Image Compression by Self-Organized Kohonen Map

Christophe Amerijckx, *Associate Member, IEEE,* Michel Verleysen, *Member, IEEE,*
Philippe Thissen, *Member, IEEE,* and Jean-Didier Legat, *Member, IEEE*

***Abstract*—This paper presents a compression scheme for digital still images, by using the Kohonen's neural network algorithm, not only for its vector quantization feature, but also for its topological property. This property allows an increase of about 80% for the compression rate. Compared to the JPEG standard, this compression scheme shows better performances (in terms of PSNR) for compression rates higher than 30.**

***Index Terms*—Discrete cosine transforms, entropy coder, image processing, JPEG, self-organizing feature maps, variable length codes, vector quantization.**

I. Introduction

In the context of image processing, compression schemes are aimed to reduce the transmission rate for still (or fixed images), while maintaining a good level of visual quality.

One of the main common methods to compress images is to code them through vector quantization (VQ) techniques [1], [2]. The principle of the VQ techniques is simple. At first, the image is splitted into square blocks of $\tau \times \tau$ pixels, for example 4×4 or 8×8; each block is considered as a vector in a 16- or 64-dimensional space, respectively. Second, a limited number (l) of vectors (codewords) in this space is selected in order to approximate as much as possible the distribution of the initial vectors extracted from the image; in other words, more codewords will be placed in the region of the space where there are more points in the initial distribution (image), and vice-versa. Third, each vector from the original image is replaced by its nearest codeword (usually according to a second-order distance measure). Finally, in a transmission scheme, the index of the codeword is transmitted instead of the codeword itself; the compression is achieved if the number of bits used to transmit this index ($log_2 l$) is less than the number of initial bits of the block ($\tau \times \tau \times m$ if m is the resolution of each pixel).

Many authors used the Kohonen's algorithm [3] or self-organized feature map (KSOM) [4] to achieve the vector quantization process of image compression. Kohonen's algorithm is a reliable and efficient way to achieve VQ, and has shown to be usually faster than other algorithm and to avoid the problem of "dead units" that can arise for example with the LBG algorithm [5].

Kohonen's algorithm has however another important property besides vector quantization: it realizes a mapping between an input and an output space that preserves topology; in other words, if vectors are near from each other in the input space, their projection in the output space will be close too.

In the proposed compression scheme, we will use a two-dimensional Kohonen map corresponding to a grid of codewords (instead of a one-dimensional table in standard VQ), as the projection of an initial space including all vectors coming from blocks of the initial image.

The main aspect of this paper is to use the topology preserving property of KSOM. In a standard image indeed, we can make the hypothesis that two consecutive blocks, along the horizontal or vertical directions, will be similar in most cases, just because uniform regions in the image are generally much larger than the size of the blocks. According to the self-organization property of KSOM, two consecutive and similar blocks will be coded into similar codewords; the use of a differential entropic scheme to encode consecutive blocks will thus improve the compression ratio.

Other authors already used the topology-preserving property of KSOM's for different reasons. In [6], this property is used for progressive transmission of the image. In [7] it is used for further differential coding, as in this paper, but on images without discrete cosine transform (DCT) transform, and with a less-performant zeroth-order predictor (instead of first-order in this paper). In [8], an original neural network model is used instead of a first-order predictor, but on images without DCT transform. Finally in [9], the topology-preserving property is used to minimize the effect of transmission errors in noisy channels.

In this paper, we present a compression scheme based on DCT transform of the original image, vector quantization by Kohonen map, differential coding by first-order predictor, and entropic coding of the differences. Simulation results are provided, together with comparisons with similar compression scheme (DCT/VQ) without differential coding, and with a standard JPEG algorithm.

II. The Global Compression Scheme

The global compression scheme for lossy compression is described in Fig. 1. After a vectorization (transformation of image blocks into vectors), a DCT [10] and a low-pass filter first reduce the quantity of information by keeping only the low-frequency coefficients. Then, the vector quantization is performed, with another loss of information. Finally, the indexes of the codewords found by the vector quantizer are transformed by a differential coding, and the results are compressed by an entropic coder [11] ; these two last steps do

Manuscript received November 2, 1996; revised August 10, 1997 and January 7, 1998.

