A NOVEL BIO-INSPIRED STATIC IMAGE COMPRESSION SCHEME FOR NOISY DATA TRANSMISSION OVER LOW-BANDWIDTH CHANNELS

Khaled Masmoudi, Marc Antonini, Pierre Kornprobst, Laurent Perrinet

A NOVEL BIO-INSPIRED STATIC IMAGE COMPRESSION SCHEME FOR NOISY DATA TRANSMISSION OVER LOW-BANDWIDTH CHANNELS

Khaled Masmoudi, Marc Antonini — I3S - UNSA - CNRS, Sophia Antipolis France, kmasmoud, am@i3s.unice.fr

Pierre Kornprobst — NeuroMathComp - INRIA, Sophia Antipolis France, pierre.kornprobst@inria.fr

Laurent Perrinet — INCM - CNRS, Marseille France, laurent.perrinet@incm.cnrs-mrs.fr

ABSTRACT

We present a novel bio-inspired static image compression scheme. Our model is a combination of a simplified spiking retina model and well known data compression techniques. The fundamental hypothesis behind this work is that the mammalian retina generates an efficient neural code associated to the visual flux. The main novelty of this work is to show how this neural code can be exploited in the context of still image compression. Our model has three main stages. The first stage is the bio-inspired retina model proposed by Thorpe et al [1, 2], which transforms an image into a wave of spikes. This transform is based on the so-called rank order coding. In the second stage, we show how this wave of spikes can be expressed using a 4-ary dictionary alphabet, through a stack run coder. The third stage consists of applying a first order arithmetic coder to the stack run coded signal. We compare our results to *JPEG* standards and we show that our model has comparable performance for lower computational cost under strong bit rate restrictions when data is highly contaminated with noise. In addition, our model offers scalability for monitoring data transmission flow. The subject matter presented highlights a variety of important issues in the conception of novel bio-inspired compression schemes and additionally presents many potential avenues for future research efforts.

Index Terms— Static image compression, spiking retina model, rank order coding, stack run.

1. INTRODUCTION

During the past two decades, research in still image compression generated several coding algorithms, especially the $JPEG$ standards. Since then, subsequent efforts followed the same schema for conceiving lossy image coders [3]. These compression algorithms were, for the most part, designed in a signal processing way, and do not account for actual biological visual systems behavior.

Yet, computational neuroscience made substantial progress during the same period of time in better understanding the internal representation of the sensory world. For example, concerning the visual system, one can find many results and heuristics on how information is encoded, transmitted and interpreted. Based on those results, it is our conviction that the visual system developed efficient skills and coding strategies that could be used as a source of inspiration to imagine novel compression techniques.

The human visual system conveys information as a set of electrical impulses called spikes. Spikes [4], which are the way that our nervous system chose to communicate information broadly, appear very early in the chain of treatment of the human visual system. At the retina level, after a chain of internal treatments, ganglion cells convert an analogous signal into a series of spikes called spike trains. So, one challenge is to be able to understand the relations between spike trains and stimuli.

In the literature, several measurable features of spike trains were proposed. In this paper, we focus on the latency time of the first spike given the stimulus onset. This choice was motivated by Thorpe et al neurophysiologic results [1] on ultra-rapid stimulus categorization. Authors showed that still image classification can be realized by the visual cortex within very short latencies of about 150 ms or even faster. As an explanation, it was stated that: *the order in which neurons emit their first spike is related to their level of excitation.* This temporal ordering of spikes was further studied in [1, 2] and designated by *rank order coding* (ROC). Interestingly, the relevance of ROC was recently confirmed by electrophysiologic experiments [5], and we would like to investigate if it could be useful for image compression.

This paper is organized as follows: In Section 2, we detail the stages of our static image compression scheme, then in Section 3 we summarize the results obtained, and in Section 4, we highlight many potential avenues for future research efforts in the conception of novel bio-inspired compression schemes.

2. PROPOSED CODING-DECODING SYSTEM

2.1. System overview

Our compression scheme encompass three stages: first, a spiking retina model based on rank order coding (ROC), second, a zero-run length coder, namely a stack run coder, and finally an arithmetic coder. This is summarized in the block diagram in Figure 1. The following sections detail each one of these stages.

