Adaptive Singlepass Codebook Generation Based on ART Neural Networks

Xueqin Li†, Zhiwei Zhao ‡ and H. D. Cheng ‡
† Department of Electrical Engineering
‡Department of Computer Science
Utah State University
Logan, UT 84322

Abstract

In this paper, we present an adaptive singlepass codebook generation based on adaptive resonance theory(ART) neural networks. The main motivation for VQ codebook generation based on ART neural network (ART-NN) is its capability of discovering cluster structure, on-line learning scheme and solving stability - plasticity dilemma. For static codebook generation, ART-NN based approach yields the results comparable to those using full search LBG algorithm with binary splitting technique in the sense of signal to noise ratio, but with much lower codebook entropy and much lower time complexity. Huffman coding of the VQ indices after applying the proposed ART-NN based approach can provide additional compression. For adaptive codebook generation, the ART-NN based algorithm can effectively on-line adapt to the changing source data statistics because of its singlepass and no batch training nature.

1 Introduction

The vector quantization algorithms and their hardware implementations for reducing the storage or the transmission bit rate for the images and speech signals have been extensively investigated [2 - 6]. It is well known that the key to vector quantization compression is to have a good representative codebook which captures the source data statistics. There are several methods for codebook generation [5], among them the Linde-Buzo-Gray (LBG) algorithm [1], which is locally optimal, has been extensively applied. However, LBG algorithm for deriving a codebook is an iterative and batch mode algorithm, which needs to access to the entire training data set during the training process. The complexity of LBG codebook design method makes it difficult to implement a high-speed VQ adapting to the changing source data statistics. Neural networks have been widely studied [7 - 9]. Artificial neural network methods appear to be very promising for intelligent information processing system because of their massively parallel computing architectures and self organization learning schemes. The neural networks are able to approximate some function which may not be known *a priori*, or to find a near optimal solution for a hard computation problem such as the vector clustering problem. A number of researches have been working on using artificial neural networks for design of VQ codebooks to circumvent the limitations of traditional algorithms. Several neural networks for VQ codebook design have been reported [10 - 20], such as the competitive learning (CL) network [11], Counterpropagation network (CPN)[15 - 17], Kohonen self organization feature map (KSFM) [18 - 20], and frequency sensitive competitive learning (FSCL) network [11].

The features of these neural networks are briefly described as follows:

1. The competitive learning network is an adaptive version of the LBG algorithm with the advantage of simple computation [11]. One problem with this neural network is that it sometimes leads to underutilization of neural units. Another problem is that a competitive learning network does not always learn temporally stable code in response to an arbitrary input environment. The network may make previous learning washed away by more recent learning in response to a changing input environment. The extensive investigation of theses two problems can be found in the literatures [22].

2. Self-organization in CPN and KSFM is a statistical natural-clustering process in which the network performs competitive learning to perceive pattern classes based on data similarity or principal component analysis. One drawback of these two networks is that the nearest-neighbor method causes a large amount of neurons to be underutilized after training. Another drawback is the computational complexity which arises from both the calculation of the neighborhood of winning neurons and updating all members of the neighborhood [10].

3. Frequency-sensitive competitive learning network was proposed to attack the problem of CL's underutilized neurons and retain its simple computation advantage. But it still cannot solve the stability - adaptability dilemma. More recycles are required to improve the learning performance. While using it to generate a VQ codebook, it maximizes the entropy of the codebook so that Huffman coding of the VQ indices would not provide significant additional compression because this algorithm ensures that all codewords in the codebook are updated equally frequently during iterations of the training process [21].

In this paper, we present a new neural network architecture and its algorithm for an adaptive singlepass codebook generation based on Grossberg's adaptive resonance theory (ART) neural network. In Section II, we describe the adaptive resonance theory (ART) neural network. The discussion about ART will be brief since it has already been deeply explored in the literature. Next, we propose an ART-NN based architecture and algorithm for codebook generation. The experimental results and performance comparison with LBG algorithm using binary splitting technique are provided in Section IV. Discussions and conclusions are given in Section V.

