# Sparse Spike Coding : applications of Neuroscience to the processing of natural images

Laurent U. Perrinet

Institut de Neurosciences Cognitives de la Méditerranée (INCM)
CNRS / University of Provence
31, ch. Joseph Aiguier, 13402 Marseille Cedex 20, France

## ABSTRACT

If modern computers are sometimes superior to cognition in some specialized tasks such as playing chess or browsing a large database, they can't beat the efficiency of biological vision for such simple tasks as recognizing a relative or following an object in a complex background. We present in this paper our attempt at outlining the dynamical, parallel and event-based representation for vision in the architecture of the central nervous system. We will illustrate this by showing that in a signal matching framework, a L/LN (linear/non-linear) cascade may efficiently transform a sensory signal into a neural spiking signal and we apply this framework to a model retina. However, this code gets redundant when using an over-complete basis as is necessary for modeling the primary visual cortex: we therefore optimize the efficiency cost by increasing the sparseness of the code. This is implemented by propagating and canceling redundant information using lateral interactions. We compare the efficiency of this representation in terms of compression as the reconstruction quality as a function of the coding length. This will correspond to a modification of the Matching Pursuit algorithm where the ArgMax function is optimized for competition, or Competition Optimized Matching Pursuit (COMP). We will particularly focus on bridging neuroscience and image processing and on the advantages of such an interdisciplinary approach.

**Keywords:** Neural population coding, decorrelation, spike-event computation, correlation-based inhibition, Sparse Spike Coding, Competition Optimized Matching Pursuit (COMP)

## 1. INTRODUCTION: EFFICIENT NEURAL REPRESENTATIONS

The architecture of modern day computers illustrate how we understand intelligence. But, if they are good at playing chess or at browsing databases, it is clear that computers are far from rivaling with what appears to be more simple aspects of intelligence such as the ones demonstrated in vision. Think for instance as something as simple as recognizing an object in natural conditions, such as while walking in the street. This necessarily involves a network of processes from segmenting its outline, perceiving its global motion, matching its different patterns invariantly to the shading, contrast, angle of view or to occlusions. Actually, while this seems obvious to us, computers cannot perform this task and it is a common practical "Turing Test" to authenticate humans versus spamming robots by challenging the login upon recognizing for instance warped letters on a noisy background (the so-called CapTchas).

As the seat of this processing, the Central Nervous System (CNS) is therefore by its efficiency clearly different from a classical von Neumann [1] computer defined as a sequential Turing-like machine with a few, very rapid Central Processing Units and a finite, adressable memory. Computational Neuroscience is a branch of neuroscience studying specifically the structure and function of computations in the CNS such as the more complex architectures imagined by von Neumann [2]. Numerous successful theories exist to explain the complex dynamics of modern Artificial Neural Networks and how we may use neuro-physiological constraints to build up efficient systems [3] that are ecologically adapted to the statistics of the input [4]. However, a main challenge involving both neuroscience and computer science is to understand how and for what class of problems the CNS outperforms traditional computers. I am interested in this paper in extracting general principles from the structure of the CNS to derive a

Further author information: E-mail: Laurent.Perrinet@incm.cnrs-mrs.fr, WWW: http://incm.cnrs-mrs.fr/LaurentPerrinet contains supplementary data and metadata about this article, as well as the scripts to reproduce the figures.

better understanding of the neural functions but also to apply these algorithms to signal processing applications. A fundamental difference of the CNS is the fact that 1) information is distributed in parallel on the different neurons, 2) processes are dynamical and interruptible, 3) information is carried by elementary events, called *spikes* which may be transmitted over long distances. This is well illustrated for the large class of pyramidal neurons of the neocortex. In a simplistic way the more a neuron is excited, the quicker and the more often it will emit spikes, with a typical latency of some milliseconds and a maximum firing frequency of the order of 200 ms. Concentrating on local cortical areas (that is in human to the order of some squared centimeters and to a billion neurons), it means that the complexity of some operation will be different on a computer (a few but very rapid CPUs) and a population of neurons (a huge number of slow dynamical event generators). For instance, the complexity of the ArgMax operator (finding the sorted indices from a vector) will increase as $O(Nlog(N))$ with the dimension $N$ of the vector, while if we apply the vector as the activation of a neuronal population, the complexity will not increase with the number $N$ of neurons*. In addition, the result is given by the generated spike list and is interruptible.

In this paper, we will explore how we may apply this class of operators to the processing of natural images by presenting an adaptive Linear/Non-Linear framework and then optimize its efficiency. We will in a first step draw a rationale for using a linear representation by linking it to a probabilistic representaiton under the condition of decorrelation. Then we will derive a linear transform adapted to natural images by constructing a simple pyramidal architecture similar to [5] and extend it to a Laplacian and Log-Gabor pyramids [6]. We will then in a third section propose that this linear information may be optimally coded by a spike list if we apply a point non-linear operation. At least, we will define an improvment over Matching Pursuit [7] by optimizing the efficiency of the ArgMax operator and which finally defines Sparse Spike Coding [8–10].