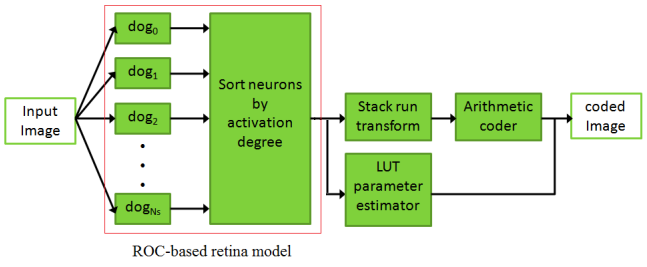


Fig. 1. Block Diagram of the compression scheme. First, a ROC based retina model. Second, the ROC code is zero-run length encoded by the stack run coder. Finally a first order arithmetic coder is applied to get the compressed image file. The decoding process goes exactly the opposite way.

2.2. From image to rank order code

Experiments show that we are very talented in categorizing images even with very short presentation durations. As an explanation for this extraordinary performance in ultra-fast categorization, Thorpe et al [1, 2] proposed that the order in which spikes are emitted encodes for the stimulus. This yielded the ROC code which will be the base of our approach. ROC relies on the following simplifying assumptions:

i) From stimulus onset, only the first spike emitted is considered in the response.

ii) The time to fire of each ganglion cell is proportional to its degree of excitation.

iii) Only the order of firing of the neurons encodes for the stimulus.

In order to obtain spikes from an image, the authors in [2] proposed an *N_s-layered dyadic grid G of filters as an architecture for the retina model*. In a layer G_k of G, ganglion cells are regularly spaced by 2^s pixels over the horizontal and vertical axes. The set of ganglion cell locations defines a subset U of $\mathbb{N}^3$. For an N^2-sized image, U is defined by :

$$U = \{(s,i,j) \text{ such that } (s,i,j) \in \mathbb{N}^3, s \in [0, N_s], \\ i < N, j < N, i \equiv [2^s] \text{ and } j \equiv [2^s]\}, \quad (1)$$

where s denotes the scale of the cell and (i,j) its position. Then the response of a ganglion cell can be approximated by a linear filter [6]. In particular, [6] proposed the *dog* filter which is a weighted difference of Gaussians and is defined as follows:

$$dog(x,y) = w_c g_\sigma(x,y) - w_s g_{\alpha\sigma}(x,y), \quad (2)$$

where σ is the so-called central standard deviation of the filter, α is an a priori fixed real number, g_σ (resp. $g_{\alpha\sigma}$) is the Gaussian kernel of standard deviation σ (resp. $\alpha\sigma$), and w_c (resp. w_s) is the weight of g_σ (resp. $g_{\alpha\sigma}$). With biologically realistic parameters a *dog* filter applied to the image is a contour detector.

So, in order to measure the degree of activation of a given neuron, we compute the convolution of the original image f by a *dog* filter defined by its scale s and its location (i,j), so that:

$$c_{sij} = \sum_{x,y=-\infty}^{\infty} dog_s(i-x, j-y)\, f(x,y). \quad (3)$$

Neuron responses are then sorted in the decreasing order of their amplitude, i.e., $|c_{sij}|$. For an N^2-sized image we obtain the series of sorted triplets $(s_k, i_k, j_k)_{0 \leqslant k \leqslant \frac{4}{3}N^2 - 1}$ defined recursively by:

$$\begin{cases} U_0 & = U \\ (s_0, i_0, j_0) & = \underset{(s,i,j) \in U_0}{\operatorname{argmax}} (|c_{sij}|) \\ U_k & = U \setminus \{(s_0, i_0, j_0), ..., (s_{k-1}, i_{k-1}, j_{k-1}))\} \\ (s_k, i_k, j_k) & = \underset{(s,i,j) \in U_k}{\operatorname{argmax}} (|c_{sij}|) \end{cases}$$

So, if we consider the set of the first N_c spikes, the reconstruction estimate $\tilde{f}_{N_c}$ of the original input f can be obtained as follows:

$$\tilde{f}_{N_c}(x,y) = \sum_{k=0}^{N_c - 1} c_{s_k\, i_k\, j_k}\, dog_{s_k}(i_k - x, j_k - y). \quad (4)$$

Formula (4) gives a progressive reconstruction depending on N_c. This feature will make the coder scalable. Note that in (4), we implicitly made the assumption that the filters considered form an orthonormal basis, which is an approximation as mentioned in [7]. Some enhanced versions of this algorithm are presented in the literature. For example, in [8], the authors use a matching pursuit procedure in order to eliminate redundancies between coefficients; and in [7], the authors proposed to invert the *dog* basis using a Moore-Penrose pseudo-inversion procedure for the same purpose.

