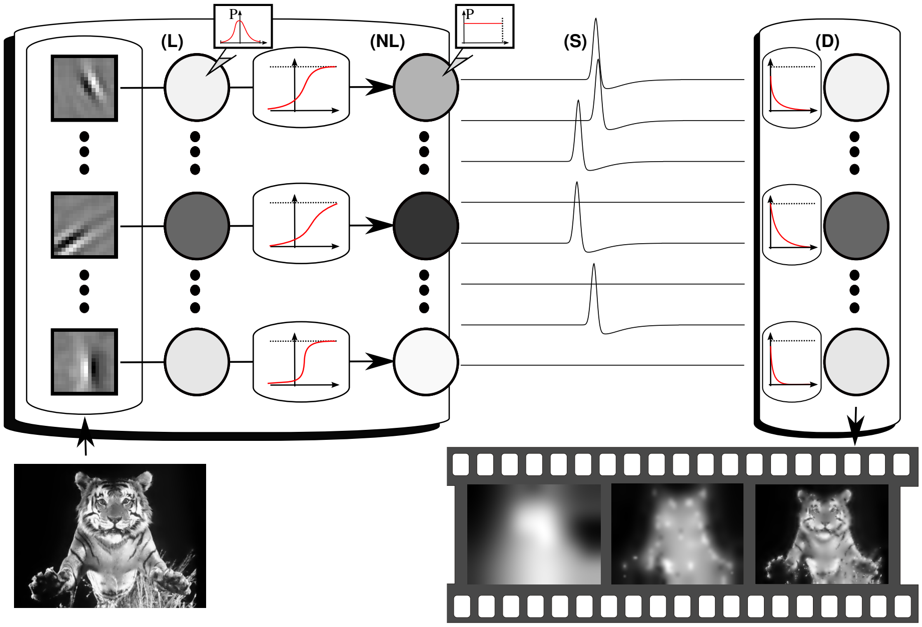

Figure 1: **Spike coding channel using homeostatic gain control.** We show here how a bundle of L-NL neurons [26, 27] tuned by a simple homeostatic mechanism allow to transfer a transient information, such as an image, using spikes. (L) The signal to be coded, for instance the match ρ_j of an image patch (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 ρ_j to be represented. (NL) By using the cumulative function as a point non-linearity f_j, one ensures that the probability of $z_j = f_j(\rho_j)$ 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 [7, 10, 11], we will use the framefork used in SparseNet [17].

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 (which is thanks to Stirling's approximation of order $\log_2(N!) = O(N.\log(N))$, that is more than 2000 bits for 256 neurons) is greater than the information contained in the particular quantization of the image[4]. 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 seems that this information would have a low impact relative to the total information [30]. More generally, one could use different mappings for the transformation of the z value into the a 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 [31] and is mathematically more tractable. In particular, one may show that the coding error is proportional to the variability of the sorted coefficients [21], 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 on 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.

1.3 Definition of the efficiency of Spike coding

Now that we defined the spike coding algorithm, we should be able to derive a generic cost function that will allow us to quantify the efficiency of different

[4] We are generally unable to detect quantization errors on an image consisting of more 256 gray levels, that is for 8 bits.

coding algorithms but also to derive a learning algorithm for the spike coding algorithm defined above. 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 different representations [32, 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 Shannon's information of solutions $\hat{\mathbf{s}}$ (the coding sequences) given the model's parameters. A goal is therefore to maximize the evidence of the model that is to minimize this information or similarly minimize the description length [33]. 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. 1.1, the total representational cost is $\mathcal{C} \overset{\text{def}}{=} E(-\log P(\hat{\mathbf{s}}|\mathbf{x}, \mathbf{A}))$, where $E(.)$ denotes averaging over multiple images. 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:

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

where Z is the partition function (that we will omit in the sequel). 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 sequel). For any coding $\hat{s}$, the first term corresponds to the information from the image which was not retrieved by the coding (reconstruction cost) and that can be encoded at best using entropic coding pixel by pixel. 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 $\hat{\mathbf{s}}$ knowing its probability distribution function.

We will assume independence of the coefficients of the LGM 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, this yields the sparseness cost defined in Olshausen and Field [17]:

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

where β is the steepness of the prior and σ is the prior scaling (see Figure 13.2 from [34]). 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 coefficients to be sparse, the "occam factor" measuring the evidence of the model knowing this prior.

