

A New Approach of Vector Quantization Applied to Image Data Compression and Texture Detection.

Laurent D. COHEN

**INRIA, Domaine de Voluceau, Rocquencourt
B.P. 105, 78153 Le Chesnay Cedex, France.**

**Proceedings of 9th International Conference on Pattern
Recognition, Rome, 1988**

ABSTRACT

This paper describes a new method based on Vector Quantization (VQ). Instead of quantizing blocks of the image by the vectors, the vectors are first rescaled to be the closest to the block. We are looking for the most representative patterns of the image, two blocks having the same pattern if they are similar after translation and scaling in gray grey levels. A new way of VQ is also introduced using a little image which captures all the useful information instead of separated quanta. The vectors to consider in the codebook become the set of all the possible blocks obtained by scanning a blocksize mask along the rows of this little image. This can be applied to picture encoding and texture detection.

1 Introduction

Several approaches of image data compression are related to the decomposition of an image or a sub-block in a well chosen base of the vector space of the matrices . Compression is then obtained when the image can be decomposed on a small number of vectors of the base. Most of the time it means that the orthogonal projection of the image on a vector space of small dimension gives a sufficiently good approximation.

The transform applied to the matrix is the first of the three operations which form the general scheme of image data compression as presented in [2] :

Mapper-Quantizer-Coder. The role of the mapper is to make the two other operations more efficient.

For example for orthogonal transforms, the image is decomposed in a base which usually represents different scales of frequencies. The goal is to find for a given family of images, a base such that all the images of this family can be well approached by giving only information on a small number of coordinates of the decomposition. This means that the total energy is concentrated on the first vectors of the base. The optimal base which packs the maximum average energy into the first k vectors of the base for all k is the Karhunen-Loeve transform. Although it is optimal, it is usually difficult to compute and is often approached by other unitary transforms such as Cosine, Fourier or Hadamard transforms (see [6] and [4]).

In predictive methods, the image is approached by its orthogonal projection in the space generated by the constant image and space translations of the original image. Assuming a Markovian model on the image justifies that only the projection on small translations are kept.

In this paper we want to find for a given image a set of vectors sufficient to project the sub-blocks of the image in the two dimensional vector spaces generated by two vectors among this set of computed vectors to minimize the mean square error. It turns out from study of orthogonal transforms that the mean of a block is the component with the most important energy. From this remark we only consider in this paper the two dimensional vector space generated by the constant vector (all coordinates equal to 1) and one vector among the set of computed vectors. This corresponds to scale changes in the grey level values. This has effect to describe by the same pattern two blocks that are similar but which have different luminance.

The idea is to use vector quantization to determine the best vectors, that is the most representative vectors of the image. That means that we want to minimize the average Euclidean distance between a sub-block of the image and the set of two dimensional spaces. So each sub-block is encoded by the index of the most similar vector in the list and the two coefficients of the projection of the sub-block in the correspondent vector space. The set of vectors can either be a list of distinct vectors or the family of sub-blocks of a little image. In the latter case we compute a little image which contains the most representative patterns of the original image.

This technique is particularly efficient when the image is formed of repeated patterns which have different luminance levels. The algorithm extracts then from the image the characteristic elements it is composed of. Thus we obtain a description which gives on one hand the patterns which are

the closest to the structure of the image and on the other hand their distribution along the image.

The new aspect of this method is to project a vector into the direction of a quantum instead of merely comparing the vector to the quantum. This is equivalent to scaling in the range of grey levels. The scaling can also be made in the space directions. This idea is related to theory of wavelets introduced by Y.Meyer ([5]) and Scale-Space decomposition of A.Witkin ([9]). In either case the idea should be to take the most important components of the decomposition.

This allows us to find only one pattern to describe objects which are similar after scaling or after any of a set of admissible transformations.

2 Vector Quantization.

2.1 The classical algorithm.

VQ is a generalization of scalar quantization. Here the normed space V where we want to quantize is a finite dimensional space instead of the scalar range. Vectors of V represent sub-blocks of the image.

Let us recall what is quantization in mathematical terms: Suppose V is a normed space and X (the set of taken values) is a subset of V . We are looking for a “smaller” subset Q (the set of quanta) of V such that every element in X can be represented by an element of Q . This process which associate to each element in X an element in Q is the quantization. To be useful, we need at the same time the set Q to be as small as possible, to have few data to encode, and the sum of the quadratic errors, that is the distances between elements of X and their quanta in Q , to be minimal. Let us assume the dimension of a block is $n_1 \times n_2$. The image size is $N_1 \times N_2$ and it is divided into blocks which are the elements of the set X of vectors to be quantized. So we are looking for a “smaller” subset Q (the set of quanta) of blocks such that every element in X can be represented by an element of Q . The distance between two blocks usually chosen is the Euclidean norm. We have a set of test vectors x_1, \dots, x_n which are used to find the best quanta, that is the blocks q_1, \dots, q_p which minimize the MSE :

$$e = \sum_s \|x_s - q(x_s)\|^2.$$

where $q(x_s)$ is the quantum associated with x_s .