A step further, it appears that we do not need to remember explicitly the amplitude of the responses $|c_{sij}|$ in (4). In [1, 2], the authors showed that there exists a mapping between the rank and the amplitude $|c_{sij}|$ which has approximately the same shape across natural images. Thus the authors supposed that the exact values of $|c_{s_k i_k j_k}|$ are known a priori at the level of the decoder. This allows a great gain in the amount of information necessary to encode the input image. The characteristic series of $|c_{s_k i_k j_k}|$ is constructed off-line and stored in a look up table (LUT). The LUT used in [2] is obtained as an average over a set of natural images. In this paper, instead of a predefined mapping as in [2], we use a parametric function LUT_γ of the rank k of (s_k, i_k, j_k) defined by:

$$LUT_\gamma(k) = C\, k^{-\gamma}, \quad (5)$$

where C is an arbitrary constant positive value and γ is the parameter of the function. The motivation is to find the best fit for each image. To do so, we implemented a gradient descent over γ that minimizes the criterion of Mean Squarred Error (MSE) between the real coefficients $|c_{s_k i_k j_k}|$ and the estimated ones $LUT_\gamma(k)$.

At the end of this stage, the ROC code generated consists only of the parameter γ of LUT_γ and a series of sorted quadruplets $(e_k)_{0 \leqslant k \leqslant N_c}$ defined by:

$$\forall\, 0 \leqslant k < N_c,\ e_k = (s_k, i_k, j_k, sign(c_{s_k i_k j_k})). \quad (6)$$

So the reconstruction formula (4) becomes:

$$\tilde{f}_{N_c}(x,y) = \sum_{k=0}^{N_c - 1} sign(c_{s_k i_k j_k})\, LUT_\gamma(k)\, dog_{s_k}(i_k - x, j_k - y).$$

An example of progressive reconstruction of Lena is shown in Figure 2. Remark that, as an enhancement to the current coder, we added a Gaussian scaling function [9] to catch low frequencies omitted by the original model. The latter modification improves the image reconstruction quality by $5\,dB$ in PSNR for 5% of fired spikes used for the reconstruction (see Figure 2). Note that the reconstruction evolution maps actual retina behavior, as low frequencies are first transmitted, then details are progressively added (see, e.g., [10]).

2.3. Coding spikes using stack run

The ROC coder presented in Section 2.2 generates a series of at most N_s spikes denoted as $(e_k)_{0 \leqslant k \leqslant N_s - 1}$, where N_s is the size of the dyadic grid. As demonstrated in Figure 2, few spikes can reasonably represent the image to code. This property is loosely referred to as sparseness in the literature. The sparseness of neural codes has been shown to help conceive meaningful representations of data. Keeping sparseness as a design principle, we define, in this section, a sparse representation of the series (e_k) in the sense that, we look for a series mainly populated with zero's. We then introduce a zero-run length coder well suited for sparse data coding, namely the stack-run coder.

We consider the definition of e_k in (6). For each triplet elements (s_k, i_k, j_k) of e_k, let us define the scalar index $r_k \in [0, N_s - 1]$ defined by:

$$r_k = s_k N^2 + \frac{i_k}{2^s} N + \frac{j_k}{2^s}, \quad (7)$$

such that, $c_{s_k i_k j_k}$ can be denoted c_{r_k}. So, the ROC response restricted to the first N_c spikes can be written as the following sorted list of N_c couples (spiking cell index, sign of the response):

Fig. 2. Progressive image reconstruction of Lena using Thorpe ROC coder with a scaling function. Upper left: Lena. Upper right and lower: The coded/decoded image reconstruction with an increasing percentage of spikes decoded.

$$M_{retina}^{N_c} = ((r_0, sign(c_{r_0})), ..., (r_k, sign(c_{r_k})), ...).$$

Coding this list may be quite expensive in terms of bit-rate. This is because from one index to the next, one does not consider their spatial arrangements. The dimension of M_{retina} is N_c but the values to encode may be large. For this reason, we propose instead a representation taking into account the relative positions of the spiking cells. So we define the vector:

$$M_{sparse}^{N_c} = (m_0, ..., m_l, ..., m_{N_s-1}),$$

so that:

$$m_l = \begin{cases} k\, sign(c_{r_k}) & \text{if } \exists k < N_c \ / \ l = r_k \\ 0 & \text{otherwise.} \end{cases} \quad (8)$$