This liberty in the definition of the sparseness leads to a wide variety of proposed solutions to optimizing this cost or *sparse coding* (for a review, see [35]) such as numerical optimization [17, 36], non negative matrix factorization [37, 38] or by using Matching Pursuit [7, 11]. Note that this cost will efficiently quantify the representation efficiency if the pdf of the coefficients is indeed well fitted to the parameterization. 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. 1.2), 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 [39] as was presented above for the Sparse Spike Coding algorithm. 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 :

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

where $\|\hat{\mathbf{s}}\|_0$ 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 [7] should include a correction term in the L_0-norm to account for this non-unifomity. It also explictely rates the economy of consumed metabolic ressources as is used in [7], but we retain this only as a consequence of the algorithm. Note first that for any spike coding solution, this cost function is dynamic since the number of spikes may increase in time. Note also that it now links efficiency to sparseness only on the basis that the spiking representation is binary 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 [40] or distorition rate [41, 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. 3 (in a nutshell, sparser representation make "peaker" cross-correlograms). However, resolving the coding problem with the L_0 norm (getting the best $\hat{s}$ in the sense of Eq. 8 knowing $\mathbf{x}$, that is $\mathrm{ArgMin}_s(\mathcal{C}_0(\mathbf{x}, s))$) is *NP-complete* with respect to the dimension N of the dictionary [41, p. 409]. We will present here a solution to this problem inspired by the architecture and dynamics of the primary visual cortex.

2 Method : adaptive Sparse Spike Coding (aSSC)

In fact, unless the dictionary is orthogonal, when choosing one component over an other (for instance the one that maximizes Eq. 3), any choice may modify the choice of the other components. If we chose the successive neurons with maximum correlation values, the resulting representation will be proportionally more redundant when the dictionary gets more over-complete. Also we saw that overcoming this inefficiency 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.

2.1 Sparse Spike Coding: Adaptive Matching Pursuit with egalitarian homeostasis

In fact, as was first proposed in the *Sparse Spike Coding* (SSC) [14], one could use a greedy algorithm on the L_0-norm cost and that these led to use of Matching Pursuit algorithm [42]. It is defined as the greedy approach applied on the efficiency criterion defined in Eq. 8. More generally, let's first define Weighted Matching Pursuit (WMP) by introducing a non-linearity in the choice step. Like Matching Pursuit, it is based on two repetitive steps. First, given the signal $\mathbf{x}$, we are searching for the *single* source $s^*_{j^*}.\mathbf{A}_{j^*}$ that corresponds to the maximum *a posteriori* (MAP) realization for $\mathbf{x}$ (see Eq. 3) transformed by a point non-linearity f_j. This Matching step is defined by:

$$j^* = \mathrm{ArgMax}_j[f_j(\rho_j)] \tag{9}$$

where $f_j(.)$ is some gain function that we will describe below and which may be set initially to strictly increasing functions and ρ_j is initialized by Eq. 3. In a second step (Pursuit), the information is fed-back to correlated sources through :

$$\mathbf{x} \leftarrow \mathbf{x} - s^*_{j^*}.\mathbf{A}_{j^*} \tag{10}$$

where $s^*_{j^*}$ is the scalar projection $< \mathbf{x}, \mathbf{A}_{j^*} >$ (see Eq. 2). Equivalently, from the linearity of the scalar product, we may propagate laterally:

$$< \mathbf{x}, \mathbf{A}_j > \leftarrow < \mathbf{x}, \mathbf{A}_j > - < \mathbf{x}, \mathbf{A}_{j^*} >< \mathbf{A}_{j^*}, \mathbf{A}_j > \tag{11}$$

that is from Eq. 3:

$$\rho_j \leftarrow \rho_j - \rho_{j^*} < \mathbf{A}_{j^*}, \mathbf{A}_j > \tag{12}$$

For any set of monotonously increasing functions f_j, WMP 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/decoding algorithm which uses WMP as the coder and where the point non-linearities are defined by Eq. 4. As described in [18], while the Matching step is efficiently performed by the LIF neurons driven by the NL input (see Fig. 1), the pursuit step could be implemented in a cortical area by a correlation-based inhibition. This type of inhibition is typical of fast-spiking interneurons though there is no direct evidence of this activity-based synaptic topology. In Fig. 1, it will correspond to a lateral interaction within the linear (L) neuronal population. In practice, the f_j functions are initialized for all neurons to the identity function (that is to a MP algorithm) and then evaluated 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. This scheme extends the Matching Pursuit (MP) algorithm by linking it to a statistical model which tunes optimally the matching step (in the sense that all choices are statistically equally probable) thanks to the adaptive point linearity. In fact, as stated before, thanks to the uniform distribution of the choice of a component, one maximizes the entropy of every match and therefore of 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. It therefore optimizes the efficiency of MP for this problem by deriving it as the greedy solution of the cost defined in Eq. 8.

