Image Compression by Vector Quantization with Recurrent Discrete Networks

Domingo López-Rodríguez[1], Enrique Mérida-Casermeiro[1],
Juan M. Ortiz-de-Lazcano-Lobato[2], and Ezequiel López-rubio[2]

[1] Department of Applied Mathematics
University of Málaga
Málaga, Spain
{dlopez, merida}@ctima.uma.es
[2] Department of Computer Science and Artificial Intelligence
University of Málaga
Málaga, Spain
{jmortiz, ezeqlr}@lcc.uma.es

Abstract. In this work we propose a recurrent multivalued network, generalizing Hopfield's model, which can be interpreted as a vector quantifier. We explain the model and establish a relation between vector quantization and sum-of-squares clustering. To test the efficiency of this model as vector quantifier, we apply this new technique to image compression. Two well-known images are used as benchmark, allowing us to compare our model to standard competitive learning. In our simulations, our new technique clearly outperforms the classical algorithm for vector quantization, achieving not only a better distortion rate, but even reducing drastically the computational time.

1 Introduction

Compressing an image is a significantly different task than compressing raw binary data. Although general purpose compression techniques can be used to compress images, the result is less than optimal. The reason is that images have certain statistical properties which in turn may be exploited by encoders specifically designed for this task. Also, some of the finer details in the image can be sacrificed for the sake of saving a little more bandwidth or storage space. This fact also means that lossy compression techniques can be used in this area.

Lossless compression involves with compressing data which, when decompressed, will be an exact replica of the original data. Lossless compression is applied to binary data as executables or documents, which need to be exactly reproduced when decompressed. On the other hand, images need not to be reproduced exactly in their original form, but an approximation of the original image is enough for most purposes, as long as the error, obtained in the compression phase, between the original and the reproduced image is tolerable.

S. Kollias et al. (Eds.): ICANN 2006, Part II, LNCS 4132, pp. 595–605, 2006.

Some error measures, commonly used in image compression, are:

- The mean square error (MSE), given by:

$$MSE = \frac{1}{MN} \sum_{i=1}^{M} \sum_{j=1}^{N} [I(i,j) - I'(i,j)]^2$$

  where I is the original image, I' is the approximated version (which is actually the decompressed image) and M, N are the dimensions of the images. A lower value for MSE means lesser error.
- The Peak Signal to Noise Ratio (PSNR), given by:

$$PSNR = 20 \log_{10} \left(\frac{255}{\sqrt{MSE}} \right)$$

  which achieves high value when MSE is low. So a good technique will obtain a high value for PSNR.
- The mean distortion, used in Vector Quantization (VQ), which will be defined in the next section.

We can use any of these three error measures to quantify the goodness of a compression technique. In the present work, we use the mean distortion measure, since it is more appropriate when dealing with VQ.

According to Egmont-Petersen et al. [5], two different types of image compression approaches with neural networks (ANNs) can be identified: direct pixel-based encoding-decoding by one ANN [2,7,16,17] and pixel-based encoding-decoding based on a modular approach [3,4,12,18,20,21]. Different types of ANNs have been trained to perform image compression: feed-forward networks [3,4,16,17,18,20,21], Kohonen Self-Organizing Maps (SOMs) [2,7], adaptive fuzzy leader clustering (a fuzzy ART-like approach) [12], a learning vector quantifier [21] and a radial basis function network [16].

Other approaches are based on competitive neural networks. The aim of competitive neural networks is to cluster the input vectors and it can be used for data coding and compression through vector quantization. It has been shown that competitive learning is an appropriate algorithm for VQ of unlabeled data. Ahalt, Krishnamurthy and Chen [1] discussed the application of competitive learning neural networks to VQ and developed a new training algorithm for designing VQ codebooks which yields near-optimal results and can be used to develop adaptive vector quantifiers. Yair, Zeger and Gersho [22] have proposed a deterministic VQ design algorithm, called the soft competition scheme, which updates all the codevectors simultaneously with a step size that is proportional to its probability of winning. In [15], Pal, Bezdek and Tsao proposed a generalization of learning VQ for clustering which avoids the necessity of defining an update neighbourhood scheme and the final centroids do not seem sensitive to initialization. Ueda and Nakano presented a new competitive learning algorithm with a selection mechanism based on the equidistortion principle for designing optimal vector quantizers [19]. The selection mechanism enables the system to escape from local minima.