The dimension of $M_{sparse}^{N_c}$ is N_s (the size of the dyadic grid), but the range of values to encode now depends on N_c. For a sufficiently small value of N_c, $M_{sparse}^{N_c}$ is a sparse data set: this is the feature we tried to enhance in our message code $M_{sparse}^{N_c}$. Zero-run length coders are well suited for the compression of such data sets. In our coder we use an enhanced run-length coding algorithm, the stack run code [11]. Stack run coding uses a 4-ary dictionary $\{0, 1, +, -\}$. $M_{sparse}^{N_c}$ is mapped into a series of couples: (zero-run length, non-zero value) . As specified in [11] the subsequent bit-wise operations are applied. First, every non-zero value is set to its absolute value after we stored its sign. Then, the most significant bit (MSB) of a non-zero value is set to "+" (resp. "-") if the sign is positive (resp. negative). Finally, binary bit "1" (resp "0") in run lengths is replaced by "+" (resp. "-"). Binary bits, other than the MSB, in non-zero values are kept as they are. The use of 4 symbols in the alphabet removes ambiguity between run lengths and coefficient values. For example, if we consider the code:

$$M_{sparse}^{N_c} = (0, 0, 0, 0, 0, 0, 0, 5, 0, 0, 0, 0, 0, -18), \quad (9)$$

then applying the rules of stack-run encoding we obtain the code M with:

$$M = +++10++-+0100-, \quad (10)$$

where the first $+++$ string encodes for 7 zero's and $10+$ for 5, etc.

2.4. Arithmetic coding

While reading the code in (10), the decoder can switch from the *run length context* to the *non-zero value context* as it encounters the character 0 or 1. Obviously the decoder switch contexts in the opposite way as it encounters the character $+$ or $-$. This remark allows us splitting the code we obtained into 2 different files, one for the run length context and the other for non-zero value context. Considering the example in (10), we get in the first file F_0:

$$F_0 = +++1+-+0, \quad (11)$$

and in the second one F_1:

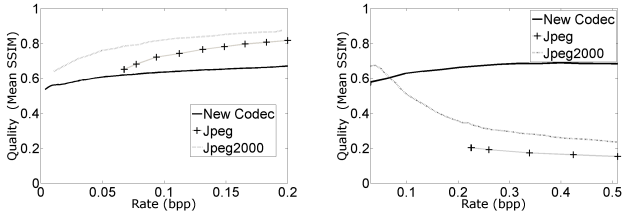
$$F_1 = 0+100-. \quad (12)$$

Thanks to this split, we get a first file F_0 mainly populated with $+$'s and $-$'s and F_1 mainly populated with 0's and 1's. Having these two files with enhanced contextual features, we can apply an arithmetic coder, which we said to be a first order coder, for compression with optimal performance. We concatenate the two resulting compressed files to obtain a single one F_c which will be the output of this stage.

3. RESULTS

Rate/quality curves are plotted to show the new codec performance. We remind the reader that, the stack run code as generated in (11) and (12) is passed through an arithmetic coder to obtain a unique compressed file F_c. We compute the rate as the size of F_c divided by the pixels number N^2 in the original image f. Quality is assessed by classical image quality criteria, PSNR and mean structural similarity measure (SSIM) [12], as we compare the reconstructed image $\tilde{f}_{N_c}$ to the original input f. Here, results are shown only in terms of mean SSIM, and PSNR results are available in [13] as both criteria show the same behavior.

Application to noiseless data We compared our results to the existing JPEG standards behavior under strong bandwidth restriction (rate < 0.2 bpp). Performance is comparable until $0.07\,bpp$ image rate (see Figure 3(a)), which shows our algorithm to have encouraging performance though still less efficient than JPEG2000. In addition, as each spike e_k encodes for the whole image, we avoid the block effect of JPEG compression, though this implies a smoothing effect on the image [13].



(a) Noiseless data (b) Noisy data

Fig. 3. Rate/Quality behavior: quantitative comparison between our new codec, JPEG, and JPEG2000 using mean SSIM as a quality measure. 3(a): noiseless input. 3(b): additive Gaussian noise ($mean = 0$, $variance = 0.05$ for a normalized image dynamic range)

Application to noisy data Our codec shows good robustness to noise compared to JPEG and JPEG2000 which also perform some kind of denoising while compressing data. Figure 3(b) shows the comparative performances of our codec and JPEG standards applied on an image strongly corrupted with a Gaussian additive noise, for low and median bit rates (< 0.5 bpp). Indeed, the wavelet-like retina behavior in the model [2] enables a better robustness.

Results show up to $6\,dB$ of gain in PSNR and 0.4 in mean SSIM for $0.25\,bpp$ of image rate compared to classic JPEG. As the rate increases JPEG codecs convey more high frequency (HF) signals, which are noise. This explains the decreasing rate/quality behavior of JPEG. As HF is encoded with loss in JPEG, artifacts appear in the decoded image. On the contrary, our new codec does not show artifacts because every spike is transmitted with no loss of information and encodes for the whole image. The scalability of our codec is monitored only by the choice of the number N_c of spikes to be encoded.