2.2 Introducing Hebbian Learning in SSC

On a longer time scale, the efficiency of the system may be optimized by slowly adapting the dictionary as in SPARSENET thanks to the sparse solution given by the coding algorithm. We may implement this for every image at every coding step since we have for each selected spike an evaluation of the log-likelihood by the distance of the residual image to the selected filter, that is to $\|\mathbf{x} - s^*_{j^*}.\mathbf{A}_{j^*}\|^2$ (which is equal to ρ_j up to a constant), the rest of the signal being regarded as a perturbation which will cancel out by the averaging. At every step after Eq. 9 and using the gradient descent approach as in [17], we similarly infer that we may slowly modify the winning weight vector corresponding to the winning filter $\mathbf{A}_{j^*}$ by taking it closer to $\frac{\mathbf{x}}{s^*_{j^*}}$:

$$\frac{\partial \mathcal{C}}{\partial \mathbf{A}_{j^*}} = \frac{\partial}{\partial \mathbf{A}_{j^*}} \frac{1}{2\sigma_n^2}.\|\mathbf{x} - \hat{s}^*_{j^*}.\mathbf{A}_{j^*}\|^2 \qquad (13)$$

$$= \frac{1}{2\sigma_n^2}.\hat{s}^*_{j^*}(\mathbf{x} - \hat{s}^*_{j^*}.\mathbf{A}_{j^*}) \qquad (14)$$

that is noting $s^* = \hat{s}^*_{j^*}$:

$$\mathbf{A}_{j^*} \leftarrow \mathbf{A}_{j^*} + \eta s^* (\mathbf{x} - s^* . \mathbf{A}_{j^*}) \tag{15}$$

where η is the learning rate, which is inversely proportional to the time scale of the features being learned. It is an "hebbian" rule [43] in the classical sense since it will enhance the weight of neurons of correlated neurons. However, the novelty of this formulation is to apply this formulation to the sparse representation. Similarly to Eq. 17 in [17] or to Eq. 2 in [11] the relation is linear. A more rigorous mathematical approach were to consider a rotation of $\mathbf{A}_j$ toward $\mathbf{x}$ using a Jacobi Matrix rotation so that all component vectors stay on the unit sphere. In practice, Eq. 15 for small learning rates η followed by a normalization is a good approximation of this high-dimensional (linear) transform. Another improvement is "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 Orthogonal Matching Pursuit as was chosen as the coding behind the learning algorithm used in [7]. 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 was selected at first and will modify the first winning filter. Then the same neuron will be selected with a higher probability in subsequent learning steps, causing a 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 independent to the norm of the filters. However, thanks to the homeostatic regulation (which has a similar time-scale than the learning) the probability of choosing any neuron in WMP remains uniform and ensures the convergence of the learning algorithm (see Annex. 5.4). The homeostasis will therefore optimize the balance between the neurons, the homeostasis constraint 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 in filter space (the unit N-dimensional sphere) corresponding to the dictionary: each centroid will (as in the K-means algorithm) be attracted toward the mean vector of inputs from its corresponding Voronoi cell. 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[5]. As a consequence, at convergence the f_j functions become equal 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) the algorithm will change the whole dictionary so that it settles to a new stable point (see Annex. 5.4 for supplementary information). 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.