It is equivalent to find the set of optimal quanta or to separate the x_i 's into clusters. For that we use the Dynamic Cluster Algorithm (see [7] for a general survey of this algorithm).

The algorithm is based upon two remarks :

1. when the different groups of x_i 's are determined, the best quanta are the average values of the x_i 's in each group.
2. When the quanta are given, the x_i 's quantized by a quantum q_i are those that are closer to q_i than to any other quantum.

Consequently, the algorithm divides into the following steps :

1. The quanta q_1, \dots, q_p are initialized.
2. For i between 1 and n , we associate to x_i the quantum which is the closest to it. So the x_i 's are separated into p clusters.
3. We choose in each cluster a new quanta to represent the group.
4. We go back to step 2 if we want a more precise quantization.

2.2 Variations and comments.

Let us analyse the steps one by one :

1. The initialization is rather important since the process can become stationary by reaching a local minimum of the quadratic error which is not the desired result. The initialization can be done in different ways. The more simple is to take at random p values for the q_i 's either among the x_i 's or among the corresponding range of values. We can also take a multi dimensional uniform quantization in this range of values. But for an extra effort work we can obtain a much better initialization by a splitting algorithm (see [3]).

Another method of initialization is introduced in [8] but it is used there by itself as the whole algorithm. The idea is to fill the table of quanta as we read the sequence of test vectors. We choose at the beginning a threshold t of the admissible error.

- The first quanta is the first vector of the training sequence.
- Suppose that at some step we read the test vector x_i and have already defined k quanta ($k \leq p$). As before we then look for the closest quantum in the already existing list. If none of them are close enough to x_i (that is the least distance is greater than t) and there is one more quantum to be defined ($k < p$) then we set $q_{k+1} = x_i$ else we attribute to x_i the closest quantum and refresh it by a weighed average of the old quantum and x_i .

This method has the advantage to take only the number of quanta we really need and to pass only once over the sequence of test vectors.

- 2-3. As remarked at the beginning of the section the best quanta for a cluster to minimize the quadratic error is the mean of its elements. But it can happen that some values are not as close to its quantum as the rest of its group. This value moves the quantum from the other values of the group and a solution should be to select the best values in a group to determine the quantum of this group.

4. We have to decide when to stop. We can remark that the mean square error has to decrease at each step and since the number of regroupments is finite, the quanta have to become stationary. We can stop when there is no more variation of the quanta or the groups. If it is too long we can stop when we reach a given value either for the mean square error or for its variation between two steps.

2.3 VQ with projection.

So far, the similarity between a vector and a quantum was measured by the Euclidean norm. It may happen that a quantum is exactly a translation of another or that two quanta are homothetic. This leads us to consider a different way to compare a vector and the set of quanta.

In this section we do not use a different distance but instead of making the projection in the set of quanta, we project a vector in the set of vector spaces generated by each quanta. In fact the basic idea is to replace each vector by its projection in a small dimension vector space. To be interesting at the same time for compression rate and execution time we limit ourselves to two dimensional spaces. Furthermore as said in the introduction we impose the first vector to be the constant vector since it is known that it concentrates generally the maximum energy. Visually this corresponds to consider as in the same class two patterns that are similar but have not the same enlightenment.

Suppose a vector X and a quantum Q are given, we want to find the linear combination of Q which is the closest to X . If C is the constant vector with all components equal to 1, we call linear combinations of Q the vector space generated by Q and C . We are looking for two coefficients α and β which minimize the distance $\|X - \alpha Q - \beta C\|$.

We know that this minimum is achieved when $\alpha Q + \beta C$ is the orthogonal projection of X in the vector space generated by Q and C . So we have the two relations :

$$(X - \alpha Q - \beta C, Q) = 0$$

$$(X - \alpha Q - \beta C, C) = 0$$

If X is any vector then we call m_X its mean. That is $m_X = \frac{(X,C)}{(C,C)}$. By definition we have $(X - m_X C, C) = 0$.

From the second of the two equations of orthogonality we find that :

$$m_X - \alpha m_Q - \beta = 0.$$

By replacing β by its value and using $(X - m_X C, C) = 0$ and $(Q - m_Q C, C) = 0$, the first equation becomes :

$$(X - m_X C - \alpha(Q - m_Q C), Q - m_Q C) = 0$$

which gives then

$$\alpha = \frac{(X - m_X C, Q - m_Q C)}{\|Q - m_Q C\|^2}$$

$$\beta = m_X - \alpha m_Q.$$

With these values of α and β we have :

$$e = \|X - \alpha Q - \beta C\|^2 = (X - m_X C - \alpha(Q - m_Q C), X - m_X C)$$

since $X - \alpha Q - \beta C$ is orthogonal to both Q and C .

$$e = \|X - m_X C\|^2 \left(1 - \frac{(Q - m_Q C, X - m_X C)^2}{\|X - m_X C\|^2 \|Q - m_Q C\|^2}\right).$$

We see here that everything is expressed as a function of $Q - m_Q C$ and $X - m_X C$. So it makes computation easier when possible if we have the vector and quantum centered to solve the problem since we then have $m_Q = m_X = 0$. It means that we have to make the projection only in the vector line generated by Q since X and Q are then already orthogonal to C .

