

# Vector Quantization of Images Using Modified Adaptive Resonance Algorithm for Hierarchical Clustering

Natalija Vlajic and Howard C. Card, *Fellow, IEEE*

**Abstract**—Most neural-network (NN) algorithms used for the purpose of vector quantization (VQ) focus on the mean squared error minimization within the reference- or code-vector space. This feature frequently causes increased entropy of the information contained in the quantizer (NN), leading to a number of disadvantages, including more apparent distortion and more demanding transmission. A modified adaptive resonance theory (ART2) learning algorithm, which we employ in this paper, belongs to the family of NN algorithms whose main goal is the discovery of input data clusters, without considering their actual size. This feature makes the modified ART2 algorithm very convenient for image compression tasks, particularly when dealing with images with large background areas containing few details. Moreover, due to the ability to produce hierarchical quantization (clustering), the modified ART2 algorithm is proven to significantly reduce the computation time required for coding, and therefore enhance the overall compression process. Examples of the results obtained are presented in the paper, suggesting the benefits of using this algorithm for the purpose of VQ, i.e., image compression, over the other NN learning algorithms.

**Index Terms**—Data processing, image compression, unsupervised neural network (NN) learning.

## I. INTRODUCTION

VECTOR quantization (VQ) is known to be a very useful technique in lossy data compression. In general, compression techniques fall into two main classes. *Lossless compression*, although it allows a perfect reconstruction of the original information, has very limited ability to reduce the amount of data. *Lossy compression*, on the other hand, is much more effective, but inevitably leads to some distortion when information is decompressed. From this perspective, the main advantage of VQ over the other lossy compression techniques is that it exploits the structure underlying a given data set and thereby provides a lower level of distortion for a given compression rate, i.e., number of bits per sample, as compared to many other methods. This is especially the case if data is significantly correlated, or, in other words, if data ensembles (groups of data points) tend to fall in clusters.

Manuscript received October 7, 1999; revised July 31, 2000. This work was supported by NSERC and CMC.

N. Vlajic is with Broadband Wireless and Internet, Research Laboratory, School of Information Technology and Engineering, University of Ottawa, Ottawa, ON K1N 6N5, Canada (e-mail: nvlajic@site.uottawa.ca).

H. C. Card is with the Department of Electrical and Computer Engineering, University of Manitoba, Winnipeg, MB R3T 5V6, Canada (e-mail: card@ee.umanitoba.ca).

Publisher Item Identifier S 1045-9227(01)06457-8.

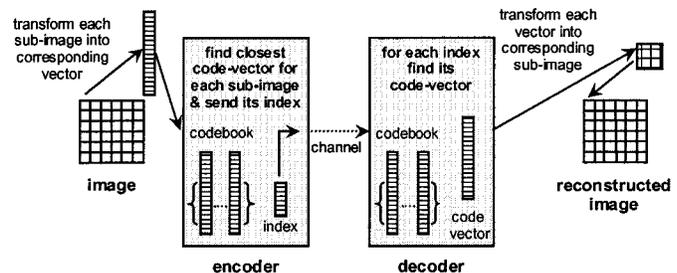


Fig. 1. Vector quantization for image compression.

The design and operation of a VQ system is based on the following principles. First, a limited set of vectors (*code-vectors*), which are assumed to be representatives of all such data ensembles that might be generated from the source, are chosen in order to form the *codebook*. The codebook is known to both the VQ encoder and decoder. In order to be processed by the system, every data stream coming from the source has to be grouped into blocks or vectors of the code-vector dimensionality. For every input data block the nearest code-vector is selected (this process is called the *encoding phase*), and only the index of that code-vector is transmitted through the channel. On the other side of the channel, since the decoder has exactly the same codebook, the original code-vectors can be easily retrieved. This process is called the *decoding phase*, since it enables the reconstruction of the original input data set.

The size of the codebook is usually of the form  $2^N$ , where  $N$  is an integer number. Accordingly,  $N$  represents the number of bits required to encode each code-vector, i.e., its index. Furthermore, if the code- and input vectors are of dimensionality  $M$ , the compression ratio obtained by the system is  $N/M$  [bits/sample]. It is apparent that either by using a codebook with fewer code-vectors or by employing code-vectors, i.e., input data blocks, of larger dimensionality, a better (greater) compression ratio can be achieved. However, the distortion of input data tends to increase as the compression ratio increases, and therefore the values chosen for  $N$  and  $M$  are supposed to provide a certain balance between these features.

Fig. 1 illustrates the VQ procedure for image compression.

The principal problem in VQ is to find a codebook which would minimize the loss of information caused by compression and decompression according to some criteria, i.e., performance measure. The most commonly used performance measures are the mean squared error (mse) and the signal-to-noise

ratio (SNR). Although the mse and SNR can provide an objective analysis of the quality of compression from a technical, or mathematical, point of view, they do not always correspond to the quality standards of human perception. Therefore, the judgment on the performance of a VQ system has to incorporate both objective mathematical and subjective human perception-related measures.

Various techniques have been proposed and used for the purpose of codebook creation, including several neural network (NN) learning algorithms. It can be observed that all these techniques mainly focus on the mse minimization within the codebook vector space, thereby attempting to increase the average information per node, i.e., quantization level, in the sense of Shannon's entropy. In this paper we argue that for datasets, particularly images, uncorrupted with noise and of nonuniform block-vector (subimage) distribution, large Shannon's average information per node, i.e., Shannon's entropy, is not an appropriate goal. Instead, large average information in the sense of arithmetic mean seems to be more related to the quality of decompressed data, and therefore a more suitable goal. Our modified ART2 algorithm is an NN learning algorithm whose primary aim is not the mse minimization within the whole reference vector (code-vector) space, as is the case with some other NN algorithms. Instead it focuses on the discovery of the main clusters while minimizing the mse within each of them. In the following sections we show that the modified ART2, having the ability to produce hierarchical clustering insensitive to nonuniform variations in the input data distribution, offers several advantages over the other NN learning algorithms in terms of vector quantization and coding for image compression.

## II. THEORETICAL BACKGROUND

Let us assume a very simple case, as presented in Fig. 2. The dataset consists of one-dimensional data-points, and these are grouped into two main clusters. The corresponding distribution density, i.e., the probability density function (pdf), is considerably larger within one than within the other cluster. Conceptually, this distribution could be conceived as an analogy to the subimage distribution obtained by subdividing an image of a fairly uniform background with a few important foreground details (see Fig. 8). For a better understanding of the following discussion, it has to be emphasized that for the case of such an image, although the details present a small portion of the overall data and statistically are almost negligible compared to the background, they actually may contain nearly all important information that the image carries.

Now, let us further assume that the dataset presented in Fig. 2 is used as a training set for a two-node NN, which is supposed to operate as a quantizer. This implies that after the training is performed, the NN will be capable of classifying the data into two mutually disjoint groups, such that each group consists only of the data-points placed within the Voronoi region of one of the nodes. In other words, for all data-points of one particular group the same node, i.e., its reference vector, presents the closest quantization level.

In general, classification, or quantization, obtained by an NN is uniquely and exclusively defined by the positions of its nodes,

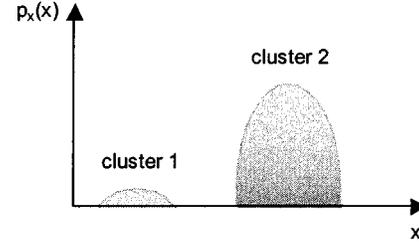


Fig. 2. Probability density function of a given dataset.

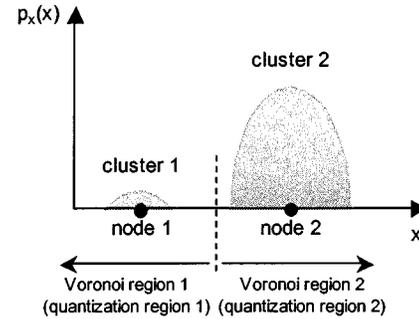


Fig. 3. Quantization focused on the discovery of the main input data clusters.

i.e., their reference vectors. Since the positions of the reference vectors are determined by the nature of the learning algorithm applied, various NN learning algorithms may result in significantly different clusterings for the same dataset.

### A. Quantization Using Algorithms for Input Data Cluster Discovery

Fig. 3 illustrates one possible clustering obtained for the dataset presented in Fig. 2. Theoretically, such clustering could be provided by a neural network algorithm capable of discovering the main input data clusters. In particular, this means that the algorithm, both during the learning and testing phase, allocates the same winning node to sufficiently similar (close) input vectors only, independent of the input data distribution density. It has been proven that adaptive resonance theory (ART) [1], and as a result modified ART (ART2) [2], satisfy these requirements, and therefore should be able to provide clustering as given in Fig. 3.