## 2. LINEAR FILTERING AND WHITENING

A first step in the definition of this algorithm is to explicit the linear operations which are used to transform the input vector into a value representative of the quality of a match. Let's define an image as a set of scalar values $\tilde{x}_i$ on a set of positions $\mathcal{P}$, $i$ being the index of the positions, so that it defines a vector $\tilde{x} \in \mathbb{R}^M$, with $M = card(\mathcal{P})$. As we saw in previous works [8], the quality of a match between the raw data $\tilde{x}$ with a known image may be linked in a probabilistic framework to the correlation coefficient. In fact, the probability of the signal $\tilde{x}$ knowing the "shape" $\tilde{h}$ of the signal to find (see the table Tab. 2 for the chosen notation) is:

$$\begin{aligned} P(\tilde{h}|\tilde{x}) &= \frac{1}{P(\tilde{x})}.P(\tilde{x}|\tilde{h}).P(\tilde{h}) \\ &= \frac{1}{P(\tilde{x})}\frac{1}{(2\pi)^{M/2}}\exp(-\frac{(\tilde{x}-\tilde{h})\mathbf{\Sigma}^{-T}(\tilde{x}-\tilde{h})^T}{2}).P(\tilde{h}) \end{aligned} \tag{1}$$

This is based on the assumption of centered data (that is $E(x) = 0$), a Linear Generative Model and a gaussian noise of covariance matrix $\mathbf{\Sigma} = E(\tilde{x}\tilde{x}^T)$ (See Chapter 2.1.4 of [9]). In the case where the noise is white (that is that the covariance matrix is a diagonal matrix) and assuming an uniform prior for the scalar value of $h$, this may be simply computed with the correlation coefficient defined by:

$$\rho = < \frac{h}{\|h\|}, \frac{x}{\|x\|} > \stackrel{\text{def}}{=} \frac{\sum_{1\leq i\leq M} x_i.h_i}{\sqrt{\sum_{1\leq i\leq M} h_i^2}.\sqrt{\sum_{1\leq i\leq M} x_i^2}} \tag{2}$$

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 ArcCos($\rho_j$) would therefore give the angle of $x$ with the pattern $h$ 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, 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. To achieve

*Note that in a noisy environment, the output will be given with a certain temporal precision and that this precision may decrease with $N$.

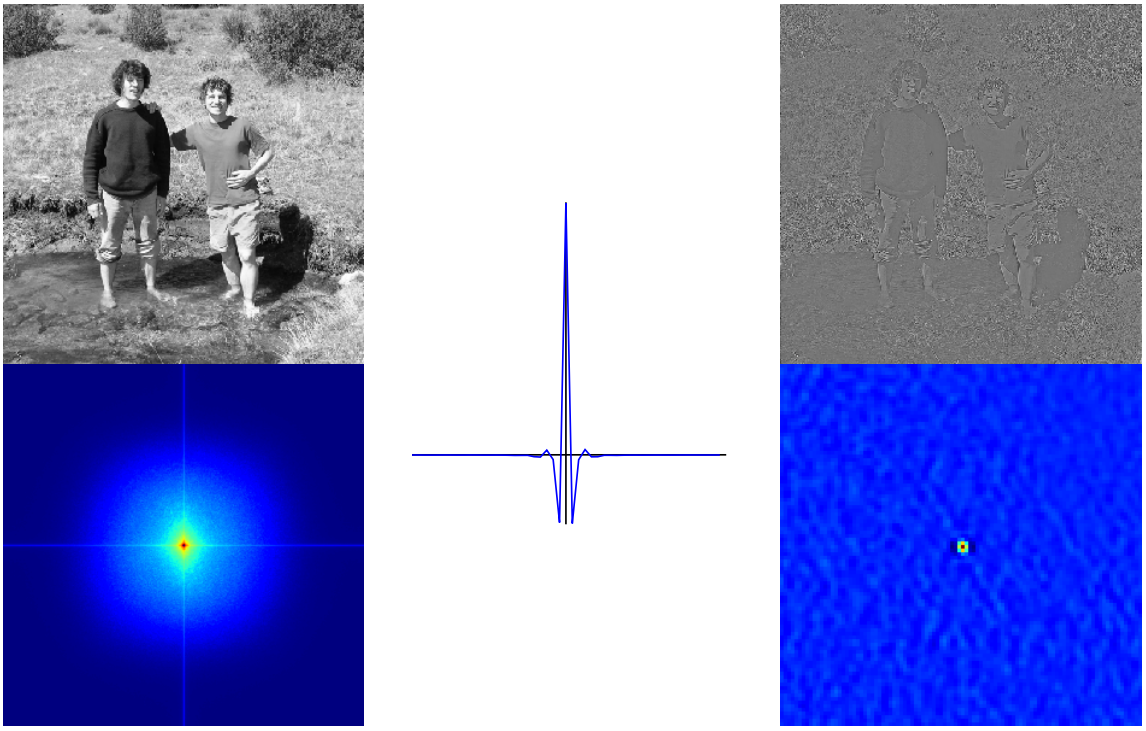

Figure 1. Spatial decorrelation. *(Top-Left)* Sample raw natural image ($M = 256^2$). *(Bottom-Left)* Mean pairwise spatial correlation in a set of 1000 natural images (Red is 1, blue is zero). It shows the typical decrease in $\frac{1}{f^2}$ of the power spectrum but also an anisotropy along the vertical and horizontal axis. *(Middle)* decorrelation filter computed from the methods of[4] (see text). This profile is similar to the interaction profile of bipolar and horizontal cells in the retina. *(Top-Right)* Whitening of the sample image. *(Bottom-Right)* The mean pairwise spatial correlations of 1000 filtered natural images is highly peaked at the origin and inferior to 0.05. As is observed in the LGN, the power spectrum is relatively whitened by our pre-processing[12].