2 Why ART neural networks for VQ codebook generation

ART neural network came from an analysis of a simple type of adaptive pattern recognition network – a competitive learning neural network [23]. But in a competitive learning network, certain instabilities can arise so that different neurons might respond to the same input pattern on different occasions. Moreover, later learning can wash away earlier learning if the environment is nonstationary or if novel input patterns arrive [23]. This stability - plasticity dilemma is faced by all intelligent systems capable of autonomously adapting in real time to unexpected changes in nonstationary world. Grossberg and Carpenter developed ART architecture to learn quickly and stably in real time in response to a possibly nonstationary environment with an unlimited number of inputs until it utilizes its full memory capacity [25].

In the ART neural networks, one of the keys to solve the stability - plasticity dilemma is its feedback mechanism between the competitive layer and the input layer of a network. This feedback mechanism protects previously learned information from being washed away while learning new information, automatically switchs between stable and adaptive modes, and stabilizes the encoding of the classes done by the neurons. Grossberg and Carpenter gave the excellent comparison between ART neural networks and other neural networks in the literature [26]. The detailed theoretical discussion of ART neural networks can be found in [23-27].

Since in VQ data compression, each data vector must be represented as one of the codebook indices, the contents of codebook determine the overall performance of the reconstructed output. As the source statistics changes, the codebook should be changed in real time for the current source statistics. How to effectively design the representative codebook with adaptive mode is our motivation to look into the features of ART neural networks. *The main reason for VQ codebook generation based on ART neural network is its capability of discovering cluster structure, on-line learning scheme and solving stability - plasticity dilemma.*

3 Neural Network Architecture and Algorithm for Codebook Generation

The proposed neural network architecture based on ART for codebook generation is shown in Fig. 1. The network consists of two subsystems: attentional subsystem and orienting subsystem. Attentional subsystem consists of three layers: an input layer that distributes the input vector to the neurons on the second layer, where each neuron computes the distortion (Euclidean distance) between its weight vector and the input vector; and an output layer based on winner-take-all network which determines the winning neuron from the distortions computed by the second layer neurons. There are k neurons in the input layer and M neurons in the second and output layers respectively, where k is the dimension

of the input vector, and M is the capacity of the neuron network which should be greater than or equal to the specified size of the codebook. The jth weight vector $w_j = (w_{1j}, w_{2j}, ..., w_{kj})$ and j = 1, ..., M in the second layer is the jth codevector. The codebook is built in the synaptic weights of the second layer. When the input vector presents to the input layer, the second layer neurons calculate the Euclidean distances between the input vector and the weight vectors, and the winner-take-all policy determines the neuron which has the smallest distortion error (Euclidean distance). The orienting subsystem always ensures that the worst-cast distortion error per vector is upper bounded by the distortion threshold. That is, we maintain "vigilance" by setting distortion threshold (radii) for the weight vectors (cluster centers). An input vector lying outside of a hypersphere would not be considered as belonging to the neuron even if the winner-take-all layer assigned it to that neuron. In this situation, the input vector is assigned to one of uncommitted neurons in the second layer until the capacity of the neuron network is full. If the input vector lying inside of a hypersphere (bounded by the distortion threshold), the winner-take-all layer assigns this input vector to neuron m which has the smallest distortion error. Then the weight vector w_m of neuron m will be updated to be a centroid of all input vectors (including the new input vector) which have fell into this weight vector since training began. All other weight vectors are left unchanged. The weight vector w_m for neuron m is updated as follows:

$$W_m = \frac{N_m}{N_m + 1} W_m + \frac{1}{N_m + 1} X$$

Where N_m is the number of input vectors which fell into the weight vector of the neuron m since training began. X is the current input vector. In this way, the training rule moves the winning codevector toward the current input vector by a reasonable fraction amount.

The basic ART training rule for the codebook generation is as follows: *Parameters* T_{dist} : distortion threshold for generating a new neuron (codevector). N_i : the number of training vectors belongings to the weight vector C_i. num_N: the number of committed neurons. L : the specified size of a codebook.