If we define the probability of a node with the respect to  $p_x(x)$  by

$$p_{\text{node } i} = \int_{\text{Voronoi reg. } i} p_x(x) dx \quad (1)$$

[in general,  $p_x(x)$  annotates the pdf of code-vectors obtained by sampling an image] and accordingly the information content that corresponds to that node by

$$I_{\text{node } i} = \log \frac{1}{p_{\text{node } i}} \quad (2)$$

( $i \in$  set of nodes) then for the case of Fig. 3  $p_{\text{node } 1} < p_{\text{node } 2}$ , and  $I_{\text{node } 1} > I_{\text{node } 2}$ . This implies that an NN algorithm capable of clustering in the manner illustrated in Fig. 3 results in node positions, probabilities, and information content which may accurately indicate the positions, probabilities, and information content of the actual clusters.

For the sake of the following discussion, let us emphasize that due to the properties

$$\bigcup_{i \in \text{set of nodes}} \text{Voronoi reg. } i = R^n \quad (3)$$

and

$$\int_{R^n} p_x(x) = 1 \quad (4)$$

obtained for any clustering, the following equation will hold:

$$\sum_{i \in \text{set of nodes}} p_{\text{node } i} = 1. \quad (5)$$

In general, an NN for vector quantization can be regarded as a *secondary source*, or a *source to the channel*, since the actual outcome from the encoder to the channel (Fig. 1) is simply the set of reference vectors of the nodes. Accordingly, it makes sense to define the entropy of such a source, and for the above case the entropy ( $H(S_1)$ ) would be

$$\begin{aligned} H(S_1) &= p_{\text{node } 1} I_{\text{node } 1} + p_{\text{node } 2} I_{\text{node } 2} \\ &= p_{\text{node } 1} \log_2 \frac{1}{p_{\text{node } 1}} + p_{\text{node } 2} \log_2 \frac{1}{p_{\text{node } 2}}. \end{aligned} \quad (6)$$

Entropy, as given in (6), can be interpreted as the sum of the nodes' self-information, weighted by the corresponding node probability. Accordingly, from a Shannon's entropy point of view, the information content of node 1 (see Fig. 3), when weighted, might appear less important than the information content of node 2. This approach to the estimation of average information coming from a secondary source (of the type that is being discussed in this section) is appropriate for the situations when the original dataset is noisy (such as an image corrupted with random noise). In those cases node 1 might be conceived simply as a representative of a cluster of meaningless data, or in other words data corrupted with noise. (A real noisy image would have many clusters of this sort.) Thus, the "weighting" from equation (6) would have the purpose of suppressing the information contained in "noisy nodes."

However, as has already been mentioned, for images uncorrupted with noise, with large background areas and relatively few important details, the above concept would be incorrect. A more appropriate measure of the average information coming from a secondary source in these cases would be simply the arithmetic mean of node information, as given in (7)

$$I_{\text{am}}(S_1) = \frac{1}{2} \left( \log_2 \frac{1}{p_{\text{node } 1}} + \log_2 \frac{1}{p_{\text{node } 2}} \right). \quad (7)$$

The average information from (7) can be seen to be a special case of (6), assuming that all nodes (clusters) are considered equally important.

### B. Quantization Using Algorithms for Input Data Density Estimation

Most of the currently used NN learning algorithms for vector quantization, including standard (hard) and soft competitive learning, self-organizing feature maps [3], growing and splitting elastic networks [4], neural gas networks [5] and networks

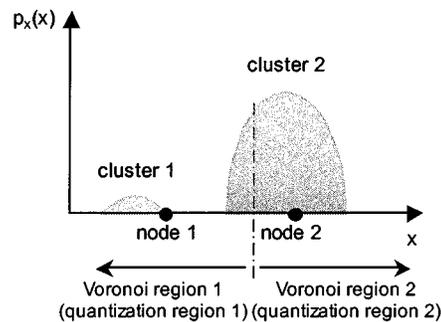


Fig. 4. Clustering influenced by the topology of input data.

based on the minimum description length principle [6] focus on the mse (or expected quantization error) minimization. For the case of a continuous input data distribution  $p_x(x)$ , or for a finite input dataset  $D$ , mses are given in (8), respectively

$$\begin{aligned} \text{EMS} &= \sum_{i \in N} \int_{\text{Voronoi reg. } i} \|x - w_i\|^2 p_x(x) dx \\ &= \frac{1}{|D|} \sum_{i \in \text{set of nodes}} \sum_{x_{ij} \in \text{Voronoi reg. } i} \|x_{ij} - w_i\|^2 \end{aligned} \quad (8)$$

where  $w_i$  is the reference vector of node  $i$ .

It has been shown that an NN algorithm with the mse minimization within the reference vector space as its primary goal is capable of positioning the nodes in the way that they match the input data distribution density [7]. Accordingly, the reference vectors (nodes) tend to be distributed such that they become "winners" with the same probability. In other words, Voronoi regions tend to contain the same number of input data-points. Therefore, for example, if the input data is of a significantly nonuniform distribution, most of the nodes will concentrate in the vicinity of regions with large distribution density. (For more details on this property see [2].)

Fig. 4 illustrates data-point quantization or clustering of the dataset presented in Fig. 2 obtained using an NN algorithm to perform a mse minimization. From the practical point of view, since the initial values of the reference vectors in this type of NN algorithm are usually determined by the statistics of the input data, it can be assumed that the learning procedure begins with both nodes within cluster 2. However, later in the learning, due to being the winner for data-points from cluster 1, one of the nodes (in our case node 1) gradually moves toward the position indicated in Fig. 4. The other node, node 2, remains within cluster 2, since there are no other data points for which it could be the winner. Eventually, the learning terminates with the probability of node 1 with respect to  $p_x(x)$  approaching the probability of node 2 with respect to  $p_x(x)$ .

If, in this case when an NN algorithm for mse minimization is employed,  $p'_{\text{node } 1}$  and  $p'_{\text{node } 2}$  are used to annotate the probabilities of node 1 and 2 with respect to  $p_x(x)$  [according to the definition given in (1)], then from the above discussion it follows that

$$p'_{\text{node } 1} \rightarrow p'_{\text{node } 2} \quad (9)$$

(i.e., the learning algorithm tends to position node 1 in such a way that  $p'_{\text{node } 1}$  and  $p'_{\text{node } 2}$  become as close as possible), and

accordingly, with respect to the clustering concept from Section II-A,

$$p'_{\text{node } 2} = p_{\text{node } 2} - \Delta p \quad (10)$$

$$p'_{\text{node } 1} = p_{\text{node } 1} + \Delta p \quad (11)$$

where  $\Delta p \geq 0$ . [Note: since an NN with only two nodes is assumed, and the respective node probabilities always sum to unity based on (5), then a new clustering policy that decreases the probability of node 1 for  $\Delta p$  inevitably implies an increase of node 2 probability for the same amount.]

Based on the above equations, the entropy ( $H(S_2)$ ) of a secondary source that produces a clustering such as that presented in Fig. 4 is

$$\begin{aligned} H(S_2) &= p'_{\text{node } 1} \log \frac{1}{p'_{\text{node } 1}} + p'_{\text{node } 2} \log \frac{1}{p'_{\text{node } 2}} \\ &= (p_{\text{node } 1} + \Delta p) \log \frac{1}{p_{\text{node } 1} + \Delta p} \\ &\quad + (p_{\text{node } 2} - \Delta p) \log \frac{1}{p_{\text{node } 2} - \Delta p}. \end{aligned} \quad (12)$$

Furthermore, the difference between  $H(S_1)$  (from Section II-A) and  $H(S_2)$  can be expressed by

$$\begin{aligned} H(S_1) - H(S_2) &= p_{\text{node } 1} \log \frac{p_{\text{node } 1} + \Delta p}{p_{\text{node } 1}} + p_{\text{node } 2} \log \frac{p_{\text{node } 2} - \Delta p}{p_{\text{node } 2}} \\ &\quad + \Delta p \log \frac{p_{\text{node } 1} + \Delta p}{p_{\text{node } 2} - \Delta p}. \end{aligned} \quad (13)$$