this condition, the raw data $\tilde{x}$ has to be preprocessed with a decorrelation filter to achieve a signal $x$ with no mean point-wise correlation[†]. To define this, we may use for instance the eigenvalue decomposition (EVD) of the covariance matrix:

$$\mathbf{\Sigma} = \mathbf{V}\mathbf{D}\mathbf{V}^T \tag{3}$$

where $\mathbf{V}$ is a rotation (and thus $\mathbf{V}^{-1} = \mathbf{V}^T$) and $\mathbf{D}$ is a diagonal matrix. This decomposition is similar to that achieved by PCA and may be computed for instance by averaging linear correlations such as is done with the linear Hebbian rule[11]. In particular, the columns of matrix $\mathbf{V}$ contain the eigenvectors and $\mathbf{D}$ is a diagonal matrix of the corresponding eigenvalues. If we set $\mathbf{W} = \mathbf{D}^{-\frac{1}{2}}\mathbf{V}^T$ and $x = \mathbf{W}\tilde{x}$, then

$$\begin{aligned} E(xx^T) &= E(\mathbf{W}\tilde{x}(\mathbf{W}\tilde{x})^T) \\ &= \mathbf{D}^{-\frac{1}{2}}\mathbf{V}^T E(\tilde{x}\tilde{x}^T)(\mathbf{D}^{-\frac{1}{2}}\mathbf{V}^T)^T \\ &= \mathbf{D}^{-\frac{1}{2}}\mathbf{V}^T\mathbf{\Sigma}\mathbf{V}\mathbf{D}^{-\frac{1}{2}} \\ &= \mathbf{D}^{-\frac{1}{2}}\mathbf{V}^T\mathbf{V}\mathbf{D}\mathbf{V}^T\mathbf{V}\mathbf{D}^{-\frac{1}{2}} \\ &= \mathbf{1}^{M\times M} \end{aligned}$$

We therefore proved that this linear transforms allows to de-correlate on average the input data. In practice, we used the power spectrum and its relation to the covariance in translation invariant data such as natural images to

[†]Of course, this does not achieve necessarily independence as is often stated.

compute the whitening filter[4]. This corresponds then to a filter with a gain proportional to the spatial frequency but with an anisotropy on the vertical and horizontal axis (see Fig. 1).
Thanks to this processing, and only when these hypothesis have been fullfilled, we may in general use the correlation coefficient (see Eq. 2) as a measure related to the probability of a match of the image with a given pattern. The next step is now to define the best patterns to represent images.

Table 1. Matrix notation and denoising Variables

| Name | Symbol | Description |
|---|---|---|
| Pixel positions | $\mathcal{P}$ | $\vec{p} \in \mathcal{P}, card(\mathcal{P}) = M$ |
| Raw image | $\tilde{x}$ | $\tilde{x} \in \mathbb{R}^M$, $E(\tilde{x}) = 0$ |
| Covariance matrix | $\mathbf{\Sigma}$ | $\mathbf{\Sigma} \in \mathbb{R}^{M \times M}$ |
| Whitening matrix | $\mathbf{W}$ | $\mathbf{W} \in \mathbb{R}^{M \times M}$ |
| Decorrelated image | $x$ | $x = \mathbf{W}\tilde{x} \in \mathbb{R}^M$ |
| Pattern image | $\tilde{h_j}$ | $h_j \in \mathbb{R}^M, j \in \mathcal{D}$ |
| Overcomplete dictionary | $\mathcal{D}$ | $card(\mathcal{D}) = N \gg M$ |
| Decorrelated pattern image | $h_j$ | $h_j = \mathbf{W}\tilde{h_j} \in \mathbb{R}^N$ |
| Transform matrix | $\mathbf{H}$ | $\mathbf{H} \in \mathbb{R}^{N \times M}$ |
| Correlation coefficient | $\rho_j$ | $\rho_j = \frac{<h_j, x>}{\|h_j\|\|x\|} \in [-1, 1]$ |

## 3. MULTISCALE REPRESENTATIONS: THE (GOLDEN) LAPLACIAN PYRAMID

Multi-scale representations are a popular method to allow for a scale invariant representation. This correspond to repeating basic shapes at different scales and it thus allows that one may easily compute the representation of a scaled image by a simple transformation in the representation space instead of recomputing the whole transform. As a consequence, this representation makes it for instance easier to compute the match of a feature at different scales. It is classically implemented in wavelet transforms but we present here a simple implementation using a recurrent scheme, the Laplacian Pyramid[5]. This transform has indeed the advantage of being computed by simple down-scaling and up-scaling operations and is easily inverted for the reconstruction of the image. It transforms an image in a list of down-scaled images, or *image pyramid*. Let's define the list $\{M^k\}$ with $0 \leq k \leq s$ of the sizes of the down-scaled images ($k = 0$ corresponds to the "base" and $M^0 = M$ while $s$ is the level of the smallest image, that is the summit of the pyramid). Typically, such as in wavelets, the size decreases geometrically with an exponent $\gamma$. The most used exponent in image processing is 2, the pyramid is then called *dyadic*. The corresponding down-scale and up-scale transform from level $k$ to $k+1$ may be defined as $\mathcal{D}_k$ and $\mathcal{U}_k$ respectively. We may therefore define the gaussian pyramid as the recursive transform from the "base" of the pyramid to the top as the list of transforms:

$$\mathcal{G} = \{\mathcal{D}^k\} \text{ with } \mathcal{D}^k = \mathcal{D}_0 \circ \cdots \circ \mathcal{D}_k \tag{4}$$