2.3 Adaptive Sparse Spike Coding (aSSC)

In summary, the solution of the coding problem 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. draw a signal $\mathbf{x}$ from the database,
3. compute ρ_j for all j using Eq. 3,
4. until $\|\mathbf{x}\|^2$ is below a threshold do sparse spike coding (SSC):
    (a) select the best match j^* with Eq. 9,
    (b) modify correlated information by updating ρ_j for all j using Eq. 12,
    (c) slowly modify $\mathbf{A}_{j^*}$ using Eq. 15,
5. then update the f_j for all j and draw a new image (step 2)

When convergence is achieved, one could simply make a coding by using steps 2, 3 and 4 and optionally for the pure spike coding evaluate the coefficient using Eq. 5 in step 4-b. In fact, since the greedy algorithm may adapt to quantization errors [21, Fig.10]. The decoding of a spike list $\{j(1), j(2), \ldots\}$ is then simply:

1. Initialize $\hat{\mathbf{x}}$ to a zero image; the rank k is one,
2. while we have spikes do :
    (a) retrieve the address $j(k)$ of the spike and the corrseponding value $\hat{s}$ of the coefficient using Eq. 5,
    (b) add $\hat{s}.\mathbf{A}_{j(k)}$ to $\hat{\mathbf{x}}$,
    (c) increment the rank k,

[5] In fact, this is true if the choice of filters is uniform, in general it will form a tiling corresponding to the *a priori* distribution of features in the filter space. Therefore, it is more correct to say that the algorithm settles when the probability of any centroid is uniform.

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 simulations 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 [18].

3 Results on natural images

3.1 Comparison with SparseNetof receptive field formation

We compared this novel aSSC algorithm with the SparseNet algorithm. In fact, this algorithm as other similar schemes mainly 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 [17] and used in particular the same database of inputs as the SparseNet algorithm and restrict ourselves to study the selection of optimal filters on imagelets (that is small patches from natural images)[7]. In particular, these images are static, grayscale and filtered according to similar parameters to allow a one-to-one comparison of the different algorithms.

Here, we show the results for 16 × 16 patches (so that $M = 256$) from the whitened images and we chose to learn $N = 324$ filters. 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 represent any salient feature (such as a sharp 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, 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

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

[7] This will give a lower bound for the efficiency of aSSC, since it is known that sparseness in natural images is greater than in imagelets since sparseness is also spatial. For instance, it is highly probable in natural images that large parts of the space —such as the sky— are flat. See [19, Sec. 3.3.4] for an extension to whole images.

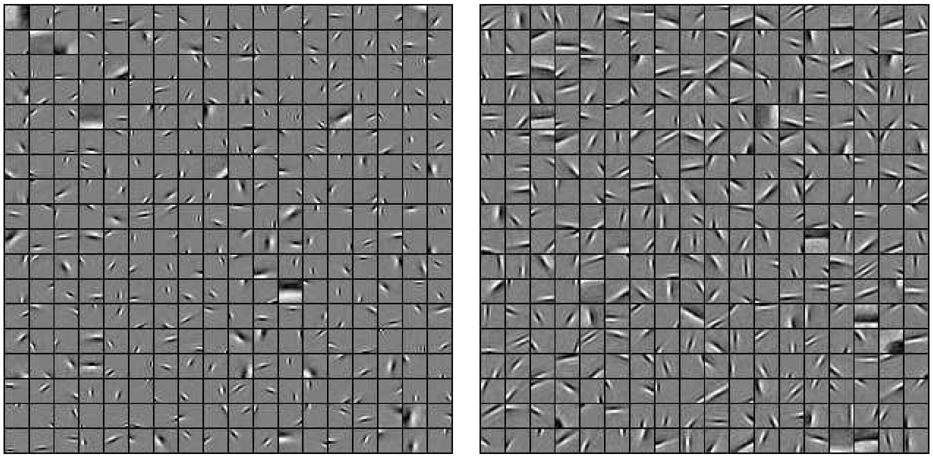

Figure 2: **Results of the proposed aSSC scheme compared to SparseNet.** Starting with random filters, we compare here the results of the learning scheme with 324 filters at convergence (20000 steps) using **(Left)** the classical conjugate gradient function method as is used in [17] with **(Right)** the Sparse Spike Coding method. Filters of the same size as the imagelets (16×16) 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 [17] 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.