To simplify we are looking for quanta that are already centered and normed. Let us see how the algorithm is modified.

- step 1 : Initialization. To initialize the quanta, we use any of the usual initialization procedures and then center and norm the quanta.
- step 2 : To each vector we associate the quanta which gives the minimal error. To simplify the computation we center the test vectors before we begin.
- step 3 : refreshment of the quanta : If we want to keep on the spirit of the previous cases we have to choose for a group the vector Q which minimizes the sum of distances to all vectors of the group. For the simple Euclidean distance, the quantum to choose is the mean of the group. If we want to take the mean here we have first to scale all vectors of the group so that they have contribution of the same range to the quantum. If we know the quantum Q , it is an intuitive result that the mean of the orthogonal projections of Q on the vector lines corresponding to all the vectors of the group is exactly the direction of Q . This will be proven below. If we want a simple estimate of the optimal quantum, we can take something between the previous quantum and the mean of the projections on the vectors of the group for the previous quantum of that group.

As usual we can decide to take into account only those vectors that have a distance small enough.

- step 4 : we can stop when there is no more change in the positions attributions or in the global error.

Best quantum for a group of vectors.

Suppose a group is formed of X_1, \dots, X_n and we want to find Q which minimizes the global error

:

$$e(Q) = \sum_{i=1}^n d(X_i, Q).$$

Let us recall what happens with the Euclidean distance.

In the classical case, $d(X, Y) = \|X - Y\|^2 = (X - Y, X - Y)$ Since we know that there is a minimum, we have at this minimum $de(Q) = 0$, which means :

$$de(Q)R = 2\left(\sum_{i=1}^n (X_i - Q), R\right), \text{ so}$$

$$\sum_{i=1}^n X_i = nQ$$

So the best quantum is the mean of the vectors G .

We can also see that :

$$e(Q) = \sum_{i=1}^n \|X_i - G\|^2 + n\|G - Q\|^2$$

which is minimal only for $Q = G$.

Let us come back with the “projection” case.

We have now $d(X, Q) = \text{Min}_{\alpha \in \mathbb{R}} \|X - \alpha Q\|^2 = \|X - \frac{(X, Q)}{(Q, Q)}Q\|^2$. By definition of d we have $e(\alpha Q) = e(Q)$ so all the values are taken on the unit sphere which is compact and the minimum of e is achieved at a point Q on the sphere such that $de(Q) = 2\lambda Q$ by the theorem of Lagrange multipliers. We have thus :

$$e(Q) = \sum_{i=1}^n (X_i, X_i) - \sum_{i=1}^n (X_i, Q)^2, \quad (Q, Q) = 1$$

$$de(Q) = -2 \sum_{i=1}^n (X_i, Q)X_i = 2\lambda Q$$

Taking the scalar product with Q we have :

$$\lambda = - \sum_{i=1}^n (X_i, Q)^2 = e(Q) - \sum_{i=1}^n (X_i, X_i)$$

The last term being a constant we see that the minimum is achieved when $-\lambda$ is maximum. If we set

$$AQ = \sum_{i=1}^n (X_i, Q)X_i$$

then A is a linear symmetric operator and the best quanta is obtained for the direction of the greatest eigenvalue. We see that for the direction of the best quanta, the sum of projections of Q on X_i 's is in the same direction. We have $e(Q) = \sum_{i=1}^n (X_i, X_i) - (AQ, Q)$. We show that if Q' is the old quanta of the same group, the direction of steepest gradient at Q' is a linear combination of Q' and AQ' So the simple estimation we take is a direction between Q' and AQ' .

Suppose that the X_i can be written in the canonical base :

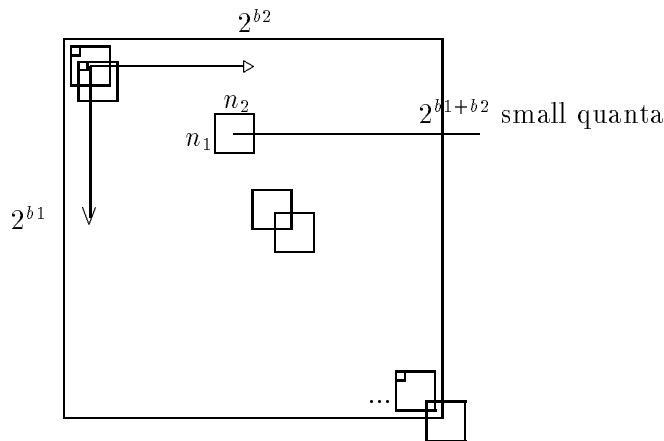
$$X_i = \sum_{j=1}^p X_{ij}e_j \quad 1 \leq n$$

where p is the dimension of the space in which the X_i 's are. If now we set $Y_j = (x_{1j}, \dots, x_{nj})$, $1 \leq p$, then the matrix of A is tMM where $M = [Y_1, \dots, Y_p]$. That is A is the Gram matrix of M , the matrix which columns are the Y_i .