1. If the initial codebook is empty, the first input vector is used to set the weight vector of the first neuron:

   $$W_1 = X_1, N_1 = 1, num_N = 1.$$

   Otherwise, we have an initial codebook with $M \leq L$, we preload the initial codebook to the weight vectors as follows: for$(i = 1; i \leq M; i++)$
   $\{W_i = C_i; N_i = 1\};$
   $num_N = M$

2. Present a new input vector X to the input layer;

3. Compute the distortion D_i between an input vector X and all weight vectors of the second layer neurons:

   $$D_i = d(X, W_i) \quad 1 \leq i \leq num_N;$$

   $d(X, W_i)$ can be Euclidean distance or absolute distance.

4. Select the neuron on the distortion calculation layer which has the smallest distortion, then set its output to 1 and set other neuron outputs to 0:

   $$O_m = \begin{cases} 1 & if \ \ D_m = min\{D_i\} \\ 0 & \text{otherwise} \end{cases} \qquad (1)$$

5. if $D_m \leq T_{dist}$, update the weight vector w_m with the training rule:

   $$W_m = \frac{N_m}{N_m + 1} W_m + \frac{1}{N_m + 1} X$$

   and $N_m = N_m + 1$. If the buffer of training vectors is empty, go to step 7, else go to step 2.

6. (In this case, $D_m > T_{dist}$) generate a new neuron (codevector) by setting the weight vector of an uncommitted neuron as the input vector X, i.e:

   $$W_{num_N} = X$$

   and increase the number of committed neurons by:

   $$num_N = num_N + 1$$

   go to step 2;

7. If $num_N > L$, delete the neurons which have fewer training vectors. (In this way, we realize the maximum utilization of all neurons in the distortion calculation layer and the winner-take-all layer when the size of the codebook is fixed.)

In this training mode, for every input vector, a new weight vector will be generated if the input statistics changes, or the weight vector is updated by encouraging a shift toward the centroid. The whole codebook generation is realized by class formation and centroid approximation. All training vectors are presented once.

4 Performance Comparisons in Image Coding

A. Performance comparison: The full search mean-residual VQ (MRVQ) and multistage VQ (MSVQ) [3] are used to evaluate the performance of the codebooks generated by both the proposed ART-NN based approach and the LBG approach using binary splitting technique. A fixed image block size of 4x4 pixels is used. The training set consists of 114688 vectors extracted from seven human figure images. The distortion of the encoded image is measured by the peak signal-to-noise (PSNR) which is defined as

$$PSNR = 10\log_{10}\{\frac{255^2}{\frac{1}{M\times N}\sum_{i=1}^{M}\sum_{j=1}^{N}(x_{ij}-\hat{x}_{ij})^2}\} \quad (2)$$

where $M \times N$=size of image, x_{ij}=pixel value of the original image at coordinate (i,j) and $\hat{x}_{ij}$=pixel value of the reproduced image at coordinate (i,j). Firstly, the static codebooks were generated by both the LBG and ART-NN based algorithms, i.e., the codebooks were trained before the image encoding took place and were not updated during image encoding. The only computation during image encoding was to find the closest codevector in the codebook to match the input vector of the original image. Table I shows the coding performance for the training images using MRVQ. Table II shows the coding performance for the training images using MSVQ.