4. DISCUSSION

We have proposed a new bio-inspired codec for static images. First, the image is converted into a ROC code via a simplified retinal model, then a stack run coder is applied, followed by a first order arithmetic compressor.

Fig. 4. Robustness to noise: qualitative comparison between our new codec, JPEG, and JPEG2000 under the same rate restriction (here 0.27 *bpp*). Upper left: Lena with additive Gaussian noise ($mean = 0$, $variance = 0.05$ for a normalized image dynamic range). Upper right: coded/decoded image using the new codec. Lower left: coded/decoded image using JPEG. Lower right: coded/decoded image using JPEG2000.

The performance of this coding scheme was tested against well established JPEG standards, and we obtain encouraging results for low bandwidth transmissions, especially when dealing with noisy data. This compression scheme also offers interesting features such as scalability and reasonable complexity. Limitations have been observed in terms of rate/quality, when compared to JPEG2000 for noiseless data transmissions. Beyond the proposition of a new compression scheme, we would like to highlight a variety of potential avenues for future research efforts in this direction.

The first perspective concerns the retina coding model of our scheme. Although we focused on the latency time of the first spike, several models take into account the whole structure of spike trains. For example, it appears that burst or synchronies are features that could encode for the stimulus. This opens new perspectives to extend this model as soon as we are able to produce realistic spike trains. In particular, we will need to consider more realistic retina models converting videos into spike trains, such as [10]. The goal is then to reproduce spiking patterns as observed in real cell recordings, and establish how spikes are triggered by a stimulus then decoded [4].

The second perspective concerns compressing spikes. In this paper, even with a simplified representation of the spiking activity as a wave of spikes, classical approaches as stack run coding are not optimal. In the general case, with a continuous spiking activity, new ideas will have to be introduced. New bio-inspired compression schemes will have to take into account the features of the neural code that are the most relevant for the stimulus representation.

5. REFERENCES

[1] S. Thorpe, "Spike arrival times: A highly efficient coding scheme for neural networks," *Parallel Processing in Neural Systems and Computers*, pp. 91–94, 1990.

[2] R. Van Rullen and S. Thorpe, "Rate coding versus temporal order coding: What the retinal ganglion cells tell the visual cortex," *Neural Computation*, vol. 13, no. 6, pp. 1255–1283, June 2001.

[3] M. Antonini, M. Barlaud, P. Mathieu, and I. Daubechies, "Image coding using wavelet transform," *IEEE Transactions on Image Processing*, vol. 1, no. 2, pp. 205–220, April 1992.

[4] F. Rieke, D. Warland, R. de Ruyter van Steveninck, and W. Bialek, *Spikes: Exploring the Neural Code*, The MIT Press, Cambridge, MA, USA, 1997.

[5] T. Gollisch and M. Meister, "Rapid neural coding in the retina with relative spike latencies," *Science*, vol. 319, no. 5866, pp. 1108–1111, February 2008.

[6] D.J. Field, "What is the goal of sensory coding?," *Neural Computation*, vol. 6, no. 4, pp. 559–601, July 1994.

[7] B. Sen and S. Furber, "Maximising information recovery from rank-order codes," in *Proceedings of SPIE conference on Data Mining, Intrusion Detection, Information Assurance, and Data Networks Security*, Orlando, FL, U.S.A., April 2007, vol. 6570, p. 65700C.

[8] L. Perrinet, M. Samuelides, and S. Thorpe, "Coding static natural images using spiking event times: do neurons cooperate?," *IEEE Transactions on Neural Networks*, vol. 15, no. 5, pp. 1164–1175, September 2004.

[9] S.G. Mallat, "A theory for multiresolution signal decomposition : the wavelet representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 11, no. 7, pp. 674–693, 1989.

[10] A. Wohrer and P. Kornprobst, "Virtual retina : A biological retina model and simulator, with contrast gain control," *Journal of Computational Neuroscience*, vol. 26, no. 2, pp. 219–249, 2009.

[11] M.J. Tsai, J.D. Villasenor, and F. Chen, "Stack-run image coding," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 6, no. 5, pp. 519–521, October 1996.

[12] Z. Wang, A.C. Bovik, H.R. Sheikh, and E.P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600–612, avril 2004.

[13] K. Masmoudi, M. Antonini, and P. Kornprobst, "Spike based neural codes : towards a novel bio-inspired still image coding schema," Tech. Rep. RR-7167, INRIA Spohia Antipolis-Mditerrane, January 2010.