Recently, Muñoz-Perez et al. [13] proposed an expansive and competitive learning for VQ capable to avoid local minima of the distortion function, and presented some optimality conditions for the set of codewords.

ANN approaches have to compete with well-established compression techniques such as JPEG, which should serve as a reference. The major advantage of ANNs is that their parameters are adaptable, which may give better compression rates when trained on specific image material. However, such a specialization becomes a drawback when novel types of images have to be compressed.

In this work, we propose a vector quantization approach to image compression by means of a discrete recurrent model, comparing its efficiency to that of the classical competitive learning.

2 Vector Quantization and Competitive Learning

A vector quantifier of dimension d and size K is a mapping Q from the d-dimensional Euclidean space $\mathbb{R}^d$ into a finite subset $C = \{\boldsymbol{c_1}, \ldots, \boldsymbol{c_K}\}$ of $\mathbb{R}^d$ containing K output or representative vectors, called code vectors, reference vectors, reproduction vectors, prototypes or codewords. The collection of all possible reproduction vectors is called the reproduction alphabet or more commonly the codebook. Hence, the input vector space, $\mathbb{R}^d$, is divided into K disjoint regions, $C_1, \ldots, C_K$, where

$$C_k = \{\boldsymbol{x} \in \mathbb{R}^d : Q(\boldsymbol{x}) = \boldsymbol{c_k}\}$$

All inputs vectors in C_k are approximated by $\boldsymbol{c_k}$. The cost introduced by this approximation is given by a nonnegative distortion measure, usually the Euclidean distance between $\boldsymbol{x}$ and the corresponding $\boldsymbol{c_k} = Q(\boldsymbol{x})$.

For a finite training set, $X = \{\boldsymbol{x_1}, \ldots, \boldsymbol{x_N}\}$, the vector quantization is a combinatorial problem that attempts to represent X (with large information contents) by a reduced set of codewords C. In other words, the goal is to select a set C of codewords such that the mean distortion function:

$$D(C) = \frac{1}{N} \sum_{k=1}^{K} \sum_{i|\boldsymbol{x_i} \in C_k} ||\boldsymbol{x_i} - \boldsymbol{c_k}||^2 \tag{1}$$

is minimum. This distortion function is generally not convex.

The standard competitive learning algorithm is a stochastic gradient descent approach to minimize this function. It consists in:

1. Selecting a point $\boldsymbol{x} \in X$ and determining $\boldsymbol{c_k} = Q(\boldsymbol{x})$.
2. Updating $\boldsymbol{c_k}$ with the rule $\Delta\boldsymbol{c_k} = \alpha_n(\boldsymbol{x} - \boldsymbol{c_k})$, where α_n is the learning rate at the n-th training epoch.
3. Repeat the previous points until a maximum of training epochs is reached or convergence is detected.

With this algorithm, it is guaranteed that $\boldsymbol{c_k}$ is the centroid of C_k, and it is the best representative vector of C_k.

3 The MREM Model

Let us consider a recurrent neural network formed by N neurons, where the state of each neuron $i = 1, \ldots, N$ is defined by its output s_i taking values in any finite set $\mathcal{M} = \{m_1, m_2, \ldots, m_L\}$. This set does not need to be numerical.

The state of the network, at time t, is given by a N-dimensional vector, $\boldsymbol{S}(t) = (s_1(t), s_2(t), \ldots, s_N(t)) \in \mathcal{M}^N$. Associated to every state vector, an energy function, is defined:

$$E(\boldsymbol{S}) = -\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} w_{ij} f(s_i, s_j) + \sum_{i=1}^{N} \theta_i(s_i) \quad (2)$$

where $w_{i,j}$ is the weight of the connection from the j-th neuron to the i-th neuron, $f : \mathcal{M} \times \mathcal{M} \to \mathbb{R}$ can be considered as a measure of similarity between the outputs of two neurons, usually verifying the conditions mentioned in [9]:

1. For all $x \in \mathcal{M}$, $f(x, x) = c \in \mathbb{R}$.
2. f is a symmetric function: for every $x, y \in \mathcal{M}$, $f(x, y) = f(y, x)$.
3. If $x \neq y$, then $f(x, y) \leq c$.

and $\theta_i : \mathcal{M} \to \mathbb{R}$ are the threshold functions. Since thresholds will not be used for image compression, therefore we will consider θ_i to be the zero function for all $i = 1, \ldots, N$.