This means that a down-scaled version of the image $\mathcal{D}^k x$ may be obtained by applying all down-scaling transforms sequentially from the base to level $k$. If the elementary operators are linear, the $\mathcal{G}$ transform is linear. The corresponding filters correspond approximately to gaussians with increasing radiuses[5] and the images in the pyramid thus correspond to progressively more blurred versions of the "base" image. This transform is usually very fast and is very likely to be implemented by the extended dendritic arbor of neurons[‡].
The Laplacian Pyramid is defined from the Gaussian Pyramid as the pyramid of images constituted by the residual between the image at one scale and the up-scaled image from the upper level. It is therefore mathematically defined as:

$$\mathcal{L} = \{\mathcal{D}^k - (\mathcal{U}_k \circ \mathcal{D}^{k+1})\} \text{ with } 0 \leq k \leq s \tag{5}$$

[‡]Note however that in vertebrates, the retinal representation the preferred spatial frequency grows with eccentricity.

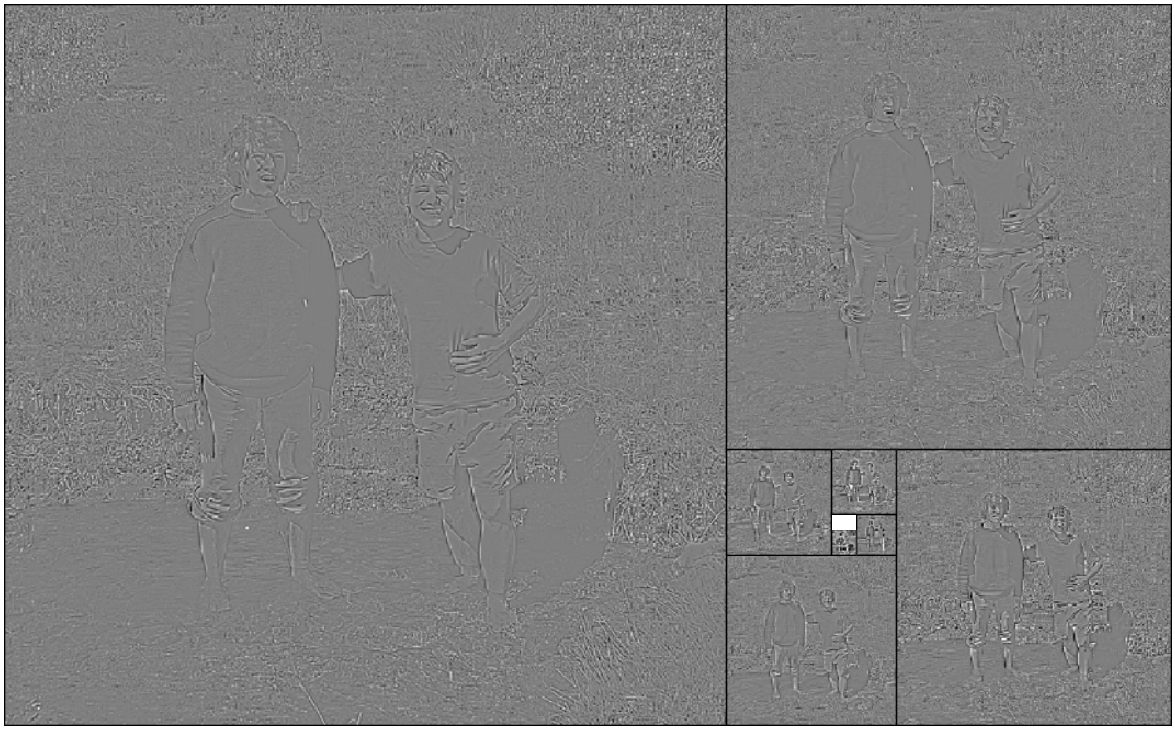

Figure 2. The Golden Laplacian Pyramid. To represent the edges of the image at different levels, we may use a simple recursive approach constructing progressively a set of images of decreasing sizes , from a base to the summit of a "pyramid". Using simple down-scaling and up-scaling operators we may approximate well a Laplacian operator. This is represented here by stacking images on a "Golden Rectangle", that is where the aspect ratio is the golden section $\phi \stackrel{\text{def}}{=} \frac{1+\sqrt{5}}{2}$. We present here the base image on the left and the successive levels of the pyramid in a clockwise fashion (for clarity, we stopped at level 8). Note that here we also use $\phi^2$ (that is $\phi + 1$) as the down-scaling factor so that the resolution of the pyramid images correspond across scales. Note at last that coefficient are very kurtotic: most are near zero, the distribution of coefficients has "long tails".