most efficient and that rather than the shape of the components individually, it is the distribution of the assembly of components that will yield different efficiencies. Such an analysis was performed with a qualitative analysis of the filters' shape, by fitting them with Gabor filters [10]. A recent study compares the distribution of the parameters of the Gabor filters with neurophysiological experiments [7]. They did indeed show that their learning scheme, which is also based on a Matching pursuit algorithm, did better match than SparseNet 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 receptive field formation. 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 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. 3 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

$$\frac{\mathbf{x}}{\|\mathbf{x}\|} = \sum\nolimits_{1\leq j\leq N} \rho_j.\frac{\mathbf{A}_j}{\|\mathbf{A}_j\|} \tag{16}$$

these coefficients then directly correspond to a measure of the correlation coefficient (see Eq. 3). It corresponds to the linear 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 [44]. This is not true in general and the ρ_j correspond here rather to 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 corresponding coefficients, one sees that distributions are approximately gaussians with the initial random filters 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 [7]. Both final distributions seemed to fit well the bivariate model introduced in [45] 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 10 for SparseNet). Plotting the decrease of the sorted

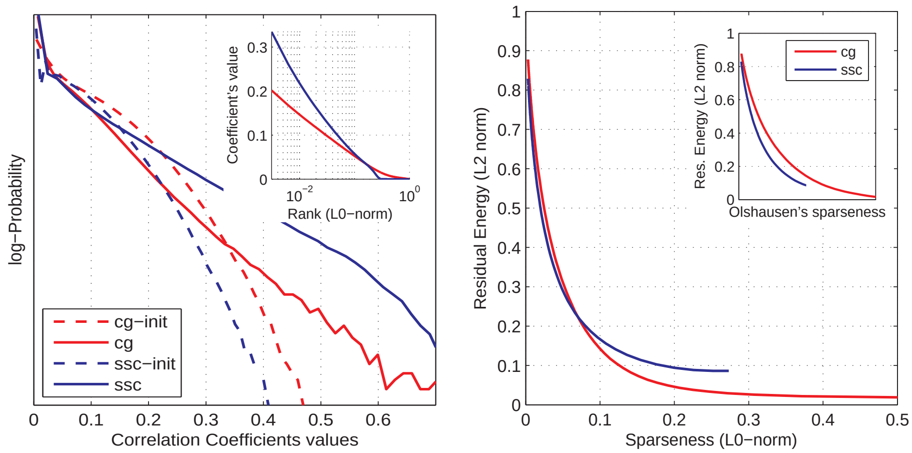

Figure 3: **Coding efficiency of adaptive Sparse Spike Coding.** We evaluated the quality of the algorithm with two different coding strategies by comparing the coding efficiency of the sparse spike coding ('ssc') method with the classical conjugate gradient function ('cgf') method as is used in [17]. 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 (with kurtosis values of respectively 20 and 60): 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 superior 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 simply binary while it is analogous in SparseNet or in [7, 10, 11] and there is no explicitly defined quantization. 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. 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 [21].

coefficients as a function of the number of selected coefficients again showed that first coefficients for SSC were higher and decreased quicker (see Fig. 3-Left Inset) following the link between both curves from Eq. 5. It illustrated also the possible bivariate parameterization of the coefficients.

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 [7] 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 more efficient for the optimized cost but also to the cost defined in the relaxed problem (see Fig. 3). Since the coding used in aSSC is rather sub-optimal (MP) compared to other methods such as [7], 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. 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 an Occam razor confirms that for a similar mean coding efficiency, aSSC is better since it is of lower *structural* complexity[8].

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

[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 aSSC on parallel machines.