Now, due to the well-known inequality  $\log x \leq x - 1$ , we have that

$$\begin{aligned} H(S_1) - H(S_2) &\leq p_{\text{node } 1} \left( \frac{p_{\text{node } 1} + \Delta p}{p_{\text{node } 1}} - 1 \right) \\ &\quad + p_{\text{node } 2} \left( \frac{p_{\text{node } 2} - \Delta p}{p_{\text{node } 2}} - 1 \right) \\ &\quad + \Delta p \left( \frac{p_{\text{node } 1} + \Delta p}{p_{\text{node } 2} - \Delta p} - 1 \right) \end{aligned} \quad (14)$$

which further implies

$$\begin{aligned} H(S_1) - H(S_2) &\leq \Delta p \left( \frac{p_{\text{node } 1} + \Delta p}{p_{\text{node } 2} - \Delta p} - 1 \right) \\ &= \Delta p \left( \frac{p'_{\text{node } 1}}{p'_{\text{node } 2}} - 1 \right). \end{aligned} \quad (15)$$

Since, according to (9),  $p'_{\text{node } 1}/p'_{\text{node } 2} \rightarrow 1$ , the important conclusion is that

$$H(S_1) - H(S_2) \leq 0 \Rightarrow H(S_1) \leq H(S_2). \quad (16)$$

In a similar fashion it can be proved that

$$\begin{aligned} Iam(S_2) - Iam(S_1) &= \frac{1}{2} \log \frac{p_{\text{node } 1}}{p_{\text{node } 1} + \Delta p} + \frac{1}{2} \log \frac{p_{\text{node } 2}}{p_{\text{node } 2} - \Delta p}. \end{aligned} \quad (17)$$

Accordingly

$$Iam(S_2) - Iam(S_1) \leq \frac{\Delta p}{2} \left( \frac{1}{p'_{\text{node } 2}} - \frac{1}{p'_{\text{node } 1}} \right) \quad (18)$$

which for  $p'_{\text{node } 1}/p'_{\text{node } 2} \rightarrow 1$  implies

$$Iam(S_2) - Iam(S_1) \leq 0 \quad (19)$$

or simply

$$Iam(S_2) \leq Iam(S_1). \quad (20)$$

[Through more complex procedures (16) and (20) can readily be extended to arbitrary number of nodes, or clusters].

In practical terms (16) implies that the entropy of a secondary source based on an algorithm for the mse minimization will always be greater than or equal to the entropy of a secondary source that simply focuses on input data cluster discovery. However, based on the elaboration from Section II-A, large entropy may be an appropriate goal only when input data is corrupted with noise. In all other cases, such as the images treated in this paper, large entropy may be entirely irrelevant. Moreover, it may be an indicator of inadequate quantization.

In terms of coding, (16) suggests another disadvantage of algorithms for mse minimization compared to algorithms for the discovery of input data clusters. If we recall from information theory [8], a lower bound on the entropy of a code (set of codewords) used to encode the symbols of a source is the source entropy itself, as in (21). (In the case of quantization by an NN, source symbols are the reference vectors, while each codeword is simply a binary symbol allocated to one of the source symbols.)

$$H(C) \geq H(S) \quad (21)$$

where  $H(S)$  is the source entropy, and  $H(C)$  is the corresponding code entropy.

Let us further recall that, for a given code that is allowed to have a variable codeword length, the code-entropy corresponds to the average length of its codewords, given in [bits/symbol] [8]. This implies the following interpretation of (21): for a source of a certain entropy  $H(S)$ , it is theoretically possible to generate a corresponding code of average codeword length as low as  $H(S)$ . Undoubtedly, regarding the transmission, codes of lower average codeword length are more efficient. With respect to the above explanation, (16) shows that a secondary source based on an algorithm for mse minimization tends to be more demanding from the transmission point of view than a secondary source based on an algorithm for the discovery of input data clusters, without necessarily providing better compression.

Finally, (20) is additional proof of a potentially disadvantageous utilization of algorithms for mse minimization. In particular, if we assume that for a given number of nodes the family of algorithms presented in Section II-A is capable of matching the probabilities of input data clusters in an optimal way, then  $Iam_1$  presents the optimal average information per node in the sense of arithmetic mean. However, based on (20), the arithmetic mean of node information that corresponds to the family of algorithms presented in this section will always be less than, or at most equal to, the optimal value.

### III. MODIFIED ART2 VERSUS ART2 FOR VECTOR QUANTIZATION PURPOSES

ART, introduced in 1976 by Grossberg, is an unsupervised learning technique, partially based on the well-known winner-take-all concept, but also influenced by the stability-plasticity dilemma [1]. (For more on winner-take-all concepts see [2] or [9].) There are two main adaptive resonance theory models: ART1 and ART2. While the ART1 model is capable of stably learning binary input patterns, the ART2 models show the same behavior for analog patterns. Since 8-bit gray-scale images consist of pixels that can take any value between 0 and 255, vector quantization for image compression requires the utilization of ART2. (In recent years another interesting form of adaptive resonance theory, the so-called fuzzy ARTMAP model, has gained a wide popularity. Fuzzy ARTMAP achieves a synthesis of fuzzy logic and ART model-based neural networks and is capable of supervised learning. Further references on fuzzy ARTMAP and other innovations in ART NNs can be found at <http://www.wi.leidenuniv.nl/art/>.)

The main difference between the ART2 learning model and the other NN algorithms used for the purpose of VQ (these algorithms are mentioned in Section II-B) is related to the adjustment of the winning node for each new training pattern. While the algorithms from Section II-B seek for the closest (winning) node from the corpus of all currently existing nodes, and unconditionally perform its adjustment, the ART2 model initiates the adjustment only if it deems the winning node to be an acceptable match to the current training pattern. In other words, ART2 modifies the profile of one of the currently recognized categories (clusters) only if the input vector is sufficiently similar (satisfies vigilance criterion) to risk its further refinement. Otherwise, if no available category provides a good enough match, a new node is created for learning a novel recognition category. However, if no adequate match exists and the full capacity of the system has also been exhausted (no further nodes may be introduced), the network cannot accommodate the new input and learning is automatically inhibited. This mechanism defends a fully committed memory capacity against eradication by new significantly different input patterns.

Although ART2 is theoretically capable of discovering the main input data clusters, and therefore could be used for compression purposes in the manner presented in Section II-A, there are several reasons why it cannot be considered as an ideal vector quantization algorithm. The following features are its principal disadvantages. First, ART2 requires a fixed number of output nodes and a predefined vigilance parameter. If the number of clusters that corresponds to the predefined vigilance parameter is greater than the capacity, or maximum number of output nodes, the network is incapable of learning all the categories. Therefore, clusters that appear after the full capacity of the network has been exhausted simply remain unaccommodated or rejected. Second, while the actual value of vigilance parameter ( $\nu$ ) is the main controlling factor in the learning process, there is no clear indication how many categories the neural network will recognize for that particular  $\nu$ . (For the same value of vigilance parameter the network may recognize different num-

bers of categories for different input data distributions.) However, as has been explained in Section I, vector quantization generally assumes codebooks to be of size  $2^N$ , which implies a requirement for same number of nodes in the network. Therefore, it may be very complicated if not impossible, requiring extensive experimentation, to find the exact value of vigilance parameter which would correspond to the required number of nodes for each particular input data distribution.

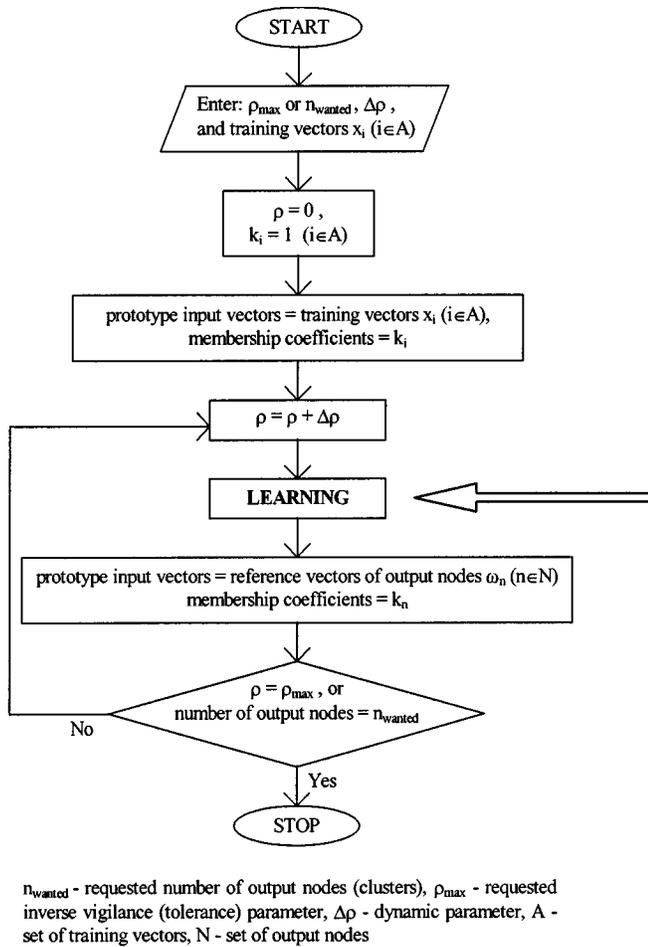
The modified ART2 algorithm we employ is a neural learning algorithm based on the conceptual approach of ART2, while at the same time overcoming the above mentioned problems. The modified ART2 algorithm was originally developed for Web page (hypertext) classification (for more details see [2]), but it is applicable to a variety of clustering tasks.

