# PERMUTATIVE VECTOR QUANTIZATION - APPLICATION TO IMAGE COMPRESSION

*J. Skowronski and I. Dologlou*

Laboratoire des Signaux et Systèmes, CNRS-ESE, GDR TDSI
Plateau de Moulon, 91192 Gif-sur-Yvette,
France

**ABSTRACT**

This paper describes the Permutative Vector Quantization (PVQ) scheme as a special case of a more general structurally constrained Vector Quantization concept. This concept makes it possible to increase the vector dimensions beyond the technical bounds of conventional VQ and to exploit, by means of this, the inter-pixel correlations in large image blocks. Furthermore, a codebook design algorithm adapted to pemutative VQ is proposed and it is shown experimentally that the coding performance of conventional VQ can be improved using the present scheme.

## 1. Introduction

Vector Quantization (VQ) is a well-kown method used for data compression. In the case of image compression [1], the image is devided into blocks of size $n \times n$ ($n = 3, 4$ usually). These blocks are then approximated by the closest (in the sense of a given distance measure) block taken from a codebook. Since the aim of the method is to exploit correlations between the pixels of the blocks, the codebook is adapted to the image statistics by means of codebook design algorithms, like the well-known LBG algorithm [2].

It is a result of Shannon's rate distortion theory [3] that for a given rate $r$ the distortion of the vector quantized image decreases with growing block dimensions $n$. On the other hand, again for a given rate $r$, the number of codebook vectors grows exponentially with $n$. This results not only in an exponentially increasing storage complexity, but also in an exponentially growing nearest neighbor (NN) search complexity. Furthermore, a reliable codebook construction becomes unfeasible, since large training sequences are necessary to represent the statistics of the signal blocks. The use of large block dimensions is therefore restricted by these technical bounds. Much work has been done in the past to overcome the complexity problems by imposing certain structural constraints onto the codebook. By means of these structures, the number of codewords to be stored can be reduced and/or the nearest neighbor (NN) search can be accelerated. In this sense the Product Code method [4], Lattice Vector Quantization [5], tree structured VQ [6] and several other VQ schemes [7] have been proposed to counteract the complexity problems. For equal block size, the performance of these alternative methods is generally inferior to that of the classical VQ. This is due to the observation, that the structural constraints imposed onto the codebook are usually not verified by the probability density function of the signal blocks. It is therefore essential to apply only constraints that are apparent in the signal statistics in order to minimize the suboptimality.

This point of view leads us to a concept of vector quantization which will be presented in the following section. It will be refered to as structurally constrained Vector Quantization. Permutative Vector Quantization (PVQ), which is investigated here for the compression of still images, can be considered as a special case of this approach. A presentation of the method as well as a discussion of its physical sense is given in section 3. The original paper [8], which presented PVQ for speech compression, did not propose an adapted codebook design algorithm, taking into account the specific structure of the coder/decoder. We will therefore elaborate such a clustering algorithm in section 4. Finally, the performance of permutative Vector Quantization is compared to classical VQ in section 5.

## 2. Structurally Constrained Vector Quantization

Let us consider a codebook vector $\mathbf{y}_i$, $i = 1, \ldots, N$. Instead of coding an incoming vector $\mathbf{x}$ with only this codevector, we could imagine to derive a whole set of codevectors

$$\bar{Y}_i = \{\mathbf{y}_{i,j} | \mathbf{y}_{i,j} = f_j(\mathbf{y}_i), j = 1, \ldots, \bar{N}\} \qquad (1)$$

by some structural relations $f_j$, $j = 1, \ldots, \bar{N}$. Optimality is achieved, if the codebook of $N \times \bar{N}$ vectors created in this manner corresponds to the classical



0-7803-2431-5/95 $4.00 © 1995 IEEE

VQ codebook. In general, this is not possible, since the structurally constrained vector quantizer has only $N + \bar{N}$ degrees of freedom, compared to the $N \times \bar{N}$ of the optimal quantizer. Hence, the optimal choice of the basic codevectors $\mathbf{y}_i$ and the relations $f_j$ is important to compensate for the imposed constraints. The codevectors $\mathbf{y}_i$ and the functions $f_j$ should minimise the global distortion $D$ of the vector quantizer applied to a training set $S = \{\mathbf{x}_l, l = 1, \ldots, |S|\}$ of $|S|$ training vectors, that is,

$$\min_{\mathbf{y}_i, f_j}(D(\mathbf{y}_i, f_j, S)) \qquad (2)$$

has to be solved. Due to the infinity of possible structural relations $f_j$, it is obviously impossible to determine the general solution to this problem. Several simplifications can be considered. Well-known examples are the Gain-Shape vector quantizer proposed in [4] or the mean removed Gain-Shape vector quantizer [9] which consider 'first-order' polynomials as structural relations. In these cases the functions $f_j$ are given by $f_j = b_j\mathbf{y} + a_j$, where $\mathbf{y}$ is a normalized, i.e. basic, codevector and $b_j, a_j$ a set of polynomial coefficients.