2.4 A quantizer instead of quanta.

When we build a table of quanta from test vectors taken in the image, it may happen that some test vectors are almost the same with a slight space shift. This has two effects, either the vectors are close enough to be in the same class and then the quanta will not be very good, or the vectors will be in two different classes and then we must use two quanta for an information which is almost the same. The problem is that the quanta must be used exactly as they are and all the quanta are independent. A way to solve this problem is to set the quanta differently. Instead of taking a table of quanta we use only one big quanta. Let us call it the quantizer. This will be a little image which is assumed to collect all the information of the image the same way it was done for the list of quanta. But now we compare a vector to all the sub-blocks of the quantizer of the right size. We give to the quantizer a torus structure, that is we link the two horizontal edges and the two vertical edges together. So to each pixel of the quantizer corresponds a sub-block which has this pixel as left upper corner. The algorithm is then exactly the same as before.

Figure 1: Principle of the quantizer.



Let us say for example that blocks are 4×4 pixels and that the quantizer is 16×16 pixels. Then when we want to quantize a 4×4 block of the image we look for the sub-block of the quantizer which matches the best. So we compare our vector to the 256 (4×4) sub-blocks of the quantizer and choose the closest. So to each block of the image corresponds a sub-block of the quantizer. After we set this correspondence, we have for each sub-block of the quantizer a group of blocks of the image that match well with it. So each part of the quantizer is refreshed using the vectors of its group. Since these parts overlap the refreshment of one sub-block has an influence on all overlapping sub-blocks, that is 16 in our case. So everything happens as if the quantizer was split into 256 quanta with which we work as usual to form groups and then refresh the quanta. Then each part of the quantizer is refreshed using the new defined quanta. The overlapping of quanta makes us take a mean at each pixel of the values at this position of all quanta that pass over it. Thus for each pixel we have to take into account the vectors of the groups which correspond to

the 16 quanta that pass over this pixel. We can take the mean either on the refreshed quanta with some weights or on the blocks of the image in each group that match to this position.

Let us examine each step to point out the new aspects of the method.

- step 1 : Initialization. To initialize the quantizer, as before we have the choice of taking random values in the same range as the image or a part of the image or the concatenation of little parts of the image chosen at random. When we take random values we can choose them such that they have the same statistical properties as the original image.
- step 2 : To each vector we associate the position in the quantizer which gives the closest block.
- step 3 : refreshment of the quantizer : we have to refresh each pixel separately. To define a new value at a pixel we take the mean of all values at this pixel of vectors which passed over it. We can decide to take into account only those vectors that have a distance small enough.
- step 4 : we can stop when there is no more change in the positions attributions or in the global error.

This method is most useful for texture detection to capture a pattern which is bigger than the block size. It has advantage to take less room than the equivalent number of quanta. the ratio is the number of pixel per block. The results obtained are a lot better when we use this quantizer for VQ with projection presented in the previous section. This implies the modifications of both sections are necessary in the algorithm.

3 Applications.

3.1 Estimates of the compression rate.

Let us see how much compression we obtain with VQ.

As at the beginning of the section, we assume that we have an image of size $N_1 \times N_2$ pixels divided into blocks of size $n_1 \times n_2$. Each block is quantized and it is replaced by the index of the quanta of its class. So to rebuild the image we need to store the table of quanta and the list of indexes and coefficients. The rate in bits/pixel is :

$$\tau = \frac{b + c}{n_1 n_2} + \frac{2^b q n_1 n_2}{N_1 N_2}$$

where we suppose that the number of quanta is $p = 2^b$, so an index is coded on b bits and that in the table of quanta each pixel is coded on q bits, and c is number of bit necessary to encode the coefficients.

When we use a great quantum, the “quantizer”, the part corresponding to the encoding of the image is the same as before but the information about the quanta changes since it is divided by the size of a block. we have :

$$\tau = \frac{b + c}{n_1 n_2} + \frac{2^b q}{N_1 N_2}$$

where 2^b is still the number of quanta which is also the size of the quantizer. We shall see results in the next section.

3.2 Texture detection.

This method is interesting to be used on an image formed of repetitive patterns which are not totally identical and that have different luminance within the image. It can extract the useful information in the quantizer and the total pattern can be found in the quantizer. Thus we obtain a description which gives us on one hand the patterns which are the closest to the structure of the image and on the other hand their distribution through the image.