The outline of our modified version of ART2 is presented in Fig. 5.

The main novelty introduced in the modified ART2, over the standard version, is its gradually increasing *tolerance parameter*  $\rho$ . This parameter is a dynamic generalization of the inverse of the standard (static) vigilance parameter  $\nu$ . (While conceptually the tolerance parameter of modified ART2 is an equivalent to the vigilance parameter of the standard ART2, the new name has been introduced intentionally in order to stress the difference that comes from the dynamic nature of the former.) It has been demonstrated that this feature can ensure several significant properties, particularly from the perspective of vector quantization. First, due to the dynamic nature of the tolerance parameter it becomes feasible to terminate the learning process when the number of output nodes reaches a required value. In other words, the number of output nodes becomes the controlling factor instead of the tolerance (vigilance) parameter. Second, for sufficiently small values of the dynamic parameter ( $\Delta\rho$  in Fig. 5) the resulting clustering is completely stable and hierarchical. (In this particular case the property of being stable and hierarchical implies that any two input vectors that shared the same winning node, and thus belonged to the same cluster, for a lower level of the tolerance parameter will have a mutual winning node for all higher levels of  $\rho$ .) Accordingly, the algorithm produces results that can be represented by a tree structure or dendogram (see Fig. 7). This property makes the modified ART2 particularly convenient for enhanced retrieval of code-vectors at the coder side of a VQ system. In other words, the modified ART2 is capable of improving the speed of coding, thereby providing very rapid compression. (For more on the hierarchical nature of clustering obtained by the modified ART2 see [2].)

#### A. Hierarchical Clustering for Fast VQ Coding

The procedure of finding the closest code-vector for a new pattern that occurs at the input of a VQ coder (Fig. 1) is a typical problem from information retrieval theory [10]. According to this theory, conventional retrieval techniques are mainly based on *serial search*, which means that a given query (new pattern) has to be matched with each prototype-vector in the collection (codebook) in order to find the most similar one. Although *serial search* provides adequate results, it is acknowledged to be extremely slow. In contrast, more sophisticated and efficient re-



**LEARNING:**

**STEP 1:** Insert one output node. Initial its reference vector to the first prototype input vector ( $w_1 = \xi_1$ ), and set its membership coefficient to the corresponding value of  $\xi_1$ .

**STEP 2:** Chose a new prototype input vector  $\xi$ .

**STEP 3:** Determine the best matching output node  $s$  - the unit with the nearest reference vector:

$$\|w_s - \xi\| < \|w_n - \xi\|, \forall n \in N$$

where  $N$  is the set of output nodes.

**STEP 4:** Verify that  $\xi$  belongs to the  $s^{\text{th}}$  cluster (the cluster determined by  $w_s$  and  $\rho$ ) if

$$\|w_s - \xi\| < \rho$$

If so proceed to step 5. Otherwise go to step 6.

**STEP 5:** Adjust the reference vector of  $s$  according to

$$W_s = (k_s / (k_\xi + k_s)) w_s + (k_\xi / (k_\xi + k_s)) \xi$$

Note that  $w_s$  becomes the arithmetic mean of all (old and new) vectors belonging to the  $s^{\text{th}}$  cluster. Adjust the membership coefficient of  $s$ :

$$k_s = k_s + k_\xi$$

Go to step 1.

**STEP 6:** Since  $\xi$  does not belong to  $s$ , which was the most probable, insert a new output node. Set its reference vector to  $\xi$  and its membership coefficient to  $k_\xi$ .

Go to step 2.

Fig. 5. Modified ART2 algorithm.

trieval systems are based on *cluster search strategies*. Instead of conducting a search through the entire collection, these systems first classify prototype vectors into subgroups, then confine the search to certain subgroups only. In other words, they use the following overall strategy:

- ◆ Clusters of sufficiently similar prototype vectors are constructed based on the Euclidean distance measure.
- ◆ Each cluster is represented by a special vector, known as the *cluster centroid*.
- ◆ A given new vector (query) is compared against the centroids of all groups, and only prototype vectors located in the group of the highest query-centroid similarity are considered for further comparison.

Undoubtedly, reducing the number of required comparisons between prototype vectors and queries, cluster search provides enhanced retrieval. The outline of a clustered prototype vector collection is given in Fig. 6. According to Fig. 6, a clustered codebook may have several types of centroids: hypercentroid—which represents the center of the complete collection, supercentroids—which represent the next level of granularity, centroids—which represent regular prototype vector clusters. Within such a hierarchically organized system, a search for the closest code-vector is conducted by comparing

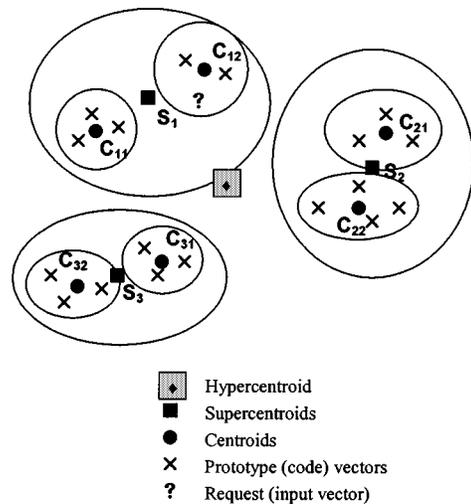


Fig. 6. Clustered codebook.

the query first against the highest-level centroids. Then, only for the higher level centroids that are shown to be the closest to the query at each particular level, the search is continued.

The following figure shows a search tree for the clustered collection of Fig. 6.

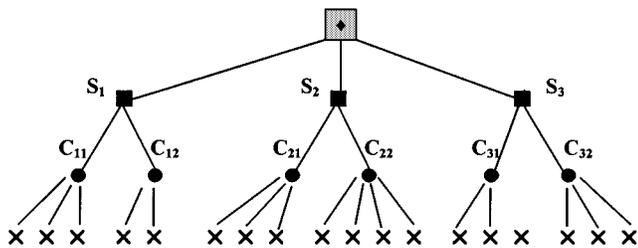


Fig. 7. Search tree for clustered codebook of Fig. 6.

Multilevel cluster search, based on a hierarchical cluster structure as illustrated in Fig. 7, has been shown to provide significant improvements in retrieval efficiency [10]. Accordingly, such a search applied to the retrieval of code-vectors in a VQ system implies very rapid coding. The results in the following section verify that the coding efficiency of a VQ system built upon a hierarchically organized codebook can be improved up to 85%.

#### IV. EXPERIMENTAL RESULTS

The experiments presented in this section were conducted in order to compare the performance of an NN algorithm for the mse minimization [in our case hard competitive learning (HCL)] versus an NN algorithm for the discovery of input data clusters (in particular modified ART2) for the purpose of image compression. One may argue that the algorithms chosen were not the only representatives of their groups. (For example,  $k$ -means [13] is another well-known representative of clustering algorithms for the mse minimization. It can be employed either in “batch” or “on-line” version. While the batch version typically provides for stable but computationally more expensive learning, the on-line version is essentially equivalent to HCL.) However, the purpose of the experiments was not to investigate subtle improvements by variants of the same learning concept, but to explore the principal advantages of one learning concept over the other.

The software simulation of the HCL algorithm was performed using the Matlab Neural-Network Toolbox [11]. In each case the network was trained for 5000 iterations, although it was observed that for most cases learning stabilization was obtained after less than 3000 iterations. The learning rate was 0.1. Some experiments with a larger learning rate failed to provide sensible results, while smaller learning rates did not contribute to better compression.

The simulation of the modified ART2 algorithm was mainly based on our software developed as explained in [2]. (Initially, the software simulation of modified ART2 was just an ingredient of a larger composite autonomous agent. The agent was implemented in Java, and built for the purpose of Web-page classification employing NN. However, due to its universal nature, it was possible to extract the core algorithm from the overall system, and apply it to any VC task.) In all cases the dynamic parameter ( $\Delta\rho$ ) was set to 0.5. By experimenting with a number of different cases this value was initially determined to provide clustering that is perfectly stable and does not depend on the initial ordering of training vectors.