by defining for clarity that $\mathcal{D}^0 = 1$ and $\mathcal{D}^{s+1} = 0$. This transform is still linear that is that $\forall x, \forall y, \forall \lambda$, $\mathcal{L}(x+y) = \mathcal{L}x + \mathcal{L}y$ and $\mathcal{L}(\lambda x) = \lambda \mathcal{L}x$. Since every level corresponds to the residual, it is easy to invert. In fact, if we write as $\mathcal{L}_k x$ the image at level $k$ and $\mathcal{U}^k = \mathcal{U}_0 \circ \cdots \circ \mathcal{U}_k$, then $\forall x$,

$$\begin{aligned}
\sum_{0 \le k \le s} \mathcal{U}^k \mathcal{L}_k x &= \sum_{0 \le k \le s} \mathcal{U}^k (\mathcal{D}^k - (\mathcal{U}_k \circ \mathcal{D}^{k+1}))x \\
&= \sum_{0 \le k \le s} \mathcal{U}^k \mathcal{D}^k x - \sum_{0 \le k \le s} \mathcal{U}^k \mathcal{U}_k \circ \mathcal{D}^{k+1} x \\
&= \sum_{0 \le k \le s} \mathcal{U}^k \mathcal{D}^k x - \sum_{1 \le k \le s+1} \mathcal{U}^k \circ \mathcal{D}^k x = x
\end{aligned} \tag{6}$$

Therefore the inverse of the Laplacian Pyramid transform is defined as:

$$\mathcal{L}^{-1} = \sum_{0 \le k \le s} \mathcal{U}^k \mathcal{L}_k \tag{7}$$

The filters corresponding to the different levels of the pyramid (and which are the inverse image of a Dirac pyramid by $\mathcal{L}^{-1}$) are similar to difference of gaussians (because they are the difference of two successive levels of the Gaussian Pyramid). The exponent $\gamma$ will therefore play the important role of the ratio of the the radiuses of the Gaussians. We choose here the exponent to be equal to the golden number $\gamma = \phi \stackrel{\text{def}}{=} \frac{1+\sqrt{5}}{2} \approx 1.618033$ for

two reasons. First, it corresponds to a value which approximates well a Laplacian-of-Gaussians with a Difference of Gaussians as is implemented here. Second, it allows to construct a natural representation of the whole pyramid in a full Golden Rectangle (see Fig. 2) where the resolution of each image will be constant.
Note the following properties of the pyramid:

- the over-completeness is equal to $\sum_{0\leq k\leq s}\frac{1}{\gamma^{2k}}\approx\frac{1}{1-\gamma^{-2}}$ so that it is equal to $\frac{1}{1-\phi^{-2}}=\frac{\phi}{\phi-\phi^{-1}}=\phi$ which is indeed the area of the Golden Rectangle compared to the area of the image. It is slightly higher than for a dyadic pyramid (indeed $\frac{1}{1-2^{-2}}=\frac{4}{3}\approx 1.333<\phi$).
- since this linear transform is over-complete, there may exist non zero pyramids which inverse image is null (that is $\exists L\neq 0$ such that $\mathcal{L}^{-1}L=0$) but this pyramids are not accessible from any non-null image.
- one may also implement a simple "Golden Pyramid" using the Fourier transform, and one may observe that in both cases, the filters corresponds to localized filters in the frequency space. The whitening (see Sec. 2) has an approximately scalar effect that corresponds to an equalization of the variances of the coefficients to natural images at the different spatial frequencies.
- Finally, once the obtained filters are normalized, the coefficients will correspond to the correlation coefficients of the image with edge detectors at different scales as defined in Eq. 2. The coefficients will therefore as in wavelet analysis correspond to the local Lipschitz coefficients of the image [13]. When ordered by decreasing absolute values they will correspond to features of decreasing singularities, from a pure singularity, to a smooth transition (as a ramp of luminosity).

Table 2. Notations used for the Laplacian Pyramid

| Name | Symbol | Description |
|---|---|---|
| sizes of the down-scaled images | $\{M^k\}$ | $0\leq k\leq s$ |
| Down-scale operator | $\mathcal{D}_k$ | from level $k$ to $k+1$ |
| Up-scale operator | $\mathcal{U}_k$ | from level $k$ to $k+1$ |
| Full Down-scale operator | $\mathcal{D}^k$ | $\mathcal{D}^0=1$ and $\mathcal{D}^{s+1}=0$ |
| Full Up-scale operator | $\mathcal{U}^k$ | |
| Gaussian Pyramid | $\mathcal{G}$ | |
| Laplacian Pyramid | $\mathcal{L}$ | $\mathcal{L}=\{\mathcal{L}_k\}$ with $0\leq k\leq s$ |

## 4. SPIKE CODING

Now that we defined a linear transform which is suitable for natural images by associating the whitening filters and the Laplacian Pyramid, we wish to transmit this information efficiently using neurons. As we saw in the previous section, the higher coefficients correspond to more singular features and therefore to more informative content. By using Integrate-and-Fire neurons [14], it is therefore natural that we may associate to every coefficient of the pyramid applied to the image a single neuron. For the linear Leaky-IF, if we associate a driving current to each value $\rho_j$ (with $0\leq j\leq N$, as noted in Tab. 2) it will will elicit spikes with latencies [13]:

$$\lambda_j=-\tau\log(1-\theta.g_j(\rho_j)) \tag{8}$$

where $\tau$ is the characteristic time constant, $\theta$ is the neuron's threshold and $g_j$ is a monotonously increasing function of $\rho_j$ corresponding to the transformation of the linear value into the driving current. By this architecture, since the relation in Eq. 8 is monotonously increasing, one implements a simple ArgMax operator where the output is the index of the neurons corresponding to the ordered list of output spikes.
However, one may observe that for some linear transforms, the distribution of correlation coefficients may be not similar for all $j$. This is contradictory with the fact that spikes are similar across the CNS since it would mean that the probability of the coefficient underlying the emission of a spike is not uniform. 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 functions $g_j$ so that:

1. for all $j$, the distributions of $g_j(\rho_j)$ are similar,

2. allow that this overall distribution has a shape adapted to the spiking mechanism (for instance by using Eq. 8).