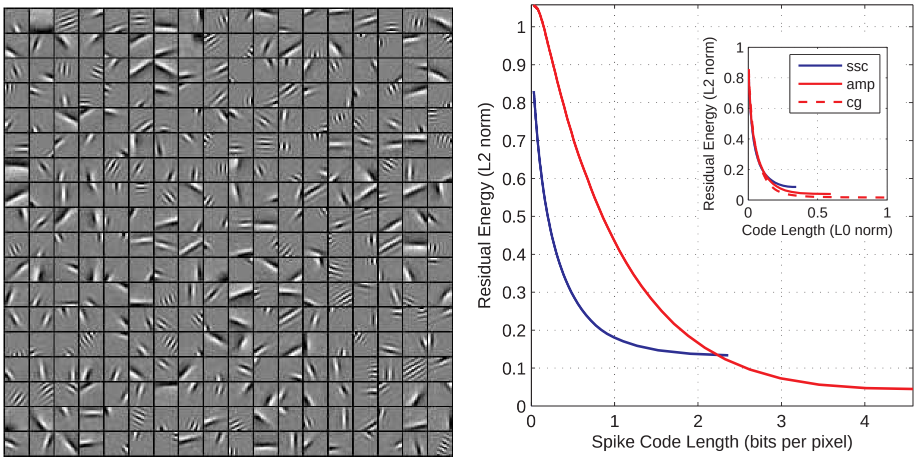

Figure 4: **Homeostasis implements efficient spike quantization.** *(Left)* When relaxing the homeostatic constraint in the SHL algorithm to the one implemented in Adaptive Matching Pursuit (AMP), the algorithm converges to a set of filters which contains some less localized filters and to some high-frequency Gabors which correspond to more 'textural' features. 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 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. It should be noted that homeostasis is important during learning but that from the inherent equalization it is not useful in coding (see Sec. 2.2).

cost function. To resolve this ambiguity, we therefore compared the efficiency for the SHL 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 [15] and which is similar to other strategies such as [7, 11].
In fact, in the AMP algorithm the homeostasis constraint 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 [4]) or the simple Hebbian rule on linear coefficients and that are still optimal to code arbitrary imagelets such as noise [46]. The SHL algorithm 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. 2.2, the homeostasis efficiently constrains the dictionary to better match the a priori pdf of natural scenes in the M-dimensional feature 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 aspects of the signal. On average, the SHL strategy provides a 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 SHL approach (see Fig. 4, Right). This is due to the higher variability of coding coefficients in AMP and therefore of 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 seems more adequate while on a longer term for a spike

[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 SHL the distribution is uniform

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 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. The advantage of our formulation is that we explicitly link here the sparseness constraint with the efficiency of inverting the generative problem. Similar approaches have been taken that could be grouped under the name of Sparse-Hebbian Learning (SHL) [3, 4, 5, 6, 7, 10, 11]. A common solution 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 [35]). A more radical solution based on neurophysiological evidence and not based on a generative model was proposed by [9], 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 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 [47]. However, a major advantage of our formulation is the fact that it tightly coupled the efficiency measure to the coding representation. In particular, the efficiency is based on the spiking nature 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 [48]. 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. Moreover, the response selectivity to rotated oriented lines will be sharper than the linear response [49]. 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 [26, 27]. As a consequence, by taking advantage of the parallel architecture of the cortex, we propose 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. 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 signals 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 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 textured filters could also be good candidates for optimal coding in V1 if the goal was set by a different coding cost. Computationally, the complexity of the algorithms and the time required by both methods was similar on the different simulations on a standard sequential computer. 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). 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. Its efficiency makes it a good candidate for future technologies of information processing. In particular, it compares favorably with compression methods such as JPEG [50]. However it should be stressed that the transfer of this technology to parallel architectures will provide a supra-linear gain of performance. In fact, the SSC algorithm consists of simple operations (integrating and spiking) particularly adapted to an implementation on parallel architecture such as an aVLSI.

To conclude, we proved that using a spiking representation could produce simple yet efficient architecture. Thus, this may explain on a functional level why spikes have been selected during evolution as an efficient signal for long range, rapid communication quanta. However, the main limit 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 [19, 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.

Moreover, an implementation of SSC using Leaky Integrate-and-Fire neurons was previously proposed [49], but this solution proved to be computationally expensive on a sequential computer and that it introduced artifacts from integration approximations. In particular it showed that indeed, the complexity of the ArgMax operator did not depend on the dimension of the vector as in classical solutions, but that on the other hand its precision decreases for a constant level of noise with the number of neurons. At least, to keep mathematical tractability, it is preferable of sticking with abstract neurons which use 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. 2.3. The advantage is that it eases the extension of this algorithm to other type of parallel event-based algorithms. One extension of the algorithm is to not use the implicit 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 [51, 52]. This last example provided a multi-scale analysis algorithm were the set of filters that were learned were a dictionary of mother wavelets of the multi-scale analysis, hence the name of SparseLet Analysis [19, 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 imagelets 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. However, the architecture of the connections between cortical areas suggests that information is distributed, that this distribution is organized according to a hierarchical but also recursive architecture and that an important feature is the generalization of the representation according to noise or common transformations (for an image a translation, a different non-uniform lighting, an occlusion,...). 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 an activation in an area.

Acknowledgments

This work was supported by a grant form the French Research Council (ANR "NatStats") and by EC IP project FP6-015879, "FACETS".