Fig. 8. Image with near uniform background together with a few important textual details (original size:  $256 \times 256$  pixels).

All images used consisted of  $256 \times 256$  pixels. The images were partitioned into subimages of  $8 \times 8$  pixels. Thus, in all cases, input- and code-vectors were 64-dimensional. In both classes of experiments, based on HCL and modified ART2, the ordering of input vectors was randomized prior to their utilization in the learning process. For each image compressed the corresponding mse error and SNR with respect to the original image were calculated, using the following expressions:

$$\text{mse} = \frac{1}{N_1 N_2} \sum_{i=1}^{N_1} \sum_{j=1}^{N_2} (x(i, j) - \bar{x}(i, j))^2 \quad (22)$$

$$\text{SNR} = 10 \log_{10} \frac{\frac{1}{N_1 N_2} \sum_{i=1}^{N_1} \sum_{j=1}^{N_2} x(i, j)^2}{\text{mse}} \quad (23)$$

where  $N_1$  and  $N_2$  are image width and length,  $x(i, j)$  and  $\bar{x}(i, j)$  are the original and reconstructed pixel intensity, respectively.

##### A. Experiment 1

The first experiment is related to the image given in Fig. 8. As can be observed, the image consists of a large, gradually shaded background, with a few important details given in the form of letters (text). (This type of images often can be seen in various Web pages, and therefore their compression may be particularly important from the point of view of Internet communications.)

Figs. 9(a)–(f), and 10(a)–(f) are the versions of Fig. 8 following compression and decompression, produced by the modified ART2 and HCL algorithms respectively, for different number of nodes—quantization levels. (Note: in the remainder of the paper the two terms, *node* and *quantization level*, will be used interchangeably, since there is a clear conceptual “one-to-one” relationship between what these terms represent in the case of VQ employing NNs.) It is observed that the best quality of decompressed data, according to the human perception of image quality, is obtained by the modified ART2 with 128 nodes [Fig. 9(f)].

However, by comparing the overall performance of the two algorithms, it may be observed that in almost all cases (for each

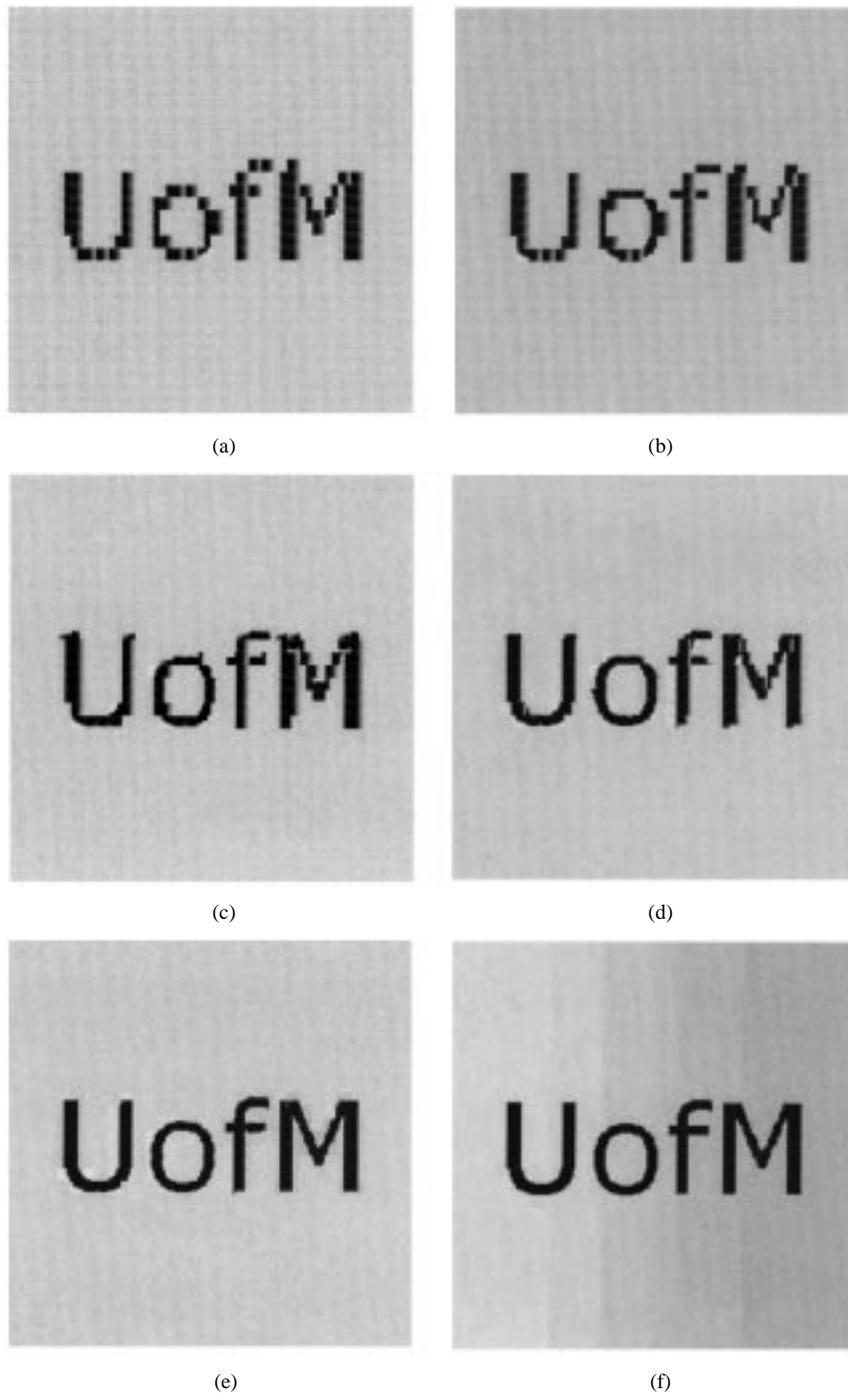


Fig. 9. (a) VQ using modified ART2—four quantization levels, comp. rate: 0.0312 bits/pixel, comp. ratio: 256:1. (b) VQ using modified ART2—eight quantization levels, comp. rate: 0.0468 bits/pixel, comp. ratio: 170.6:1. (c) VQ using modified ART2—16 quantization levels, comp. rate: 0.06625 bits/pixel, comp. ratio: 128:1. (d) VQ using modified ART2—32 quantization levels, comp. rate: 0.0781 bits/pixel, comp. ratio: 102.4:1. (e) VQ using modified ART2—64 quantization levels, comp. rate: 0.0937 bits/pixel, comp. ratio: 85.3:1. (f) VQ using modified ART2—128 quantization levels, comp. rate: 0.1093 bits/pixel, comp. ratio: 73.1:1.

particular number of nodes) modified ART2 outperforms HCL. This is because HCL, due to its mse minimization, is dependent on input data distribution density. Accordingly, in the case of Fig. 8, HCL concentrates most of the nodes within or close to the input vectors related to the background, since, statistically, these represent the most plentiful data. That portion of the image, however, apparently does not contain a significant amount of valuable information, and therefore high decompression quality of the background obtained in Fig. 10(d)–(f) is not

of significant importance from the point of view of human perception.

In contrast to HCL, the modified ART2 algorithm allocates very few nodes to the background. Note that in all the decompressed versions up to 128 quantization levels it remains completely uniform (nonshaded), and therefore considerably different from the original. However, the textual portion of the image, which contains all valuable information, in spite its small direct statistical weight occupies most of the nodes. Therefore,