The introduction of this similarity function provides, to the network, of a wide range of possibilities to represent different problems [9,10]. So, it leads to a better and richer (giving more information) representation of problems than other multivalued models, as SOAR and MAREN [6,14], since in those models most of the information enclosed in the multivalued representation is lost by the use of the signum function that only produces values in $\{-1, 0, 1\}$.

If function $f(x, y) = 2\delta_{x,y} - 1$, which equals 1 if and only if its two parameters coincide, and -1 in the rest of cases, is used and $\mathcal{M} = \{-1, 1\}$, MREM reduces to Hopfield's bipolar model (BH) [8]. So, MREM is a powerful generalization of BH and other multivalued models, because it is capable of representing the information more accurately than those models.

The energy function characterizes the dynamics of the net, as happened in BH. In every instant, the net evolves to reach a state of lower energy than the current one.

In this work, we have considered discrete time and semi-parallel dynamics, where only one neuron is updated at time t. The next state of the net will be the one that achieves the greatest descent of the energy function by changing only one neuron output.

Let us consider a total order in $\mathcal{M}$. The potential increment when p-th neuron changes its output from s_p to $l \in \mathcal{M}$ at time t, is

$$U_p(l) = -\Delta E = \frac{1}{2}\sum_{i=1}^{N} [w_{p,i} f(l, s_i(t)) + w_{i,p} f(s_i(t), l) -$$

$$-(w_{p,i}f(s_p(t),s_i(t))+w_{i,p}f(s_i(t),s_p(t)))]-\frac{1}{2}w_{p,p}[f(l,l)-f(s_p(t),s_p(t))] \quad (3)$$

If f verifies the similarity conditions and if matrix W is symmetric and $w_{p,p}=0$ (as in the case of the problem studied in this paper, it will be made clearer in the next section), then the *reduced potential increment* is obtained:

$$U_p^*(l)=\frac{1}{2}\sum_{j=1}^{N}w_{p,j}\left[f(s_p,s_j)-f(l,s_j)\right] \quad (4)$$

We use the following updating rule for the neuron outputs:

$$s_p(t+1)=\begin{cases} l, & \text{if } U_a(l)\geq U_q(k)\,\forall k\in\mathcal{M} \text{ and } \forall q\in\{1,\ldots,N\} \\ s_p(t), & \text{otherwise}\end{cases} \quad (5)$$

This means that each neuron computes in parallel the value of a L-dimensional vector of potentials, related to the energy decrement produced if the neuron state is changed. The only neuron changing its current state is the one producing the maximum decrease of energy.

It has been proved that the MREM model with this dynamics always converges to a minimal state [9]. This result is particularly important when dealing with combinatorial optimization problems, where the application of MREM has been very fruitful [9,10].

4 Two-Stage Image Compression with MREM

In this section we will describe the two-stage VQ algorithm that uses the multivalued model MREM in its first phase.

4.1 Clustering with MREM

In the first stage, MREM is used to obtain a good clustering of the input pattern set.

In order to apply MREM at this step, this clustering problem must be formulated as an optimization task.

Although there are lots of possible formulations for this clustering problem, one of the most used formulations consists in minimizing the sum of intra-cluster distances, that is, if $X=\{\boldsymbol{x_1},\ldots,\boldsymbol{x_N}\}$ is the pattern set to be clustered into K groups, we look forward to minimizing the quantity:

$$d=\sum_{k=1}^{K}\sum_{i|\boldsymbol{x_i}\in C_k}\sum_{j|\boldsymbol{x_j}\in C_k}||\boldsymbol{x_i}-\boldsymbol{x_j}||$$

which is the sum of the distances between patterns in the same cluster. With this formulation, we will obtain homogeneous clusters.

The above expression can be easily re-written as

$$d = \sum_{i=1}^{N}\sum_{j=1}^{N} ||\boldsymbol{x_i} - \boldsymbol{x_j}||\rho_{\boldsymbol{x_i},\boldsymbol{x_j}} \tag{6}$$

where $\rho_{\boldsymbol{x},\boldsymbol{y}}$ equals 1 if and only if $\boldsymbol{x}$ and $\boldsymbol{y}$ belong to the same cluster, otherwise it will be 0.

This new expression can be used to an energy function for the MREM model.