PVQ, on the other hand, proposes an alternative way of approximating the solution of (2), that is it simply imposes *a priori* physically justified relations $f_j$. Taking into account these relations $f_j$, the basic codevectors $\mathbf{y}_i$ can then be determined using an adapted clustering algorithm presented in section 4.

Let us turn our attention to the complexity problems of Vector Quantization. Using structurally constrained VQ, the storage complexity can be reduced by a factor of $\bar{N}$ to

$$N = \frac{2^{n^2 r}}{\bar{N}}. \qquad (3)$$

As to the NN search complexity, different strategies can be considered, depending on the functions $f_j$. Most beneficial is the case where the optimal $f_j$ and $\mathbf{y}_i$ can be found separately or sequentially as in the case of Gain-Shape coding [4]. In other cases, such as PVQ, fast algorithms exist, that find the best structural relation $f_j$ for a given codevector $\mathbf{y}_i$ in less than $O(\bar{N})$ operations.

## 3. Permutative Vector Quantization

This section outlines the structural relations used in the case of PVQ. For this, let us consider a codevector $\mathbf{y}_i$ which is devided into $m$ subvectors of equal size. The subvectors could be the lines or columns of the image blocks or simply subblocks. A subcodebook $Y_i$ can be derived from this vector by generating all the permutations of the $m$ subvectors, which will give us

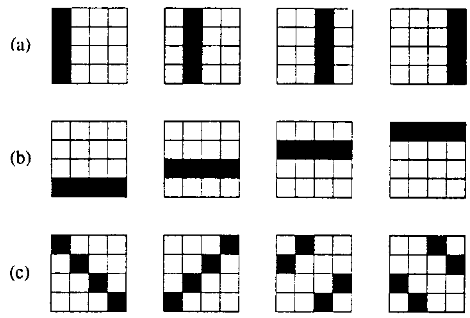


Figure 1: Some Permutations of contours blocks; (a) vertical contour, (b) horizontal contour, (c) diagonal contour

$m!$ vectors. The size of the basic codebook to be stored becomes therefore

$$N = \frac{2^{n^2 r}}{m!}$$

Also the search complexity is reduced, since fast NN search algorithms exist [8] [10] that find the closest permutation in $O(m^3 log(m))$ instead of $O(m!)$ operations.

The special interest of this method for image coding can be seen in figure 1, where several permutations of the lines (columns) of a contour block are shown. Different orientations and positions of the contours can therefore be obtained by permutating a few basic blocks. This method appears to be even more adapted for video coding, since from one frame to the next the motions of the scene can be followed by permutating the lines or columns of the image blocks. A further possibility is to permutate sub-blocks of the image block rather than lines or columns, since in this case the correlation within a block is exploited more efficiently.

## 4. Codebook Design Algorithm

In order to fully exploit the coding gain due to the increased block size, it is essential to include the structural constraint of the method on the codebook during the design of the code. The optimal way of doing this is to use an algorithm derived from the LBG algorithm [2]. The principle of the LBG is the successive optimization of the coder, while keeping the decoder fixed, and of the decoder, while keeping the coder fixed. In practice, this means that at each iteration, the algorithm performs first the partitioning of the training set vectors $\mathbf{x}_l$, $l = 1, \ldots, |S|$, into $N$ clusters using the codevectors of the previous iteration. Then the new codevectors are determined as the centroids of the clusters.