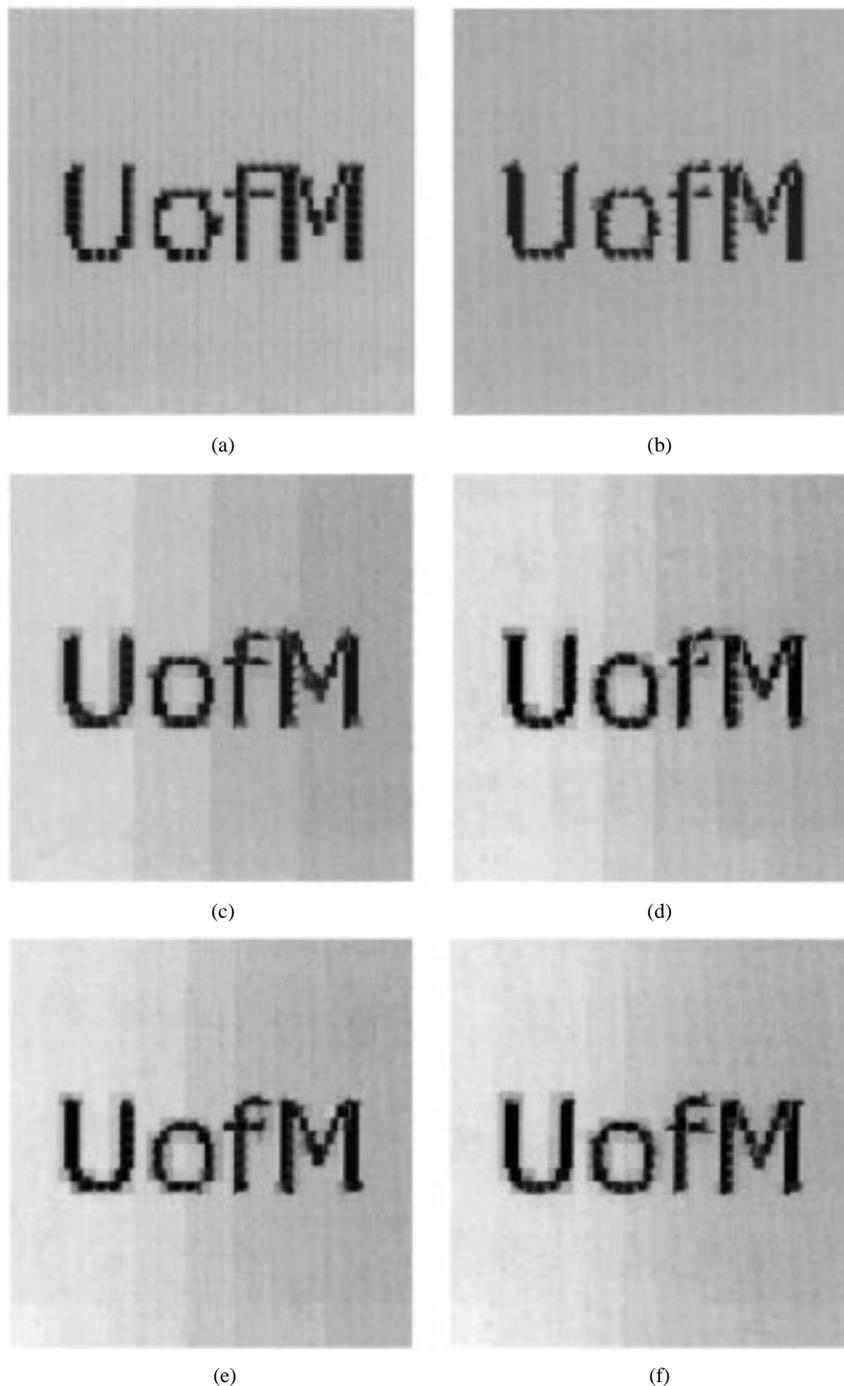


Fig. 10. (a) VQ using HCL—four quantization levels, comp. rate: 0.0312 bits/pixel, comp. ratio: 256 : 1. (b) VQ using HCL—eight quantization levels, comp. rate: 0.0468 bits/pixel, comp. ratio: 170.6 : 1. (c) VQ using HCL—16 quantization levels, comp. rate: 0.06625 bits/pixel, comp. ratio: 128 : 1. (d) VQ using HCL—32 quantization levels, comp. rate: 0.0781 bits/pixel, comp. ratio: 102.4 : 1. (e) VQ using HCL—64 quantization levels, comp. rate: 0.0937 bits/pixel, comp. ratio: 85.3 : 1. (f) VQ using HCL—128 quantization levels, comp. rate: 0.1093 bits/pixel, comp. ratio: 73.1 : 1.

with just 64 quantization levels modified ART2 appears to be capable of providing almost 100% accurate reconstruction of the primary (textual) information [Fig. 9(e)].

It is interesting to observe that although HCL attempts to minimize the mse within the reference vector space, it does not always ensure an equally efficient performance on the pixel level. Moreover, it seems that for a sufficient number of quantization levels modified ART2 provides a considerably lower pixel-to-pixel mse (Table I), i.e., higher SNR (Table II).

### B. Experiment 2

The second experiment is related to the well-known Lena image (Fig. 11).

Figs. 12 and 13 correspond to the compression obtained with the two algorithms using 256 quantization levels. A careful inspection of these images leads to the conclusion that HCL provides evenly distributed distortion, and therefore, a better quality on average. On the other hand, although modified ART2

TABLE I  
mse FOR DECOMPRESSED VERSIONS OF FIG. 8

MSE	Modified ART2	HCL
4 nodes	947.3693	897.8713
8 nodes	834.8995	725.4022
16 nodes	619.4570	411.7769
32 nodes	391.5331	369.3275
64 nodes	237.4687	348.1657
128 nodes	4.0443	333.7010

TABLE II  
SNR FOR DECOMPRESSED VERSIONS OF FIG. 8

SNR	Modified ART2	HCL
4 nodes	16.5746	16.8076
8 nodes	17.1234	17.7340
16 nodes	18.4196	20.1931
32 nodes	20.4121	20.6656
64 nodes	22.5837	20.9219
128 nodes	40.2712	21.1062



Fig. 11. Lena image.



Fig. 12. VQ of Lena image using modified ART2, 256 quantization levels, comp. rate: 0.125 bits/pixel, comp. ratio: 64 : 1, mse: 122.5710, SNR: 20.0655.

is not as efficient as HCL regarding the areas of slowly varying shading without much entropy or information (for example the shoulder, the hat, etc.), it successfully preserves all important details. Accordingly, Fig. 12 appears to be more appropriate for recognition of important features than Fig. 13.

Figs. 14 and 15 are the decompressed versions of the Lena image produced by modified ART2 and HCL with 512 nodes



Fig. 13. VQ of Lena image using HCL, 256 quantization levels, comp rate: 0.125 bits/pixel, comp. ratio: 64 : 1, mse: 382.4340, SNR: 15.1238.



Fig. 14. VQ of Lena image using modified ART2, 512 quantization levels, comp. rate: 0.1406 bits/pixel, comp. ratio: 56.8 : 1, mse: 17.9365, SNR: 28.4120.



Fig. 15. VQ of Lena image using HCL, 512 quantization levels, comp rate: 0.1406 bits/pixel, comp. ratio: 56.8 : 1, mse: 319.4959, SNR: 15.9047.

(quantization levels), respectively. While Fig. 14 is undoubtedly a significant improvement over Fig. 12, it appears that Fig. 15 is almost of the same quality as Fig. 13 (however, note that the mse of Fig. 15 is actually lower than that of Fig. 13). The performance mainly results from the mse minimization property of the algorithm. However, there are some indications that for this particular image and for a large number of nodes (over 256), HCL becomes trapped in a local minimum.

TABLE III  
CODING OF LENA IMAGE USING MULTILEVEL SEARCH STRATEGY

	number of comparisons	coding time compared to serial search strategy
1-level	524288	100 %
2-levels	287976	54.92 %
3-levels	176200	33.60 %
4-levels	123945	23.64 %
5-levels	98346	18.75 %

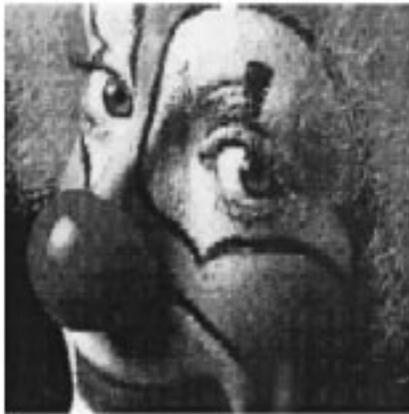


Fig. 16. Clown image.

From a practical point of view, serial search coding (see Section III-A of a  $256 \times 256$  pixel image, based on  $8 \times 8$  pixel subimages, requires each of the 1024 subimages to be matched against each of the code (reference) vectors. Thus, if there are 512 quantization levels, it implies 524 288 comparisons in total. In order to provide a more efficient coding, a multilevel coding strategy is applied.

In particular, 512 reference vectors that correspond to the Lena image, obtained by the modified ART2, are used as the “root” (zeroth level) of a five-level dendogram (search tree). This means that, as described in Section III-A, these 512 are first grouped into 256 clusters (first level), with each cluster represented by the cluster centroid. Then, the 256 centroids are further grouped into new 128 clusters, and similar procedures follow for the next 64, i.e., 32 clusters. Accordingly, the search for the reference vector (among the 512) closest to a new input vector that appears at the input of a VQ coder, begins from the highest level of the tree. Table III shows that with the five-level dendogram the complete coding of the Lena image is performed in just 18.75% of the time required when a serial search strategy is applied. Also, from Table III, it can be observed that fewer levels in the dendogram implies a less efficient coding performance.