Thus, let us consider a neural network with N neurons. The output s_i of the i-th neuron belongs to the set $\mathcal{M} = \{1, \ldots, K\}$, meaning that the pattern $\boldsymbol{x_i}$ is assigned to the s_i-th cluster.

If we compare Eq. (2) and Eq. (6), and taking into account that θ_i is the zero function for all i, we can obtain the value for the synaptic weights $w_{i,j}$ and an appropriate definition of the similarity function.

This comparison leads us to define:

$$w_{i,j} = -2||\boldsymbol{x_i} - \boldsymbol{x_j}||$$

and

$$f(a,b) = \delta_{a,b} = \begin{cases} 1, \text{ if } a = b \\ 0, \text{ otherwise} \end{cases}$$

So, the energy function will be as follows:

$$E = \sum_{i=1}^{N}\sum_{j=1}^{N} ||\boldsymbol{x_i} - \boldsymbol{x_j}||\delta_{s_i,s_j} \tag{7}$$

that is, the sum of intra-cluster distances.

In order to minimize this energy function, we propose semi-parallel dynamics for the network, as mentioned before:

- In parallel, each neuron computes a vector of reduced potential increments, $\boldsymbol{V_p} = (U_p^*(1), \ldots, U_p^*(K))$, by using Eq. (4), which in this case is

$$U_p^*(l) = \frac{1}{2}\sum_{j=1}^{N} ||\boldsymbol{x_p} - \boldsymbol{x_j}|| \left[\delta_{s_p,s_j} - \delta_{l,s_j}\right]$$

- Each neuron computes in parallel the maximum potential in its corresponding $\boldsymbol{V_p}$. It will be stored in $v_p = \max(\boldsymbol{V_p})$ and n_p will be the value of $l \in \{1, \ldots, K\}$ which produces the maximum potential increment in $\boldsymbol{V_p}$.
- The scheduling selects the neuron q for which $v_q \geq v_p$ for all $p \in \{1, \ldots, N\}$, and updates its output according to $s_q = n_q$. This last step is not made in parallel.

With this dynamics, the energy function is minimized and therefore a clustering of the input pattern space is obtained.

4.2 Computation of the Codebook

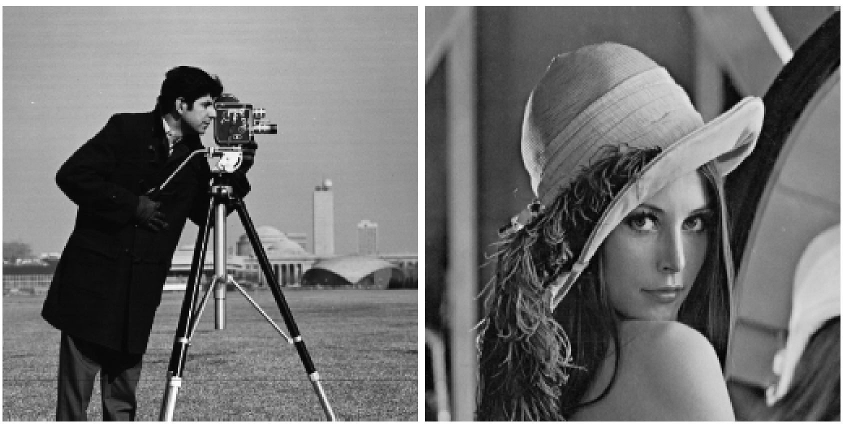
In this second stage, we use the recently obtained clustering to compute the set of codewords.

As we want the mean distortion, given by Eq. (1), to be minimized, we compute $\boldsymbol{c_k}$ as the centroid of the k-th cluster C_k, that is,

$$\boldsymbol{c_k} = \frac{1}{N_k} \sum_{i|\boldsymbol{x_i} \in C_k} \boldsymbol{x_i}$$

where N_k is the number of patterns that belong to C_k.

So, we have guaranteed the (local) optimality of the codebook.



(a) (b)

Fig. 1. Test images used in this work: (a) cameraman, (b) lenna

5 Experimental Results

Two well-known images have been used in this work to show the efficiency of the proposed technique: cameraman and lenna, see Fig 1.

The size of these images was 256x256 pixels, with 256 graylevels. Each image was divided into windows of size $L \in \{8, 10, 12, 16\}$, obtaining a total of $\frac{256^2}{L^2}$ windows. Every window is represented by a L^2-dimensional vector.