The second point —finding a global non-linearity $g$— will be out of scope of this paper, and we will for the sake of generality only ensure that we find functions $f_j$ (with $g_j = g \circ f_j$) such that the variables $z_j = f_j(\rho_j)$ are uniformly distributed.
This condition is easily performed by operating a point non-linearity on the different variables $\rho_j$ based on the statistics of natural images[4]. 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[15] or dually the minimization of intrinsic noise[16]. It may be easily derived from the probability $P$ of variable $\rho_j$ (bounded in absolute value by 1) by choosing the non-linearity as the cumulative function

$$f_j(\rho_j) = \int_{-1}^{\rho_j} dP(\rho) \tag{9}$$

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[17]. 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 schemes within a map[18]. 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. 9, 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 $P(\rho_{j(1)})$ on the emitting side. We may therefore decode the corresponding value with the best estimate $\hat{\rho}_{j(1)} = f^{-1}_{j(1)}(\frac{1}{N})$ where $N$ is the total number of neurons. This is also true for the following spikes and if we write as $z_{j(k)} = \frac{k}{N}$ the relative rank of the spike (that is neuron $j(k)$ fired at rank $k$), we can reconstruct the corresponding value as

$$\hat{\rho}_{j(k)} = f^{-1}_{j(k)}(1 - z_{j(k)}) \tag{10}$$

This corresponds to a generalized rank coding scheme[19;20]. 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 (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[§]. On a practical note, we may 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.
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[21]. More generally, one could use different mappings for the transformation