Note: Table III does not provide the exact coding times in seconds. Instead, it provides the number of required comparisons (input vectors-to-code vectors) as a valid measure of coding efficiency. We intentionally avoid using “time in seconds,” since with this performance metric the same algorithm could suggest significantly different results depending on the specific machine used for simulation. Also, note that in Table III a 1-level search strategy implies that the tree consists of the zeroth level only, a two-level search strategy implies the zeroth and first level, etc.



Fig. 17. VQ of Clown image using modified ART2, 256 quantization levels, comp. rate: 0.125 bits/pixel comp. ratio: 64 : 1, mse: 351.8208, SNR: 15.8068.



Fig. 18. VQ of Clown image using HCL, 256 quantization levels, comp rate: 0.125 bits/pixel, comp. ratio: 64 : 1, mse: 664.8278, SNR: 13.0429.

### C. Experiment 3

The main purpose of this experiment was to investigate the quality of a decompressed image whose compression is based on the codebook obtained for some other image.

Figs. 19 and 20 are the decompressed versions of the familiar Clown image (Fig. 16) based on the codebooks obtained for the Lena image using modified ART2 and HCL, respectively (see Figs. 17 and 18). According to the mse and SNR measures, the modified ART2 algorithm again produces an image of better quality than HCL. However, the decision as to which algorithm in this case provides better results from the point of view of human perception, is best left to the reader.

### D. Experiment 4

The fourth experiment provides a comparison between VQ based on modified ART2 and the widely used JPEG algorithm, in terms of compression quality and efficiency.

In general, the main principles of vector quantization and JPEG compression are substantially different. As explained in Section I, VQ requires for each subimage (input data block) the nearest code-vector to be selected, and only the index of that code-vector is transmitted through the channel (Fig. 1). Since the overall sizes of the codebook, code-vectors, and image itself are known in advance, it is possible to predict the corresponding



Fig. 19. Compression of Clown image based on the codebook obtained for Lena Image using modified ART2, 256 quantization levels, comp. rate: 0.1406 bits/pixel, comp. ratio: 56.8 : 1, mse: 918.1306, SNR: 11.6410.



Fig. 20. Compression of Clown image based on the codebook obtained for Lena image using HCL, 256 quantization levels, comp rate: 0.1406 bits/pixel, comp. ratio: 56.8 : 1, mse: 1180.2129, SNR: 10.5504.

compression rate and ratio in any particular case. In other words, the total amount of information which has to be sent through the channel when utilizing a VQ compression technique is

$$\text{compressed\_information} = \frac{\text{image\_size}}{\text{subimage\_size}} \cdot \text{codevector\_size[bits]}. \quad (24)$$

However, in VQ techniques the quality of a decompressed image cannot be directly controlled or predicted, and it mainly depends on the nature of the codebook used. In particular, good quality can be expected if the codebook has been obtained for an image similar to the image which is the subject of compression. Otherwise, the decompressed image may significantly differ from the original, or it may appear to be very noisy, even though the actual compression ratio may not be very large.

The JPEG compression algorithm, on the other hand, is based on the following principles. As each  $8 \times 8$  subimage is encountered, its 64 pixels are "level shifted" by  $2^{p-1}$ , where  $2^p$  is the maximum number of gray levels. The discrete cosine transform (DCT) coefficients of the block are then computed, quantized (normalized), and coded using a variable length code scheme, as presented in Fig. 21. (For more details on JPEG see [8], and [12].)

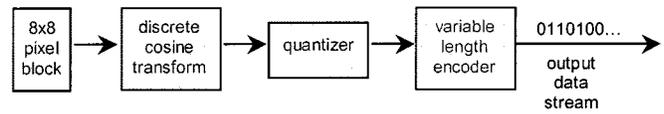


Fig. 21. JPEG compression scheme.



Fig. 22. Decompressed JPEG version of Fig. 8 (compressed filesize: 1660 bytes).



Fig. 23. Decompressed JPEG version of Lena image (compressed filesize: 1639 bytes).

In contrast to the VQ-based compression, JPEG provides better control over the quality of decompressed images, primarily through the adjustment of the quantization step sizes (Fig. 21). Thus, for example, large quantization steps mostly result in substantial compression and significantly reduced quality, with most of the high-frequency coefficients removed. Small quantization steps imply the opposite. However, since the quantization is followed by the encoding process, which introduces some additional compression of the lossless type, it is difficult if not impossible to predict the exact size of the resulting compressed images [8]. In other words, with the JPEG compression algorithm it becomes practically impossible to predict the precise value of the compression ratio required. Therefore, most of the existing image editing tools provide their own subjective compression scales, enabling the users to only qualitatively control the actual size and quality of compressed JPEG files.

Figs. 22–24 were produced by the Matlab Image Processing Toolbox, with the JPEG quality factor set to 2. (In general, this



Fig. 24. Decompressed JPEG version of Clown image (compressed filesize: 1783 bytes).



Fig. 26. Decompressed JPEG version of Bird image (compressed filesize: 1358 bytes), comp. ratio: 51.5 : 1.

TABLE IV  
FILESIZES OF COMPRESSED IMAGES PRODUCED USING JPEG ALGORITHM

image	compressed filesize (information)
Figure 22	1660 [bytes]
Figure 23	1639 [bytes]
Figure 24	1783 [bytes]

TABLE V  
FILESIZES OF COMPRESSED IMAGES PRODUCED USING MODIFIED ART2

image	compressed filesize (information)
Figure 9-f)	896 [bytes]
Figure 12	1024 [bytes]
Figure 16	1024 [bytes]

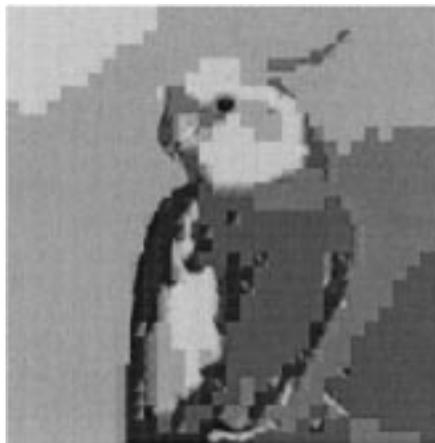


Fig. 27. VQ of Bird image using modified ART2, 64 quantization levels, comp. rate: 0.0937 bits/pixel, comp. ratio: 85.3 : 1.



Fig. 25. Bird image (filesize: 69 902 bytes).



Fig. 28. VQ of Bird image using modified ART2, 128 quantization levels, comp. rate: 0.1093 bits/pixel, comp. ratio: 73.1 : 1.

factor can take any value in the range 0–100, where 0 implies the worst quality but the most intensive reduction in filesize, while 100 implies the best quality and no reduction in filesize.) In our case the quality factor of 2 resulted in images of sizes close to 1024 [bytes] (Table IV). Therefore, an adequate quality-related comparison of these images with the corresponding images produced using VQ based on modified ART2 (as given in Table V) was possible.

Even a very superficial inspection suggests that Figs. 9(f), 12, and 16 are of considerably better quality than Figs. 22–24, respectively. In general, this implies that VQ based on modified

ART2 can outperform the standard JPEG compression scheme at comparable compression ratios provided the VQ codebooks are appropriate to the images being compressed. Note that the ratios in the above figures are well in excess of those normally employed with JPEG.

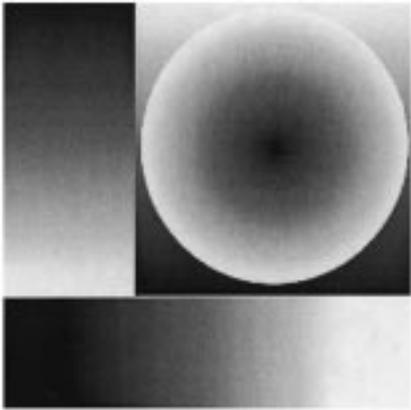


Fig. 29. Slope image (filesize: 69 706 bytes).

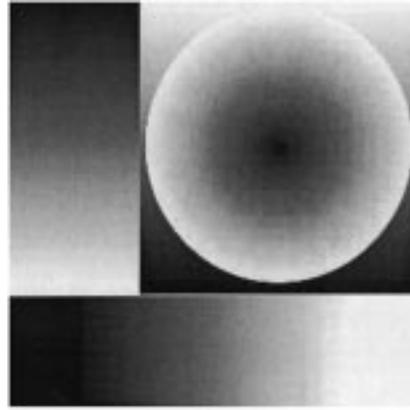


Fig. 32. VQ of Slope image using modified ART2, 256 quantization levels, comp rate: 0.125 bits/pixel, comp. ratio: 64 : 1.