**(0) Initialisation :**

- iteration : $q = 0$
- initial codebook : $Y(0) = \{\mathbf{y}_i(0), i = 1, \ldots, N\}$
- initial global distortion : $D(-1) \to \infty$

**(1) Coder Optimization :**

- clustering of the permutated training set $S_p = \{P_j(\mathbf{x}_l), l = 1, \ldots, |S|, j = 1, \ldots, m!\}$ into $N$ classes $\mathcal{C}_i, i = 1, \ldots, N$, with the classification rule :

$$P_h(\mathbf{x}) \in \mathcal{C}_k \text{ iff } d(P_h(\mathbf{x}), \mathbf{y}_k) \leq d(P_j(\mathbf{x}), \mathbf{y}_i), \quad \forall i = 1, \ldots, N; j = 1, \ldots, m!$$

- determine $D(q)$ and stop if

$$\frac{D(q-1) - D(q)}{D(q)} < \text{threshold}$$

**(2) Decoder Optimization :**

- optimal decoder for the classification of step 1

$$\mathbf{y}_i(q) = \text{centroid}(\mathcal{C}_i), i = 1, \ldots, N$$

- set $q \mapsto q + 1$ and goto step 1

Table 1: The Codebook Design Algorithm PLBG

It is the optimization of the coder that changes during the PVQ codebook design. In this case not just each training set vector $\mathbf{x}_l$, $l = 1, \ldots, |S|$, but a permutation of each training set vector $P_j(\mathbf{x}_l)$, $l = 1, \ldots, |S|$, $j = 1, \ldots, m!$, is appointed to one of the $N$ clusters such as to minimise the distortion of the clustering. More precisely, for a signal vector $\mathbf{x}_l$ the distance of each of its $m!$ permutated versions $P_j(\mathbf{x}_l)$ to each of the $N$ basic codevectors $\mathbf{y}_i$ is determined. If the vector pair $P_h(\mathbf{x}_l)$ and $\mathbf{y}_k$ gives the minimum distance, the permutated codevector $P_h(\mathbf{x}_l)$ is appointed to class $\mathcal{C}_k$. Note that this classification can be done using the fast NN search algorithm proposed in [8] or [10]. It should also be noted, that the coder actually replaces a signal vector by the closest permutation of one of the basic codevectors, i.e. it determines the distances $d(\mathbf{x}_l, P_j(\mathbf{y}_i))$. However, it has been shown in [11] that this is equivalent to the determination of $d(P_j(\mathbf{x}_l), \mathbf{y}_i)$, as is done here.

| $n$ | $N$ | method | PSNR/dB | |
|---|---|---|---|---|
| | | | Salesman | Puppet |
| 3 | 42 | CVQ | 28.4 | 21.17 |
| 4 | 768 | CVQ | 31.36 | 22.48 |
| 4 | 32 | LBG | 27.2 | 19.90 |
| 4 | 32 | PLBG | 27.37 | 20.16 |
| 5 | 269 | PLBG | 29.16 | 20.59 |
| 6 | 4322 | PLBG | 32.57 | 21.33 |

Table 2: Coding Results; CVQ - conventional VQ, PLBG - permutative VQ using the PLBG algorithm, LBG - permutative VQ using the LBG algorithm

Subsequently, as for the LBG algorithm, the new codevectors, i.e. cluster centroids, are calculated. Table 1 shows the codebook design algorithm referred to as PLBG (permutative LBG). Its demonstration of convergence is straightforward, since it is easily seen from table 1 that neither during the coder optimization step nor during the decoder optimization step the global distortion can be increased.

## 5. Experimental Results

In the following, the permutative VQ scheme is compared with conventional VQ. The image coding results stated hereafter show that by using the permutative VQ scheme, it is possible to obtain higher signal to noise ratios compared to conventional VQ when the bit per sample rate is fixed or, alternatively, lower bit rates for a given signal distortion. Table 2 lists the PSNR-values for the two coding methods at a constant rate $r = 0.6$bpp. In the present case, the columns of the image blocks are permutated, but it was observed, that similar results are obtained, when the lines are permutated. The codebooks are calculated on the basis of four different images, including the Salesman image but not the Puppet image. Naturally the coding quality increases with the image block sizes and it is clear that for equal block size, the classical VQ gives better results than the permutative VQ. However, since complexity becomes prohibitive for block sizes greater than $n = 4$, at the indicated rate, no further improvements are possible with the standard method. On the other hand, permutative VQ makes greater image blocks possible and for $n = 6$ the results of classical VQ is outperformed by more than 1dB in the case of the Salesman image. This numerical improvement goes along with a substantial increase in visual quality as can be seen in figure 2, the Salesman image ($288 \times 360$) coded by conventional VQ (a) and coded by permutative VQ (b) is illustrated. Note that, no block effect can be observed in the image coded by the permutative VQ scheme, despite the fact that relatively large blocks have been used. Furthermore it has been observed that contours are well represented by the new VQ method. The effects of permutations on contours, described in section 3., could be considered as an explanation of this fact.

In the case of the Puppet image, which did not belong to the training set, the same observations could be made, but we see from table 2 that greater vector dimensions ($n > 6$) are needed to improve the coding results of conventional VQ.