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

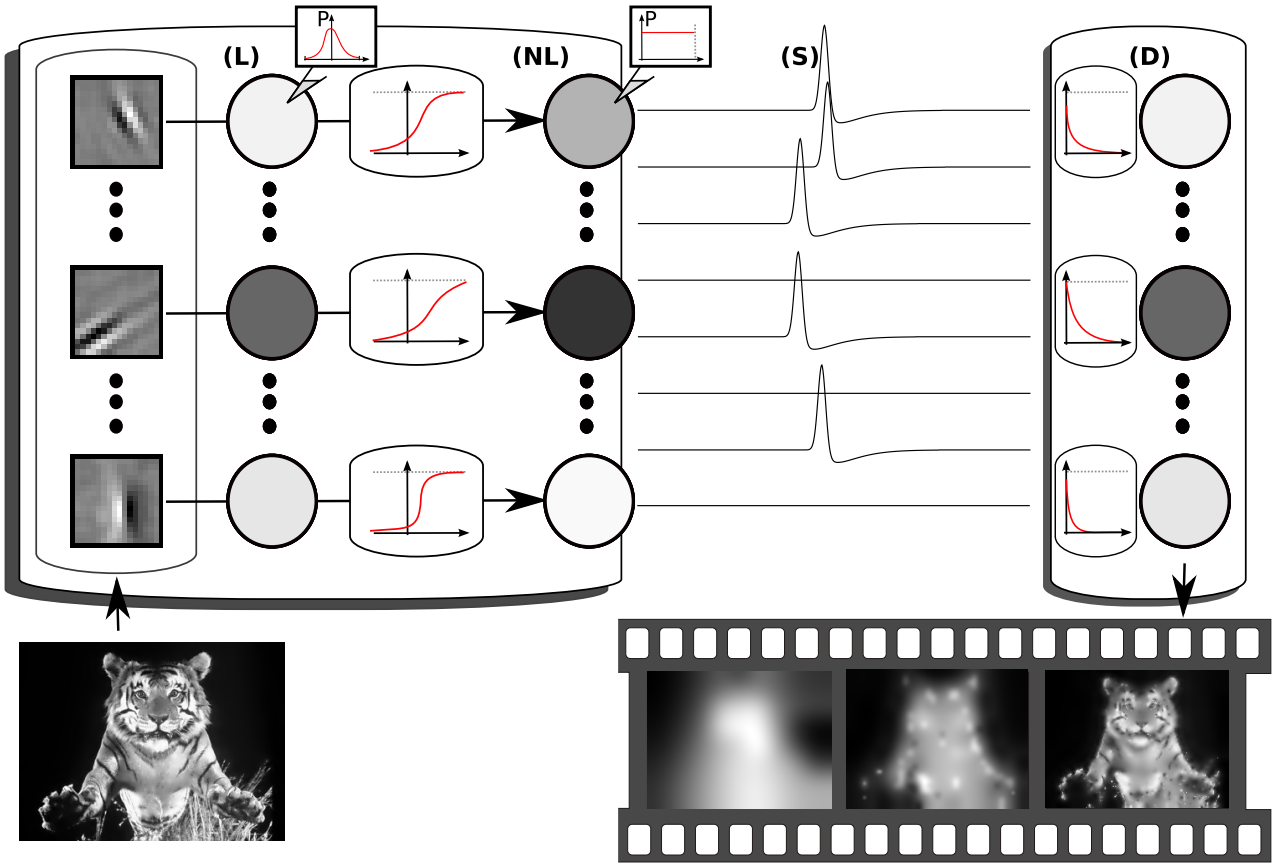


Figure 3. Spike Coding of natural images. We did build here a simple framework of pyramidal neurons illustrating the efficiency of neural architectures compared to classical computer architectures. We show here how a bundle of L-NL neurons[26;27] tuned by a simple homeostatic mechanism allow to transfer a transient information, such as an image, using spikes. (L) The signal to be coded, for instance the match $\rho_j$ 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 $\rho_j$ 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. 10). 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 efficient compression algorithms. Note that this scheme is similar to the N-NL scheme but that instead of generating a Poisson point process, we use the the exact timing. This is allowed by the point non-linearity which permits to code the value by the timing and not the firing frequency.

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 [22] and is mathematically more tractable. In particular, one may show that the coding error is proportional to the variability of the sorted coefficients [13], 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 simple terms, there is no reason to have a reconstruction of the image in the CNS. 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).

## 5. SPARSE SPIKE CODING

However, as we described before [8–10], if we use over-complete dictionaries of filters, the resulting spiking code gets redundant. In fact, unless the dictionary is orthogonal, when choosing one component over an other, 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 optimizing the choice leads then to a combinatorial explosion [23]. 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 [6;24;25]. 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 the *Sparse Spike Coding* (SSC) algorithm [10], one could use a greedy algorithm on the $L_0$-norm cost and that these led to use of Matching Pursuit algorithm [7]. 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 $x$, we are searching for the *single* source $s^*_{j^*}.h_{j^*}$ that corresponds to the maximum *a posteriori* (MAP) realization for $x$ (see Eq. 2) transformed by a point non-linearity $f_j$. This Matching step is defined by:

$$j^* = \text{ArgMax}_j[f_j(\rho_j)] \tag{11}$$

where $f_j(.)$ is some gain function that we will describe below and which may be set initially to strictly increasing functions and $\rho_j$ is initialized by Eq. 2. In a second step (Pursuit), the information is fed-back to correlated sources through :

$$x \leftarrow x - s^*_{j^*}.h_{j^*} \tag{12}$$

where $s^*_{j^*}$ is the scalar projection $< x, h_{j^*} >$. Equivalently, from the linearity of the scalar product, we may propagate laterally:

$$< x, h_j > \leftarrow < x, h_j > - < x, h_{j^*} >< h_{j^*}, h_j > \tag{13}$$

that is from Eq. 2:

$$\rho_j \leftarrow \rho_j - \rho_{j^*} < h_{j^*}, h_j > \tag{14}$$

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. 11 until some stopping criteria is reached. The signal may be reconstructed from the spike list as $x = \sum \hat{\rho}_{j(k)} h_{j(k)}$, where $\hat{\rho}_{j(k)}$ is the value reconstructed using Eq. 10. We then define Competition Optimized Matching Pursuit (COMP) as WMP where the point non-linearities are defined by Eq. 9 and Sparse Spike Coding (SSC) is then defined as the spike coding/decoding algorithm which uses COMP as the coder. As described