Every component of these vectors is normalized to avoid the negative effect of a bad scaling.

This set of vectors is then clustered to obtain $K \in \{16, 32\}$ prototypes and the mean distortion is measured. The results of mean distortion achieved in these experiments are shown in Tables 1 and 2.

In these Tables, a comparison with Standard Competitive Learning (SCL) is made. The learning rate α_n of SCL decreased from 0.9 to 0.01 for 100 training epochs, and 10 executions were performed for each image and algorithm. Columns labeled Min. and Av. show the minimum and average mean distortion

Table 1. Mean distorsion for cameraman image

			MREM			SCL			
L	K	N	Min.	Av.	t	Min.	Av.	t	Impr.
16	16	256	5.31	5.35	0.1946	14.65	14.75	16.4758	175.7%
16	32	256	4.68	4.78	0.3860	14.61	14.69	29.9586	207.3%
12	16	441	3.52	3.56	0.4532	10.79	10.85	20.2672	204.7%
12	32	441	3.19	3.21	0.8867	10.66	10.76	39.5268	235.2%
10	16	625	2.76	2.81	1.1248	8.80	8.88	27.2470	216.0%
10	32	625	2.47	2.50	1.9042	8.84	8.91	50.5668	256.4%
8	16	1024	2.07	2.08	3.7546	6.94	7.02	47.9990	237.5%
8	32	1024	1.86	1.88	7.7969	6.89	6.94	64.9956	269.1%

Table 2. Mean distorsion for lenna image

			MREM			SCL			
L	K	N	Min.	Av.	t	Min.	Av.	t	Impr.
16	16	256	7.12	7.19	0.1725	16.37	16.50	20.2954	129.4%
16	32	256	6.33	6.37	0.3377	16.28	16.41	35.6092	157.6%
12	16	441	4.69	4.72	0.4157	12.15	12.23	22.2540	159.1%
12	32	441	4.12	4.14	0.8644	12.08	12.13	45.0909	192.9%
10	16	625	3.72	3.75	0.8182	9.84	9.99	27.6838	166.4%
10	32	625	3.24	3.26	1.7484	9.86	9.94	46.1765	204.9%
8	16	1024	2.66	2.67	3.2199	7.75	7.82	46.6706	192.8%
8	32	1024	2.32	2.34	7.9586	7.75	7.78	76.5742	232.4%

achieved by the two algorithms. Columns labeled t contain the time spent by each of them. In the last column, Impr., a measure of the improvement achieved by MREM over SCL:

$$\text{Impr.} = \frac{\text{Av}_{SCL} - \text{Av}_{MREM}}{\text{Av}_{MREM}} \cdot 100$$

It is remarkable that MREM highly outperforms SCL on average quality in all cases, achieving improvements of about 150-200%. The time spent by MREM is also a fraction of the spent by SCL. So, MREM is much more efficient than SCL.

In order to show the efficiency of this technique, we have made a simulation in which $L = 4$. If $K = 32$ representatives are used, and $L = 4$, then 128 bits are needed to represent each window, but only 5 to represent the codewords, so we may obtain a compression rate of 128 to 5, that is, 25 to 1 approximately. By using JPG compression, we obtained 45Kb for the original cameraman image, 34Kb for the SCL-compressed and 29Kb for the MREM-compressed. For lenna image, these quantities were 43, 32 and 35Kb, respectively. In Fig. 2, the compressed images, obtained by both techniques, are shown.

(a) (b)

Fig. 2. Compressed test images with $L = 4$ and $K = 32$: (a) by using Standard Competitive Learning (distortions=3.18 and 3.56, from up to down) and (b) by using MREM (distortions=0.67 and 0.81, respectively)

6 Conclusions

In this work we have proposed an alternative method to competitive learning in vector quantization tasks.

This approach is based on a multivalued recurrent network suitable for combinatorial optimization problems, as proved in other works. The intrinsic semi-parallelism provided by this model improves the efficiency of the net when compared to SCL, since the time consumption is drastically reduced. We have applied this approach to image compression, achieving great advantages over SCL, not only on computational time, but even on quality of the quantization, obtaining improvements above 100%. One of the reasons for this improvement is that our algorithm divides the entire task of vector quantization into a two-stage problem: first, it finds a (locally) optimal clustering of the input pattern space, and then it computes the optimal codebook associated to the given partition. Our future work in this problem consists in finding new formulations to help MREM avoid local minima in the clustering task, which will lead to an improvement of the quantization results.