The reconstruction is all the more precise that the set of admissible transformations is large. Here the only transformations used are scale changes in grey levels. It would also be interesting to consider scale changes in the plane of the image. In [1], the set of admissible transformations includes also rotations which angles are multiples of $\frac{\pi}{4}$. This means that a pattern is the same not only for scale changes in grey levels but also for changes of orientation. This is particularly efficient for example for a fingerprint where the whole image is composed of black and white stripes which have different orientations. be reconstructed using a very small pattern of stripes.

3.3 Texture synthesis.

This method can be used to create images of the same texture as the original image. The idea is to imitate the distribution of the quanta in the original image. The original image is described by means of the quanta or quantizer and the set of indexes and coefficients. To create an image we have to simulate a model of repartition of the indices and a model of distribution of the coefficients. This can be done by a predictive method using the transition probabilities between two indexes and coefficients. It is better to find a distribution for blocks which overlap in the image. When we generate a distribution of indexes of blocks we find the coefficients of scaling by maximization of the correlation of the overlapping parts of the previous computed block and the current quanta. The value at a pixel is then the average value of all the computed blocks that pass by it. This was implemented in one dimension in [1] with encouraging results.

4 Results.

We give examples of image coding below.

τ_1 represents the part of the data to quantize the image. This is the first term of the expression of τ of the previous section. If we keep the same codebook for a set of images, this is the real rate to look at. 2^b is the number of quanta or the size of the quantizer.

To be interesting from the point of view of time computation and also to have a second term in τ not too large, the value of b should be less than 8.

For the first term to be small enough and ensure a minimum compression, the size of blocks should be at least $n_1 n_2 = 4 \times 4$ since c is about 10 to 16 bits to encode two real numbers.

Figure	Method	block size	2^b	τ	τ_1
1	original	1	256	8	8
2	VQ	4×4	256	1	0.5
3	PVQ	6×6	256	1.2	0.6
4	QPVQ	5×5	16×16	1	1
5	QPVQ	6×6	16×16	0.6	0.6
6left	PVQ	8×8	256	2.3	0.37
6right	VQ	4×4	256	1	0.5
7	QPVQ	6×6	20×20	0.75	0.7

Where VQ means classic Vector Quantization (section 2.1), PVQ means VQ with projection (section 2.3), QPVQ means VQ with projection and a quantizer (section 2.4).

Figures 6 and 7 give an example of texture detection. The texture is extracted in the quantizer on the right of figure 7.

Acknowledgements

Most of this work was performed while the author was at SPAR. The author wishes to thank Andrew Witkin for suggesting this research, and André Gagalowicz for useful comments on a draft of this paper.

REFERENCES

- [1] L.D. Cohen, "Applications de la Quantification Vectorielle à la detection de texture". Research Report Schlumberger Palo Alto Research-Université Paris VI.1985.
- [2] R.C. Gonzales, P. Wintz, "Digital Image Processing". Addison-Wesley, 1977.
- [3] R.M. Gray, "Vector Quantization". IEEE ASSP Magazine, April 1984.
- [4] A.K. Jain, P.M. Farelle, V.R. Algazi, "Image Data Compression". Digital Image Processing Techniques.
- [5] S.G. Mallat, "A Theory for Multiresolution Signal Decomposition : The Wavelet Representation". Tech. Rep. MS-CIS-87-22, U. of Penn., May 1987.
- [6] A. Rosenfeld, A.C. Kak, "Digital Image Processing". Academic Press. 1976.
- [7] J.-C. Simon, "La reconnaissance des formes par algorithmes," Masson, Paris, 1984.
- [8] A. Tassy, L. Miclet. "Quantification vectorielle et reconnaissance de mots multilocuteur".
- [9] A. Witkin, "Scale-space Filtering". Image Understanding 1984, Ed. S.Ullman, W.Richards.

Appendix

4.1 Best quantum for a group of vectors.

Suppose a group is formed of X_1, \dots, X_n and we want to find Q which minimizes the global error :

$$e(Q) = \sum_{i=1}^n d(X_i, Q).$$

4.1.1 classical case.

In the classical case, $d(X, Y) = \|X - Y\|^2 = (X - Y, X - Y)$ Since we know that there is a minimum, we have at this minimum $de(Q) = 0$, which means :

$$de(Q)R = 2\left(\sum_{i=1}^n (X_i - Q), R\right), \text{ so}$$

$$\sum_{i=1}^n X_i = nQ$$

So the best quantum is the mean of the vectors G .

We can also see that :

$$e(Q) = \sum_{i=1}^n \|X_i - G\|^2 + n\|G - Q\|^2$$

which is minimal only for $Q = G$.

4.1.2 “projection” case.