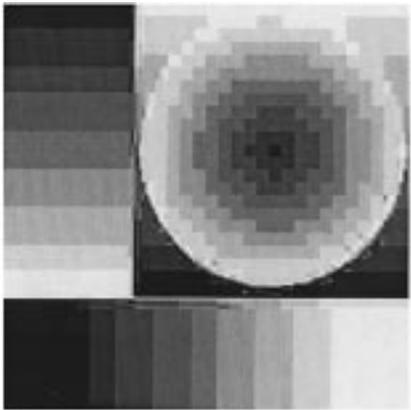


Fig. 30. Decompressed JPEG version of Slope image (compressed filesize: 1537 bytes), comp. ratio: 45.3 : 1.



Fig. 33. Montage image (filesize: 70 514 bytes).

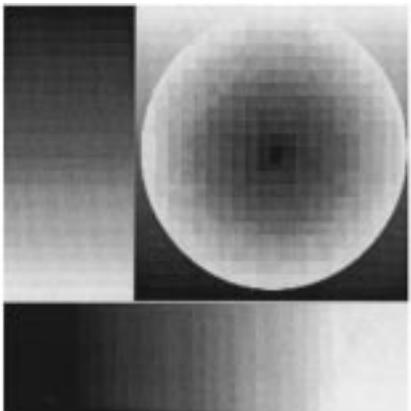


Fig. 31. VQ of Slope image using modified ART2, 128 quantization levels, comp. rate: 0.1093 bits/pixel, comp. ratio: 73.1 : 1.



Fig. 34. Decompressed JPEG version of Montage image (compressed filesize: 1612 bytes), comp. ratio: 43.7 : 1.

### E. Various Experiments

This section contains a series of decompressed images (Figs. 25–40) obtained by experimenting with the JPEG scheme and VQ based on modified ART2.

The main goal of the experiments was to provide a better understanding of the features and compression capabilities of modified ART2, and also to give ground for a more valid comparison against JPEG which is considered to be the most commonly used compression scheme on the Internet today. We hope

the obtained results will motivate further investigations and a wider use of modified ART2 for the purposes of image compression.

### V. CONCLUSION AND FUTURE WORK

In this paper, the main disadvantages of most of the existing VQ techniques based on the mse minimization principle have been described. The theoretical discussion and reported results have suggested the main benefits of utilizing the modified ART2



Fig. 35. VQ of Montage image using modified ART2, 256 quantization levels, comp. rate: 0.125 bits/pixel, comp. ratio: 64 : 1.



Fig. 38. Decompressed JPEG version of Casablanca image (compressed filesize: 1706 bytes), comp. ratio: 41.7 : 1.



Fig. 36. VQ of Montage image using modified ART2, 512 quantization levels, comp. rate: 0.1406 bits/pixel, comp. ratio: 56.9 : 1.



Fig. 39. VQ of Casablanca image using modified ART2, 256 quantization levels, comp. rate: 0.125 bits/pixel, comp. ratio: 64 : 1.



Fig. 37. Casablanca image (filesize: 71 212 bytes).



Fig. 40. VQ of Casablanca image using modified ART2, 512 quantization levels, comp. rate: 0.106 bits/pixel, comp. ratio: 56.9 : 1.

algorithm over the other NN algorithms for the purpose of code-book creation. It has been proven that modified ART2, due to its ability to preserve fine details in decompressed images, is capable of providing particularly satisfactory results when dealing with text-based images, i.e., images of excessively nonuniform distribution. Some situations in which modified ART2 outperformed the well-known JPEG compression algorithm have also been reported.

Although based on the results presented in the paper modified ART2 clearly outperform a typical representative of mse-mini-

mizing neural networks (HCL), we are not claiming its superiority over all other classes of NN algorithms. The performance of NN algorithms based on error criteria other than mse is still to be investigated.

Another area of possible future research is related to the creation and size of codebook in the general case of vector quantization for image compression purposes. In particular, at the present stage the main obstacle toward a mass implementation of VQ techniques, including VQ based on modified ART2 as

described in this paper, seems to be the lack of a standardized codebook. Accordingly, most VQ models assume that a codebook is created for each image compressed, and the knowledge of the given codebook is immediately present at the receiver side, typically at no cost. Of course, this assumption is not entirely realistic, and clearly such a codebook can be expected to provide a noticeably better performance when used in the compression and decompression of the original than of an arbitrary image. One possible solution to the above problem is the utilization of a larger universal codebook, which would ensure equally good performance for a wide class of images. However, every increase in the size of the codebook results in an increase in the number of bits per each code-vector identifier. Accordingly, with a larger codebook lower compression ratios are to be obtained. Still, it should be noted that the number of bits per code-vector identifier, and therefore the size of compressed images, only grows logarithmically with the size of the universal codebook. Thus, the idea of quite large universal codebooks should be entertained (with copies at all sites as in JPEG encoders/decoders) perhaps with progressively larger universal codebooks employed when quality requirements are demanded.

#### ACKNOWLEDGMENT

The authors would like to thank Prof. W. Kinsner for valuable discussions on data compression.

#### REFERENCES

- [1] G. A. Carpenter and S. Grossberg, "ART2: Self-organization of stable category recognition codes for analog input patterns," *Appl. Opt.*, vol. 26, pp. 4919–4930, 1987.
- [2] N. Vljajic, "Adaptive algorithms for hypertext clustering," M.Sc. thesis, Univ. Manitoba, Winnipeg, MB, Canada, 1998.
- [3] N. R. Pal, J. C. Bezdek, and E. C.-K. Tsao, "Generalized clustering networks and Kohonen's self-organizing scheme," *IEEE Trans. Neural Networks*, vol. 4, pp. 549–557, July 1993.
- [4] B. Fritzke, "A growing and splitting elastic network for vector quantization," in *Proc. IEEE-NNSP*, Baltimore, MD, 1993.
- [5] B. Zhang, M. Fu, and H. Yan, "Application of neural 'gas' model in image compression," in *Proc. IEEE-IJCNN*, May 1998, pp. 918–920.
- [6] H. Bischof and A. Leonardis, "Design of vector quantization networks by MDL-based principles," in *Proc. IEEE-IJCNN*, May 1998, pp. 2294–2299.

- [7] B. Fritzke, (1997) Some competitive learning methods. [Online]. Available: <http://www.neuroinformatik.ruhr-uni-bochum.de/ini/VDM/research/gsn/JavaPaper/>
- [8] K. Sayood, *Introduction to Data Compression*. San Francisco, CA: Morgan Kaufmann, 1996.
- [9] S. Haykin, *Neural Networks*. Englewood Cliffs, NJ: Prentice-Hall, 1994.
- [10] C. J. van Rijsbergen, *Information Retrieval*. Boston, MA: Butterworths, 1975.
- [11] H. Demuth and M. Beale, *Neural Network Toolbox for Use with Matlab*. Natick, MA: The Mathworks, Inc., 1996.
- [12] R. C. Gonzales and R. E. Woods, *Digital Image Processing*. New York: Addison-Wesley, 1993.
- [13] C. M. Bishop, *Neural Networks for Pattern Recognition*. Oxford, U.K.: Clarendon, 1995.



**Natalija Vljajic** received the M.Sc. degree in electrical engineering from the University of Manitoba, Winnipeg, MB, Canada, in 1998. She is pursuing the Ph.D. degree at the University of Ottawa, Ottawa, ON, Canada.

She is presently a Research Associate at the University of Ottawa. Her current research interests include artificial neural networks, wireless data communications, and signal and data processing.



**Howard C. Card** (S'68–M'72–SM'83–F'94) received the Ph.D. degree from the University of Manchester, Manchester, U.K., in 1971.

He has held academic appointments at Manchester, Manitoba, Waterloo, and Columbia University, industrial appointments at IBM Watson Research Center and AT&T Bell Labs, and has spent a sabbatical year at Oxford. His interests include device physics and modeling, VLSI systems, parallel processing and artificial intelligence. His current research is on neural networks and stochastic computation, as well as in the relationship between biology, physics, and computation.

Dr. Card has been an Athlone Fellow and a Mullard Fellow. He has received several teaching awards including the Stanton Award for Excellence in Teaching and UMFA-UTS Award. He also has a number of research awards including the NSERC E.W.R. Steacie Memorial Fellowship, the Ross Medal of IEEE Canada, the Rh. Institute Award for Multidisciplinary Research, the Sigma Xi Senior Scientist Award, and the ITAC-NSERC Award for Academic Excellence. He is a Fellow of the Canadian Academy of Engineering and is a registered Professional Engineer.