C. Amerijckx, M. Verleysen, and J.-D. Legat are with Université catholique de Louvain, Microelectronics Laboratory, B-1348 Louvain-la-Neuve, Belgium.

P. Thissen is with VLSI Technology, Sophia Antipolis, France.

Publisher Item Identifier S 1045-9227(98)02770-2.

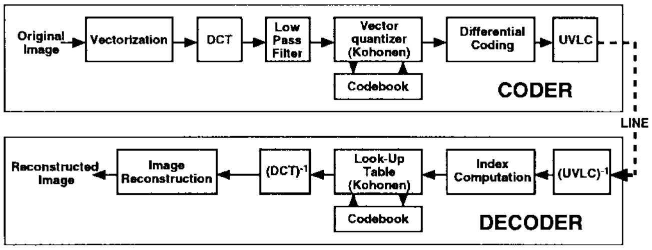


Fig. 1. Global compression scheme for lossy compression.

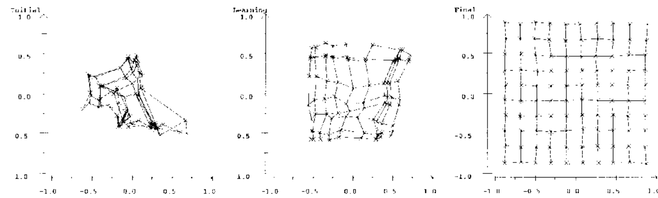


Fig. 2. Organization of a Kohonen map.

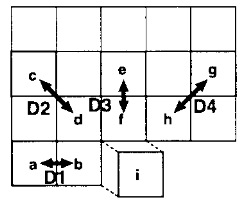


Fig. 3. Determination of the best direction to calculate the difference between blocks.

not introduce any loss in the information. The decompression scheme performs the same operations in the opposite way.

A. Image Preprocessing

The image is first decomposed into blocks (4×4 or 8×8 pixels as usual); the DCT transform is applied on each block, in order to eliminate a part of the information contained in the image, that is, high frequencies not visible to human eyes.

The DCT transform of a τ by τ pixels block is again a τ by τ block. However, in the transformed block, low-frequency coefficients are grouped in the upper-left corner, while high-frequency ones are grouped in the lower-right corner. The low-pass filter on the transformed block will thus keep only the c coefficients nearest from the upper left corner, with $c \leq \tau^2$; the remaining $\tau^2 - c$ coefficients are disgarded, supposing that they do not contribute too much to the visual quality of the image.

B. Kohonen's Self-Organizing Maps

As mentioned in the introduction, the goal of this algorithm is to create a correspondence between the input space of stimuli and the output space constituted of the codebook elements, the codewords, or neurons. After learning [12], these last ones have to approximate the vectors in the input space in the best possible way.

All neurons, or codewords, are physically arranged on a square grid; it is thus possible to define k-neighborhoods on the grid, which include all neurons whose distance (on the grid) from one (central) neuron is less or equal to k.

Each of the M codewords is represented by its weight $\hat{X}_j \in R^N$, where N is the dimension of the space ($= \tau \times \tau$ in the compression scheme). For each presentation of an input vector $X \in R^N$ during the training phase, the index i of the codeword nearest from X is determined, according to the Euclidean distance

$$d(X, \hat{X}_i) = \min(d(X, \hat{X}_j)),\ 1 \leq j \leq M. \tag{1}$$

The selected neuron i, and all neurons in a k-neighborhood of neuron i, are then "moved" in the direction of the input vector X, according to the relation

$$\hat{X}_m(t+1) = \hat{X}_m(t) + \alpha(t)(X - \hat{X}_m) \tag{2}$$

where m represents the index of all neurons in the k-neighborhood of neuron i and $\alpha(t)$ the learning factor. $\alpha(t)$ and k must decrease during the learning to ensure a good convergence of the algorithm. The topology-preserving property of KSOM, resulting from this learning process, means that two vectors in the input space, close with respect to the Euclidean distance, will activate two close centroïds on the grid (Fig. 2).