5 Supplementary material

5.1 Annex: Computational implementation

The whole collection of simulation scripts were written with the intention of controlling 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/SparseHebb` Version 1.5 and experiment 20080129T193338 was 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). The original parameters of SparseNet were used for the CGF algorithm.

Table 1: Parameters used in the simulations

script revision	mp-sparsenet-1.5
number of SVN's release	603
dimension of imagelets (CGF & SSC)	256
dimension of dictionary (CGF & SSC)	324
number of learning steps (CGF & SSC)	32001
learning rate (CGF)	1
learning rate (SSC)	0.1
batch size (CGF & SSC)	100
noise variance (CGF)	0.017
noise variance (SSC)	0.008
homeostasis' learning rate (SSC)	0.001
homeostasis' learning rate (CGF)	0.0025
homeostasis smoothing rate (CGF) α	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

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. 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,

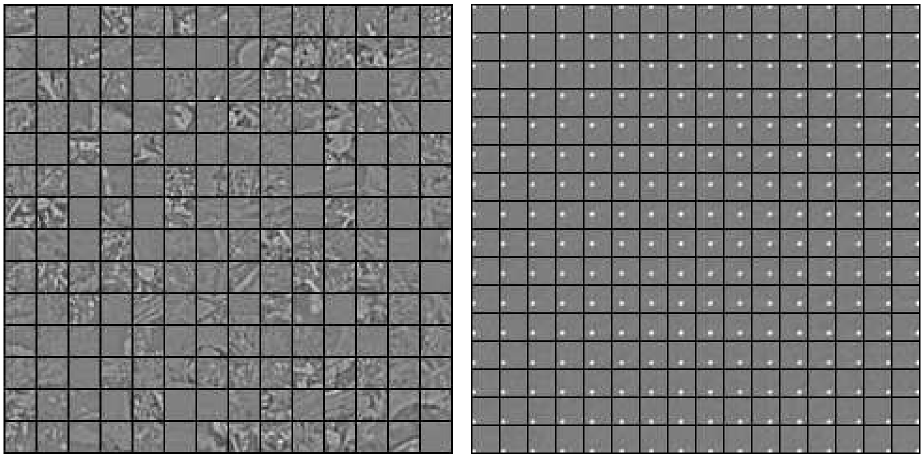

Figure 5: **Control of the statistics of the inputs**. *(Left)* One set of 324 16 × 16 imagelets drawn from the database provided with SparseNet . We use the same presentation as Fig. 2. In order to have comparable patches, the images had to be fairly homogeneous (and therefore textured) since it happened 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. *(Right)* We show here a 16 × 16 matrix of 16 × 16 correlation values representing the covariance matrix of the set of images. 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.

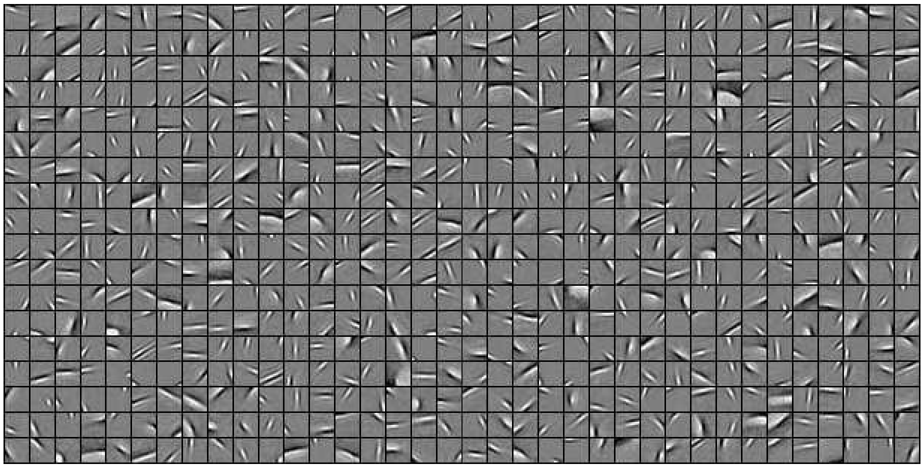

Figure 6: **Solution with non-negative coefficients**. When releasing the symmetry constraint, the learning algorithm converged to a similar set of filters. However, the convergence was quicker and proved to be of higher efficiency when the information for the poalrity (that is, one bit) was rather used 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 is more efficient. See script experiment_symmetric.m to reproduce the figure.