We have now $d(X, Q) = \text{Min}_{\alpha \in R} \|X - \alpha Q\|^2 = \|X - \frac{(X, Q)}{(Q, Q)}Q\|^2$. By definition of d we have $e(\alpha Q) = e(Q)$ so all the values are taken on the unit sphere which is compact and the minimum of e is achieved at a point Q on the sphere such that $de(Q) = 2\lambda Q$ by the theorem of Lagrange multipliers. We have thus :

$$e(Q) = \sum_{i=1}^n (X_i, X_i) - \sum_{i=1}^n (X_i, Q)^2, \quad (Q, Q) = 1$$

$$de(Q) = -2 \sum_{i=1}^n (X_i, Q)X_i = 2\lambda Q$$

Taking the scalar product with Q we have :

$$\lambda = - \sum_{i=1}^n (X_i, Q)^2 = e(Q) - \sum_{i=1}^n (X_i, X_i)$$

The last term being a constant we see that the minimum is achieved when $-\lambda$ is maximum. If we set

$$AQ = \sum_{i=1}^n (X_i, Q)X_i$$

then A is a linear symmetric operator and the best quanta is obtained for the direction of the greatest eigenvalue. We see that for the direction of the best quanta, the sum of projections of Q on X_i 's is in the same direction. We have $e(Q) = \sum_{i=1}^n (X_i, X_i) - (AQ, Q)$. We show that if Q' is the old quanta of the same group, the direction of steepest gradient at Q' is a linear combination of Q' and AQ' . So the simple estimation we take is a direction between Q' and AQ' .

Suppose that the X_i can be written in the canonical base :

$$X_i = \sum_{j=1}^p X_{ij} e_j \quad 1 \leq n$$

where p is the dimension of the space in which the X_i 's are. If now we set $Y_j = (x_{1j}, \dots, x_{nj})$, $1 \leq p$, then the matrix of A is ${}^t M M$ where $M = [Y_1, \dots, Y_p]$. That is A is the Gram matrix of M , the matrix which columns are the Y_i .

Images

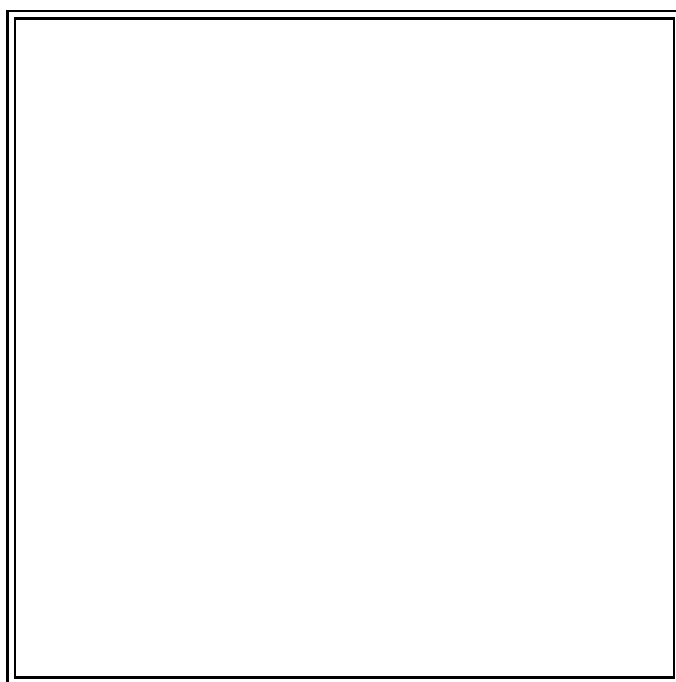


Figure 1 : Original Image

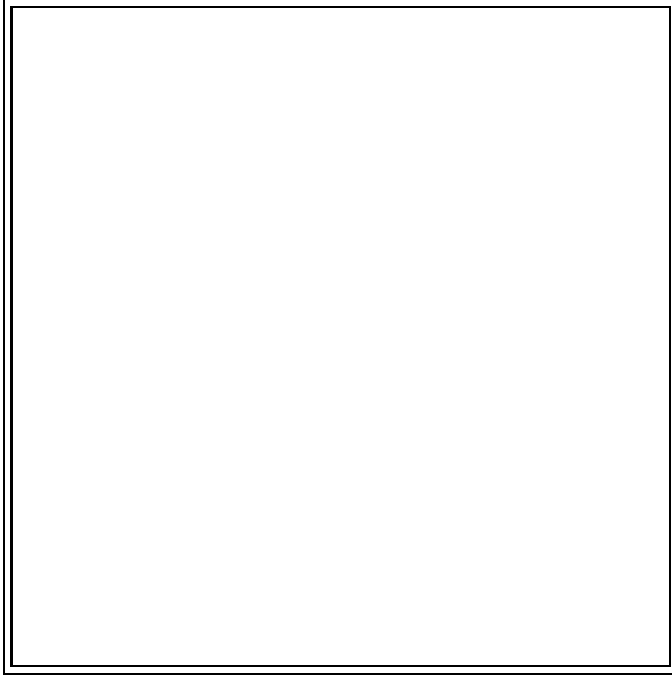


Figure 2 : Classic VQ with 4 by 4 blocks on 8 bits.