KSOM's have two properties used in our compression scheme. First, it quantizes the space like any other vector quantization method, what constitutes a first (lossy) compression of the image. Then, the topology preserving property of KSOM, coupled with the hypothesis that consecutive blocks in the image will often be similar, and to a differential entropic coder, constitutes a second (nonlossy) compression of the information.

C. Differential Coding

As mentioned above, if we suppose that most parts of the image are smooth, a differential coding applied to the codewords after vector quantization will lead to "small" codes in average. The use of an entropic coder, which encodes these differences into variable-length words (i.e., words which will use fewer bits if the differences themselves are small), will thus lead to further compression.

1) Smooth Gradient Principle: Because of the entropic coder, the compression ratio will be higher if the difference between codewords are low. Instead of using a simple differential scheme (zeroth-order predictor) where each codeword is substracted from the codeword corresponding to the previously encoded block in the image (i.e., the one at the left of the current block), we will use the following principle (first-order predictor): we suppose that gradients in the image are smooth, and thus that the direction in which the differences between two successive blocks was minimum for already encoded blocks will be the same as the direction in which the difference is minimum for a new block to encode. In other terms, and with the notations in Fig. 3, we suppose that the minimum difference between blocks i and b, i and d, i and f, and i and h, will be, respectively, in the same direction ($D1, D2, D3$, or $D4$) as the minimum difference between already encoded blocks b and a, d and c, f and

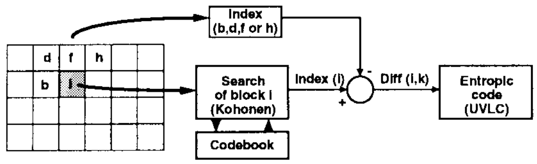


Fig. 4. Coding the difference between indexes.

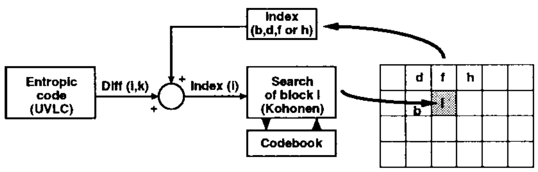


Fig. 5. Decoding the difference between indexes.

e, and h and g. In most cases, with images having smooth gradients (which is usually the case except in the regions with sharp variations, where the differential scheme does not bring any advantage), the selection between four possible directions will give smaller differences than a conventional scheme. Moreover, the selected direction does not have to be coded: since blocks a to h have already been transmitted when coding block i, the direction of the minimal difference between them can be computed with the already existing blocks, and not being transmitted.

The coding of a block is summarized in Fig. 4. The best direction is first computed according to the above scheme, and block b, d, f, or h is selected. The new block i is then encoded by the Kohonen's algorithm, and the difference between the two indexes is sent to the entropic coder.

The decoding is illustrated in Fig. 5. The best direction is first computed with the already decoded blocks; the sum of the selected index and of the transmitted difference is then the index of the new block i to decode, and is converted again in an image block by the look-up table created with Kohonen's algorithm.

It must be mentioned that the scheme proposed in Fig. 3 is only valid when the two first lines have already been encoded. For the two first lines, a classical differential scheme is used.

D. Entropic Coder: UVLC

Run length coding (RLC) and variable length coding (VLC) are widely used techniques for lossless data compression. We used an entropy coding system combining these two techniques in order to achieve a higher compression ratio. This system is similar to the one described in [11].

III. Simulations and Results

For the simulation of the whole compression process (see Fig. 1), we used 4 × 4 points image blocks; the Kohonen algorithm was trained with a decreasing function. While the simulations have been carried out on different images with similar results, the conventional Lena image is used in this paper for illustration purposes.