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 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).

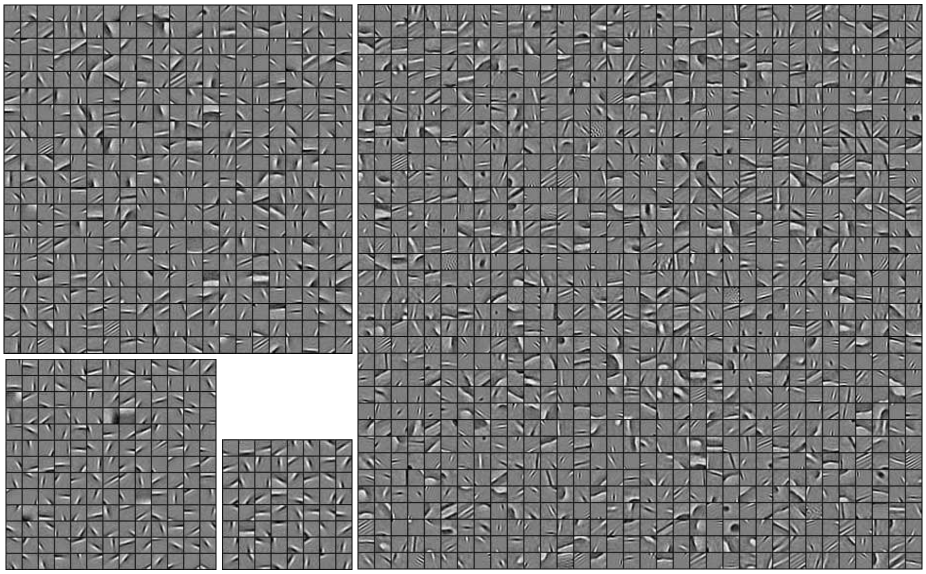

Figure 7: **Solution with an increasing number of coefficients**. When increasing the number of coefficients from 8 × 8 (center,bottom), 13 × 13 (left,bottom), 21 × 21 (left, top) to 34 × 34 one sees that the complexity of the features represented by the filters progressively increases. However, only 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.

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 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 big 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 β and σ) were of particular importance. See scripts experiment_stability_eta.m, experiment_stability_homeo.m and experiment_stability_cgf.m to reproduce the experiments.

References

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

[2] 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`.

[3] 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.

[4] 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`.

[5] 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`.

[6] 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 2004. doi: 10.1023/B:NACO.0000027753.27593.a7. URL `http://incm.cnrs-mrs.fr/LaurentPerrinet/Publications/Perrinet04nc`.

[7] 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*, 2007.

[8] 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`.

[9] 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*, 18(3):249–266, Sep 2007. doi: 10.1080/09548980701661210. URL `http://www.hubmed.org/fulltext.cgi?uids=17926194`.

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

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

[12] 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. [53], mar 2007. doi: 10.1140/epjst/e2007-00061-7. URL `http://lanl.arxiv.org/pdf/nlin/0609038`.

[13] 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`.

[14] 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.

[15] 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.

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

[17] 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.

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

[19] 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. [53], mar 2007. doi: 10.1140/epjst/e2007-00061-7. URL http://incm.cnrs-mrs.fr/LaurentPerrinet/Publications/Perrinet06.

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

[21] 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/.

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

[23] 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.

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

[25] 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.

[26] 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.

[27] 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`.

[28] 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`.

[29] 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.

[30] 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.

[31] 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.

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

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

[34] Bruno A. Olshausen. *Probabilistic Models of the Brain: Perception and Neural Function*, chapter Sparse Codes and Spikes, pages 257–72. In , Rao et al. [54], 2002.

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

[36] 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.

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

[38] 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.

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

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

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

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

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

[44] 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.

[45] 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`.

[46] 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.

[47] 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.

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

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

[50] 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, special issue on Image Perception*, 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`. Received 1 December 2005; Revised 7 September 2006; Accepted 18 September 2006 Recommended by Javier Portilla.

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

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

[53] 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=`

[54] Rajesh P. N. Rao, Bruno A. Olshausen, and Michael S. Lewicki, editors. *Probabilistic Models of the Brain: Perception and Neural Function.* MIT Press, 2002.