Table 2 also indicates the improvement of the permutative VQ scheme by the codebook optimization algorithm presented here. It has also been observed that the coding results vary slightly when the initial conditions are changed. These variations are comparable to those observed in the case of the LBG algorithm.

## 6. Conclusions

The paper presented structurally constrained VQ, a concept allowing the design of a whole series of VQ schemes. We restricted ourselves to a special case, permutative VQ, and we showed that it can improve the compression results of conventional Vector Quantization. Further research is being carried out on the development of alternative coding schemes derived from the structurally constrained VQ concept, with the aim of further reducing the coding complexity.

## 7. References


[1] N. Nasrabadi and R. King, "Image coding using vector quantization: A review," *IEEE Tr.Com.*, vol. 36, pp. 957–971, Aug. 1988.

[2] Y. Linde, A. Buzo, and R. Gray, "An algorithm for vector quantizer design," *IEEE Tr.Com.*, vol. 28, pp. 84–95, Jan. 1980.

[3] T. Berger, *Rate-distortion theory*. New Jersey: Prentice Hall, Englewood Cliffs, 1971

[4] M. Sabin and R. Gray, "Product code vector quantizers for waveform and voice coding," *IEEE Tr. on ASSP*, vol. 32, pp. 474–488, June 1984.

[5] M. Antonini *et al.*, "Image coding using lattice vector quantization of wavelet coefficients," in *ICASSP-91*, (Toronto, Canada), pp. 2273–2276, May 1991.

[6] A. Benazza, *Quantification Vectorielle en Codage d'Images*. PhD thesis, Université de Paris-Sud, Centre d'Orsay, 1993.

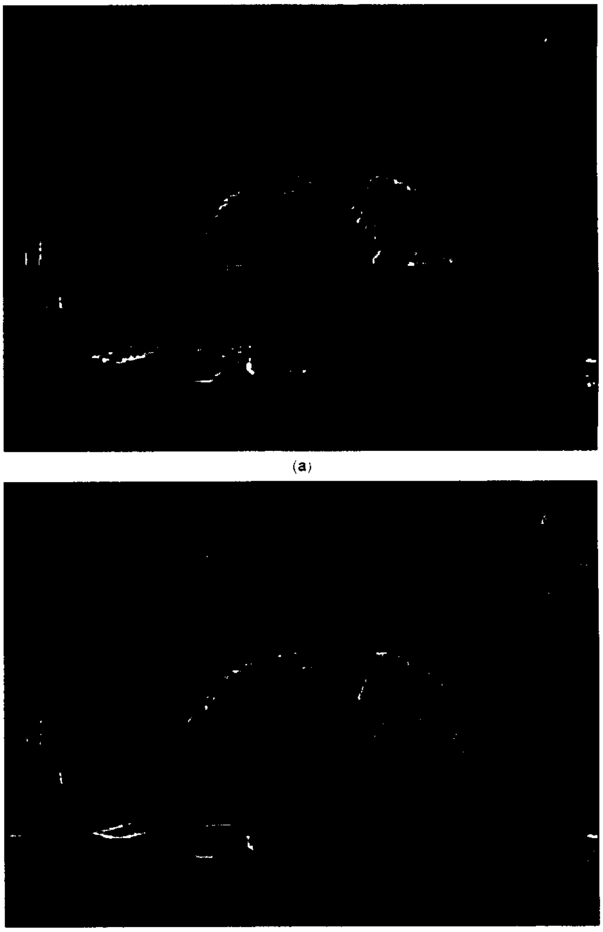
[7] A. Gersho and R. Gray, *Vector Quantization and Signal Compression*. Boston: Kluwer Academic Publishers, 1992.

[8] A. Nandi *et al.*, "Varation of vector quantization and speech waveform coding," *IEE Proceedings-I*, vol. 138, pp. 76–80, April 1991

[9] T. Murakami, K. Asai, and E. Yamazaki, "Vector quantizer of video signals," *Electronics Letters*, vol. 7, pp. 1005–1006, Nov. 1982.

[10] H. Kuhn, "The hungarian method for the assignment problem," *Naval Research Logistics Quaterly*, vol. 2, pp. 83–97, 1955.

[11] J. Skowronski and I. Dologlou, "Image compression using permutative vector quantization," submitted to *IEEE Tr.on Image Processing*.

(a)

(b)

Figure 2: (a) Salesman coded with conventional VQ. (b) Salesman coded with permutative VQ