TABLE I
PSNR (in decibels) After the Low-Pass Filtering

	DCT2	DCT3	DCT4	DCT5	DCT6	DCT7	DCT8
Lena	25.09	26.54	26.83	27.87	29.35	30.07	31.15

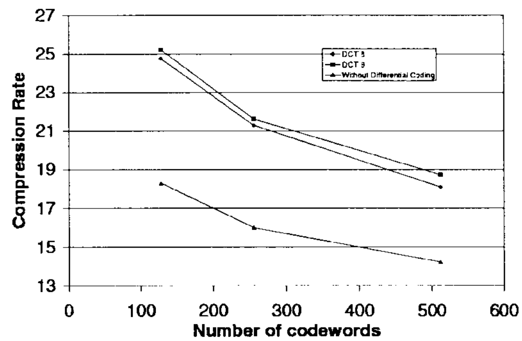


Fig. 6. Compression rate evolution with the number of codewords.

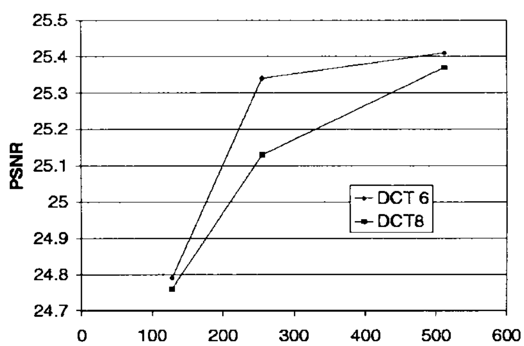


Fig. 7. Peak signal-to-noise ratio evolution with the number of codewords.

First, we have to show the consequences of the low-pass filtering: by removing a part of the high frequencies, we delete a part of the information contained in the image. An immediate consequence of this is a reduction of the peak signal to noise ratio (PSNR), even though the image visual quality remains more or less unchanged. In other words, before compression, the image quality will be degraded by the filtering. We can see in Table I the evolution of the PSNR on the Lena image, when the number of DCT coefficients kept after the low-pass filtering varies from "2" to "8."

To keep an acceptable subjective quality of the image, six and eight DCT coefficients will be kept in the next simulations (PSNR of around 30 dB before vector quantization).

The tests made with our compression scheme consist of a variation of the codebook size for a given cut frequency (six or eight DCT coefficients). The obtained compression rate in function of the codebook size (differential coding) are shown in Fig. 6 while their associated PSNR are given in Fig. 7.

As a comparison, the compression rate without differential coding (and thus without use of the topology-preserving property of KSOM's) is equal to $CR = (16 \times 8)/Nbr$, where each bit of the image is coded on eight bits, and Nbr is the number

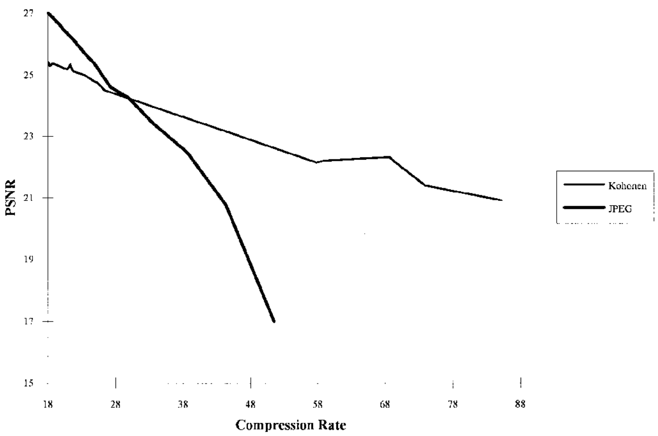


Fig. 8. Comparison of PSNR for the proposed lossy compression scheme and the JPEG algorithm.

of bits necessary to code all codewords (seven for $128 = 2^7$ codewords, eight for 256, and nine for 512).

Two remarks can be made.

- The compression rate does not vary too much with the cut frequency (Fig. 6): the difference is about 5% between DCT6 and DCT8.
- The signal to noise ratio (PSNR) does not vary too much with the cut frequency (Fig. 7). The difference between DCT6 and DCT8 is about 1%.

The fact that we obtain a better PSNR for the DCT6 can be explained by the fact that the space quantized with DCT6 has a lower dimension (dimension six); by this way, with a codebook of given size, the space can be better quantized than with the DCT8 one (dimension equal to eight). Nevertheless, the image quality of the DCT6 image, before the compression step, is lower (1.8 dB for Lena in Table I) than the DCT8 one.