References

1. S.C. Ahalt, A.K. Krishnamurthy, P. Chen, and D.E. Melton, *Competitive learning algorithms for vector quantization*, Neural Networks, **3**, 277-290, 1990.
2. C. Amerijckx, M. Verleysen, P. Thissen et al., *Image compression by self-organized Kohonen map*, IEEE Trans. Neural Networks **9(3)**, 503–507, 1998.
3. J.G. Daugman, *Complete discrete 2-D Gabor transforms by neural networks for image analysis and compression*, IEEE Trans. Acoustics, Speech Signal Process. **36(7)**, 1169-1179, 1988.
4. R.D. Dony, S. Haykin, *Optimally adaptive transform coding*, IEEE Trans. Image Process. **4(10)**, 1358-1370, 1995.
5. M. Egmont-Petersen, D. de Ridder and H. Handels, *Image processing with neural networksa review*, Pattern Recognition **35**, 2279-2301, 2002.
6. M. H. Erdem and Y. Ozturk, *A New family of Multivalued Networks*, Neural Networks **9,6**, 979–989, 1996.
7. G. Hauske, *A self organizing map approach to image quality*, Biosystems **40(1-2)**, 93-102, 1997.
8. J.J. Hopfield, *Neural networks and physical systems with emergent collective computational abilities*, Proc. of National Academy of Sciences USA, **79**, 2254–2558, 1982.
9. E. Mérida Casermeiro, *Red Neuronal recurrente multivaluada para el reconocimiento de patrones y la optimización combinatoria*, Ph. D. dissertation (in Spanish). Univ. Málaga, España, 2000.
10. E. Mérida Casermeiro, J. Muñoz Pérez and R. Benítez Rochel, *A recurrent multivalued neural network for the N-queens problem*, Lecture Notes in Computer Science **2084**, 522–529, 2001.
11. E. Mérida-Casermeiro and D. López-Rodríguez, *Multivalued Neural Network for Graph MaxCut Problem*, Lecture Series on Computer and Computational Sciences, **1**, 375–378, 2004.
12. S. Mitra and S.Y. Yang, *High fidelity adaptive vector quantization at very low bit rates for progressive transmission of radiographic images*, J. Electron. Imaging **8(1)**, 23-35, 1999.
13. J. Muñoz-Perez, J.A. Gomez-Ruiz, E. Lopez-Rubio and M.A. Garcia-Bernal, *Expansive and Competitive Learning for Vector Quantization*, Neural Processing Letters **15**, 261-273, 2002.
14. Y. Ozturk and H. Abut, *System of associative relationships (SOAR)*, Proceedings of ASILOMAR, 1997.
15. N.R. Pal, J.C. Bezdek, and E.C. Tsao, *Generalized clustering networks and Kohonens self-organizing scheme*, IEEE Trans. Neural Networks, **4(4)**, 549-557, 1993.
16. S.A. Rizvi, L.C. Wang and N.M. Nasrabadi, *Nonlinear vector prediction using feed-forward neural networks*, IEEE Trans. Image Process. **6(10)**, 1431-1436, 1997.
17. W. Skarbek and A. Cichocki, *Robust image association by recurrent neural subnetworks*, Neural Process. Lett. **3**, 131-138, 1996.
18. D. Tzovaras and M.G. Strintzis, *Use of nonlinear principal component analysis and vector quantization for image coding*, IEEE Trans. Image Process. **7(8)**, 1218-1223, 1998.

19. N. Ueda, and R. Nakano, *A new competitive learning approach based on an equidistortion principle for designing optimal vector quantizers*, Neural Networks, **7(8)**, 1211-1227, 1994.
20. L.C. Wang, S.A. Rizvi and N.M. Nasrabadi, *A modular neural network vector predictor for predictive image coding*, IEEE Trans. Image Process. **7(8)**, 1198-1217, 1998.
21. A. Weingessel, H. Bischof, K. Hornik et al., *Adaptive combination of PCA and VQ networks*, IEEE Trans. Neural Networks **8(5)**, 1208-1211, 1997.
22. E.K. Yair, K. Zeger and A. Gersho, *Competitive learning and soft competition for vector quantizer design*, IEEE Trans. Signal Processing, **40(2)**, 294-308, 1992.