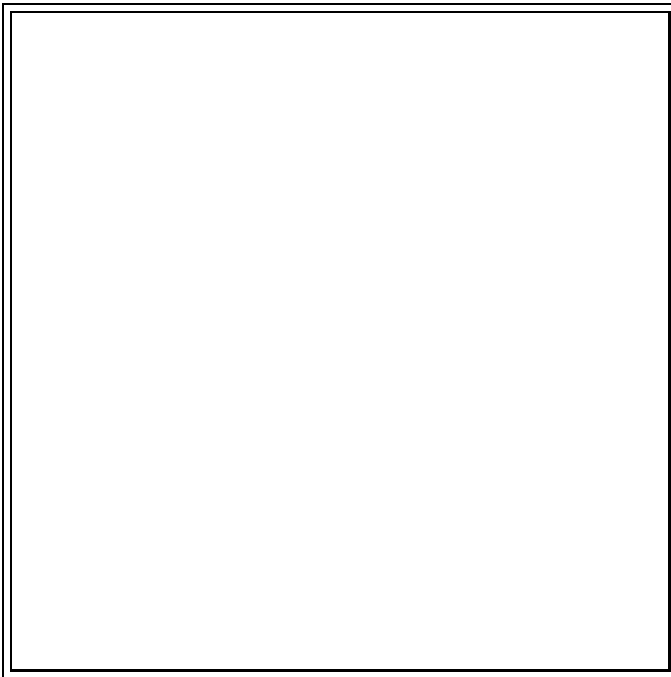


Figure 3: VQ with projection. 6 by 6 blocks on 8 bits.

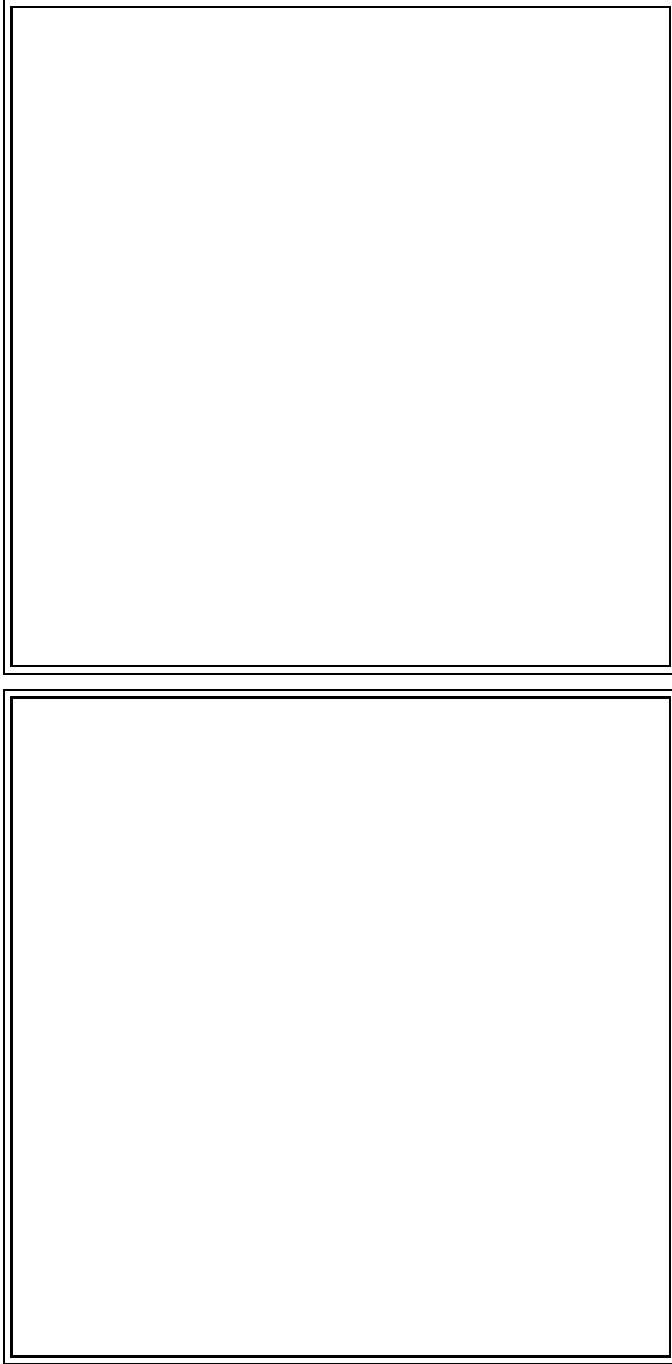


Figure 4 : VQ with projection with the quantizer (section 2.4). 5 by 5 blocks.
The quantizer is 16 by 16.