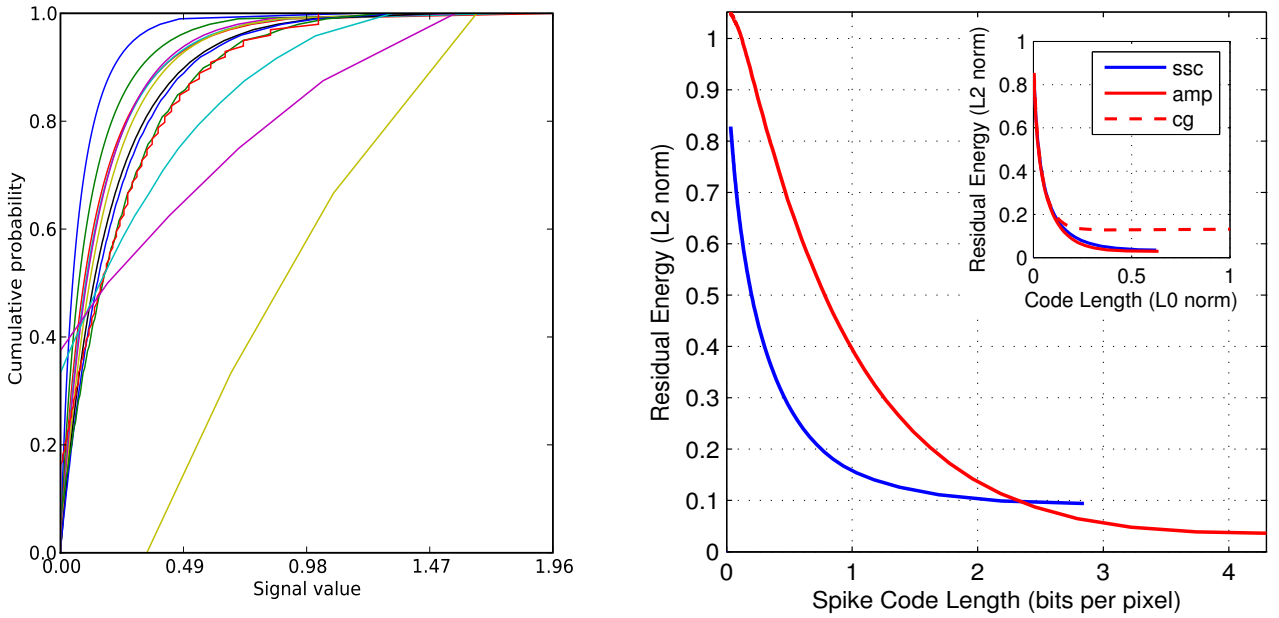


Figure 4. Efficiency of Competition Optimized Matching Pursuit (COMP). Spike Coding and Sparse Spike Coding (using COMP) produce flows of spikes representing the image. By representing the the distance of the original image with a reconstruction, one may quantify the dynamical efficiency of this solution as a function of the number of spikes. *(Left)* When applying the algorithm on a set of natural images, the coefficients exhibited differences in their probability density functions. We show this by plotting the cumulative density functions of the coefficients for different levels in the pyramid. Using these cumulative pdf, one could transform the pyramids of coefficients in pyramids for which all coefficients where *a priori* equiprobable. This optimizes the ArgMax operator which is at the heart of the Sparse Spike Coding scheme. *(Right)* The resulting COMP solution gives a similar result than MP in terms of residual energy as a function of pure $L_0$ sparseness (see inset). In fact, in MP, by taking the maximum absolute, and since the decrease of energy is proportional to the square of the coefficient (see Chapter 3.1.2 of [9]) one ensures that the decrease of MSE *per coefficient* is optimal for MP. These are both better for that purpose 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 COMP model is more efficient than MP because of the quantization errors inherent to the higher variability of coded coefficients. Thus, including homeostasis improved the efficiency of adaptive Sparse Spike Coding by ensuring that the decrease of MSE *per bit of code* is optimal. It should be noted that the homeostasis mechanism is important during "learning" but that it is not useful for "pure" coding (see Sec. 5).

in[8], while the Matching step is efficiently performed by the LIF neurons driven by the NL input, 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. 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. 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. 9. 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 the Sparse Spike Coding problem (see Fig. 3).

Extensions of this type of event-based algorithms are multiple. First, It extends naturally to the temporal domain. In fact, we restricted us ourselves here to static flashed images, but is easily extendable to causal filters (see Ch. 3.4.1 in [9]). It however raises the unsolved problem of a dynamical compromise between precision and rapidity of the code which is still unanswered. It may also be extended in a adaptive code, showing the emergence of V1-like receptive fields[23]. At last, using in these sparse representations of long-range interactions such as those present in the primary visual cortex should prove to be very helpful to resolve generic image processing problems such as denoising.

**Reproducible science / Acknowledgments**

All algorithms used in this paper were implemented using Python, Numpy, SciPy (FFT and image libraries) and Matplotlib (for the visualization). Scripts are available upon request.

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

## References

[1] John von Neumann. *Theory of Self-Reproducing Automata*. University of Illinois Press, 1966.

[2] John v. von Neumann. *The Computer and the Brain : Second Edition (Mrs. Hepsa Ely Silliman Memorial Lectures)*. Yale University Press, July 2000. ISBN 0300084730. URL `http://www.amazon.ca/exec/obidos/redirect?tag=citeulike09-20\&path=ASIN/0300084730`.

[3] Stephen Grossberg. How does the cerebral cortex work? development, learning, attention, and 3-d vision by laminar circuits of visual cortex. *Behavioral and Cognitive Neuroscience Reviews*, 2(1):47–76, March 2003.

[4] 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`.

[5] Peter J. Burt and Edward H. Adelson. The Laplacian Pyramid as a compact image code. *IEEE Transactions on Communications*, COM-31,4:532–40, 1983.

[6] Sylvain Fischer, Rafael Redondo, Laurent U. Perrinet, and Gabriel Cristóbal. Sparse Gabor wavelets by local operations. In Gustavo Linan-Cembrano Ricardo A. Carmona, editor, *Proceedings SPIE*, volume 5839 of *Bioengineered and Bioinspired Systems II*, pages 75–86, Jun 2005. doi: 10.1117/12.608403.

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

[8] 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/`.

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

[10] 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.

[11] Erkki Oja. A Simplified Neuron Model as a Principal Component Analyzer. *Journal of Mathematical biology*, 15:267–273, 1982.

[12] Y Dan, Joseph J. Atick, and RC Reid. Efficient coding of natural scenes in the lateral geniculate nucleus: experimental test of a computational theory. *Journal of Neuroscience*, 16(10):3351–62, May 1996.

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

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

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

[16] 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.

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

[18] 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`.

[19] 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`.

[20] 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.

[21] 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.

[22] 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.

[23] Laurent U. Perrinet. Optimal signal representation in neural spiking codes: A model for the formation of simple cell receptive fields. 2008. URL `http://fr.arxiv.org/abs/0706.3177`.

[24] Sylvain Fischer, Gabriel Cristóbal, and Rafael Redondo. Sparse Overcomplete Gabor Wavelet Representation Based on Local Competitions. *IEEE Transactions in Image Processing*, 15(2):265, 2006.

[25] Sylvain Fischer, Filip Sroubek, Laurent U. Perrinet, Rafael Redondo, and Gabriel Cristóbal. Self-invertible 2D log-Gabor wavelets. *Int. Journal of Computional Vision*, 2007.

[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`.