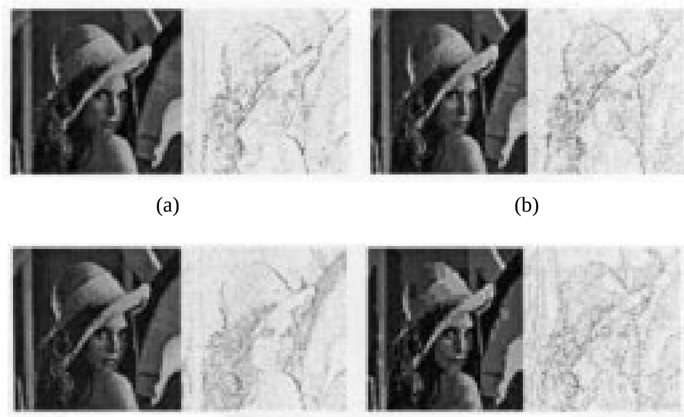
Differences between DCT6 and DCT8 in Fig. 6 are not significant, but the higher compression rate for DCT8 could be explained by a better "organization" of the Kohonen map, because the input vectors (in dimension 8 instead of 6) better represent the "correlated" image vectors.

Due to the use of Kohonen's self-organization relation, the compression rate is increased by about 80%. It is an important result showing the effectiveness of the proposed compression scheme.

A. Comparison with JPEG

To give an idea of the performance of the proposed coder-decoder using Kohonen maps, we compare our results with those obtained with the JPEG standard [13]; the JPEG experiments were carried out with default tables for the quantization process.

Fig. 8 compares the compression rates obtained by our proposed compression scheme (Kohonen on the figure) and by the JPEG algorithm. We can see that our approach gives a higher PSNR as soon as the compression rate is higher than about 25. This simulation was carried out on the Lena image. From a visual point of view, we can compare the Lena image (and the difference image) compressed by our method and by the JPEG algorithm, in Fig. 9, for a compression rate of about 25. The PSNR is slightly higher for the JPEG, but the visual quality of the image is slightly better for our compression scheme. Fig. 9 shows the Lena image compression with our method and with the JPEG, for a compression rate of about 38. In this case, we can see that the PSNR is higher for our method, and claim with no doubt that the visual quality is much increased.



(a) (b) (c) (d)

Fig. 9. (a) Lena image compressed by the proposed method, compression rate = 25.22, PSNR= 24.7dB. (b) Lena image compressed by the JPEG algorithm, compression rate = 25.04, PSNR = 25.3dB. (c) Lena image compressed by the proposed method, compression rate = 38, PSNR = 24dB. (d) Lena image compressed by the JPEG algorithm, compression rate = 38.55, PSNR = 22.46dB.

IV. Conclusion

In this paper, we proposed a new compression scheme based on the use of the organization property of Kohonen maps. It is based on the fact that consecutive blocks in an image are often similar, and thus coded by similar codewords with a vector quantization algorithm. The Kohonen map organization property makes the indexes of the coded vectors similar too, and, using an entropy coder, this property is used to increase in a significant way the compression ratio, for a given image quality (in a lossy compression scheme). The same method can also be used in a lossless compression scheme. Comparisons with JPEG also show that the quality of a compressed image is better with our proposed scheme, for compression ratios greater than about 25.

References

[1] R. M. Gray, "Vector quantization," *IEEE Acoust., Speech, Signal Processing Mag.*, pp. 9–31, Apr. 1984.

[2] A. Gersho and Robert M. Gray, *Vector Quantization and Signal Compression*. London: Kluwer, 1992.

[3] N. Nasrabadi and Y. Feng, "Vector quantization of images based upon the Kohonen self-organizing feature maps," in *IEEE Int. Conf. Neural Networks*, San Diego, CA, vol. 1, 1988, pp. 101–108.

[4] O. T.-C. Chen, B. J. Sheu, and W.-C. Fang, "Image compression using self-organization networks," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 4, pp. 480–489, Oct. 1994.

[5] Y. Linde, A. Buzo, and R. M. Gray, "An algorithm for vector quantizer design," *IEEE Trans. Commun.*, vol. COMM-28, pp. 84–95, Jan. 1980.