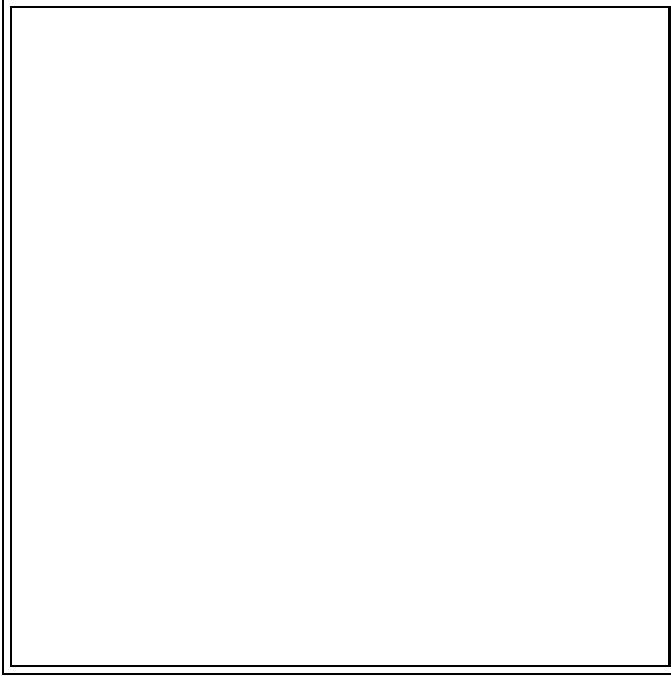


Figure 5 : VQ with projection with the quantizer (section 2.4). 6 by 6 blocks. The quantizer is 16 by 16.

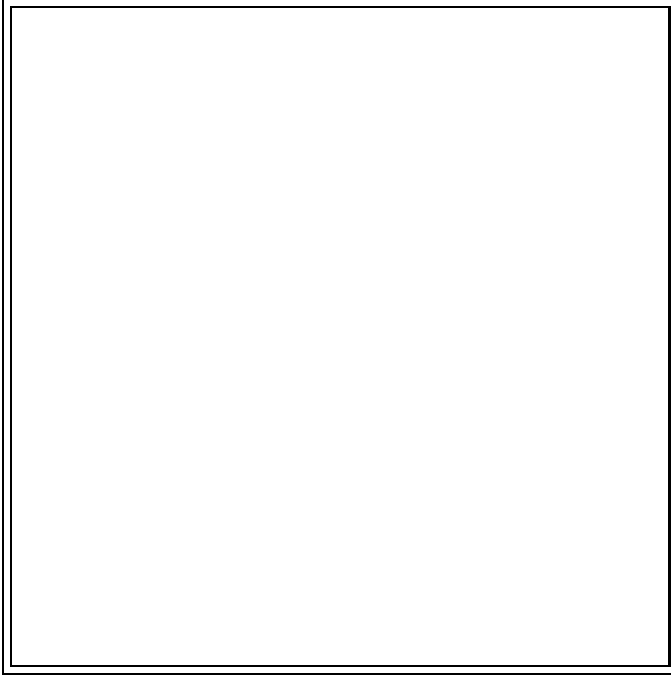


Figure 6 : Texture. The original is lower left.
above is the quantized image with projection with 8 by 8 blocks
on 8 bits.
on the right classic VQ with 4 by 4 blocks on 8 bits

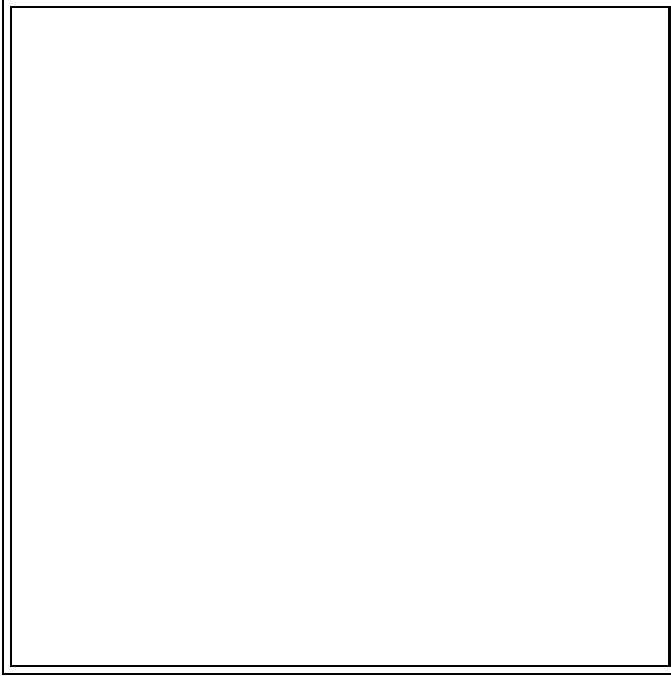


Figure 7 : Texture. The original is lower left.
above is the quantized image with projection and quantizer with
6 by 6 blocks. The quantizer is 20 by 20.
The quantizer is shown on the right