[6] E. A. Riskin, L. E. Atlas, and S. R. Lay, "Ordered neural maps and their applications to data compression," in *Neural Networks for Signal Processing '91, IEEE Wkshp.*, pp. 543–551.

[7] S. Carrato, G. L. Sicuranza, and L. Manzo, "Application of ordered codebooks to image coding," in *Neural Network for Signal Processing '93, IEEE Wkshp.*, pp. 291–300.
[8] G. Burel, "A new approach for neural networks: The scalar distributed representation," *Traitement du Signal*, vol. 10, no. 1, pp. 41–51, Apr. 1993.
[9] T. Kohonen, E. Oja, O. Simula, A. Visa, and J. Kangas, "Engineering applications of the self-organizing map," in *Proc. IEEE*, Oct. 1996, vol. 84, pp. 1358–1384.
[10] N. Ahmed, T. Natarajan, and K. R. Rao, "Discrete cosine transform," *IEEE Trans. Comput.*, vol. C-23, pp. 90–93, Jan. 1974.
[11] Shaw-Min Lei and Ming-Ting Sun, "An entropy coding system for digital HDTV applications," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 1, pp. 147–155, Mar. 1991.
[12] T. Kohonen, *Self-Organization and Associative Memory*, 3rd ed. Berlin: Springer-Verlag, 1989.
[13] G. K. Wallace, "The JPEG still picture compression standard," *Commun. ACM*, vol. 34, no. 4, pp. 32–34, Apr. 1991.

Christophe Amerijckx (S'95–A'95) received the bachelor's degree in electrical engineering from the Catholic University of Louvain, Louvain-la-Neuve, Belgium .

He is working under the frame of the European Community's ESPRIT Mivip project in rehabilitation of vision for blind people and working toward the Ph.D. degree in this field in the Microelectronics Laboratory at the Catholic University of Louvain. His research activities and interests include image processing, RISC architectures, FPGA architectures, and VLSI implementations.

Michel Verleysen (S'87–M'92) received the bachelor's degree in engineering and the Ph.D. degree in microelectronics from the Catholic University in Louvain, Louvain-la-Neuve, Belgium, in 1987 and 1992, respectively.

He is a Research Associate of the Belgian National Fund for Scientific Research, and Invited Lecturer at the Catholic University in Louvain. He was an Invited Professor at the E.P.F. Lausanne in 1992. He has been an author and coauthor of more than 50 publications in the field, and wrote a book in the "Que Sais-je?" series (in French).

Dr. Verleysen organizes the European Symposium on Artificial Neural Networks, and is the Editor-in-Chief of *Neural Processing Letters*. He is a member of the International Neural Network Society.

Philippe Thissen (S'90–M'92) received the bachelor's degree in electrical engineering from the Universite catholique de Louvain, Belgium, in 1992 and the Ph.D. degree in microelectronics from the same university in 1996. His doctoral research concerned digital and mixed signal implementations of neural networks for classification purposes.

Since 1996, he joined VLSI Technology, Sophia Antipolis, France, where he is involved in analog and mixed-signal designs for wireless products in submicron technology.

Jean-Didier Legat (S'79–M'81) was born in Brussels, Belgium, on April 15, 1957. He received the bachelor's and master's degrees in engineering from the Université Catholique de Louvain, Louvain-la-Neuve, Belgium in 1981. He received the Ph.D. degree in February 1987.

From 1981 to September 1984, he was granted a Belgian fellowship to undertake research in the field of document processing at the Microelectronics laboratory at the Université Catholique de Louvain. At the end of 1984, he joined the Belgian Company Anfima with the proposal of creating a new company which would design and commercialize integrated devices for image processing. In April 1987 a company called Image Recognition Integrated Systems (I.R.I.S.) was formed. He was cofounder of I.R.I.S. and was Vice-President. In October 1990, he came back to the Microelectronics Laboratory of Louvain-la-Neuve. He is presently a full Professor. His current interests are computer architecture, parallel and distributed systems, and design of embedded integrated circuits in the areas of artificial neural networks, digital signal processing, computer vision, and pattern recognition.