From Table I and Table II, for these static codebooks, the proposed ART-NN based codebook generation approach yields the results comparable to those using the full search LBG in the sense of peak signal to noise ratio. However, the potential power of ART-NN based codebook generation is its adaptability. Table III shows the performances using 6-bit local , non-adaptive and adaptive MRVQ codebooks. The local MRVQ codebook was generated by means of LBG algorithm and using single image as training image. Each training image has its own local codebook. Then we used this local codebook to encode the training image. Non-adaptive MRVQ codebook was obtained by the LBG algorithm and the training set is the seven human figure images. Using ART-NN based algorithm, the adaptive MRVQ codebook was generated based on the above non-adaptive MRVQ codebook and the current image. If the current image has similar statistics to the training images which are used for training non-adaptive MRVQ codebook by LBG algorithm, most codevectors of the adaptive MRVQ codebook are similar to the codevectors of non-adaptive MRVQ codebook. On the other hand, if the current image has different statistics with the training images, there will be relatively more new codevectors to be generated and more old codevectors will be significantly updated by the proposed ART-NN based algorithm. For example, in the Table III, the 6-bit non-adaptive MRVQ codebook was previously generated by LBG algorithm using seven human figure images as training images. If the current image is pepper image which has different statistics with the human figure images, the number of new codevectors generated by the ART-NN based algorithm is 11 and the number of codevectors which have been significantly updated is 15. The new codevectors are generated according to the distortion threshold T_{dist}. That is, if the incoming input vector of the current image could not find the closest codevector bounded by the distortion threshold, a new codevector will be generated to accommodate the new input vector. The old codevectors will be updated and replaced if the old codevector has larger difference with the corresponding updated codevector (beyond the updated threshold T_{update}). The adaptive codebook is obtained by using the new and frequently used codevectors to replace the old and rarely used codevectors in the old codebook or update old codevectors which have larger differences with the corresponding updated codevectors. In the above testing, lena and swing images are within the training set for generating old codebook. Both peppers and yatcht images are outside training set. For all the images (inside or outside the training set), there must be $PSNR_{local} > PSNR_a > PSNR_{na}$. For the images outside the training set, $PSNR_a$'s have been improved greatly compared with their $PSNR_{na}$'s. For pepper image, the improvement is 4.43 (dB). For the yacht image, the improvement is 0.75 (dB). For the images within the training set, their $PSNR_a$'s have a little improvement compared with their $PSNR_{na}$. For lena image, such improvement is 0.31 (dB). For the swing image, the improvement is 0.065 (dB). Therefore, for the images which are within the training set or have the similar statistics to the training set, we do not need to change the old codebook. Otherwise, we should change the codebook (partially or entirely) so that the new codebook is able to adapt to the statistics of the incoming image. Whether to change the old codebook and (if so) how to change it depend on the number of generated new codevectors and the number of the updated codevectors. Table IV shows the performances using 8-bit local, non-adaptive and adaptive 8 MRVQ

codebooks. From the Table III and IV, the number of the generated new codevectors and the number of the updated codevectors for the images outside the training set are larger than those for the images inside the training set.

Parameters.

- $PSNR_{local}$: Peak signal to noise ratio when we encode the current image applying the codebook which was generated using only this image as the training data.
- $PSNR_{na}$: Peak signal to noise ratio when we encode the current image applying the codebook which was generated using seven human figure images as training data.
- $PSNR_a$: Peak signal to noise ratio when we encode the current image by adapting the old codebook to the local statistics of the current image.
- N_{num}: the number of generated new codevectors;
- U_{num}: the number of significantly updated codevectors.

Because the proposed ART-NN based codebook generation is a singlepass algorithm. The codebook's updating or changing can be done on-line with the image encoding.

B. Codebook entropies: Codebook entropies give the lower bound of the number of bits with which each vector could be encoded if Huffman encoding is employed after VQ. If the codebook entropy is much lower than the number of bits to represent the generated codebook, Huffman coding can contribute significant additional compression. The entropy of a codebook is calculated as:

$$E = -\sum_{i=1}^{L} P_i \log_2(P_i)$$

where P_i is the frequency with which codevector i was used in encoding the whole training images. We used two kinds of training image data: i) seven human figure images; ii) 5 fingerprint images. Table V and VI show the codebook entropies obtained from the encoding process. Fig. 2 shows the codevector distribution when we encode fingerprint images using two 6-bit codebooks which are generated by the LBG and ART-NN based algorithm, respectively. In Fig. 2, the dot curve represents the codevector distribution for LBG algorithm. The solid curve represents the codevector distribution for the proposed ART-NN algorithm. X-axis indicates the index of the codevectors. Y-axis indicates the number of input vectors used for the codevector. The proposed ART-NN based training process generates codebook with lower entropies because the codebooks are constructed with unequal probable codevectors. As a consequence, Huffman coding after ART-based VQ can provide more additional compression.

5 Conclusions and Discussions

1. Conclusions

   (a) The proposed algorithm yields the results comparable to those using the LBG algorithm in the sense of mean square errors and signal-to-noise ratios.

   (b) It is a singlepass algorithm and does not require "batch" of the training data vectors, and these features are essential and critical for many real time implementation.

   (c) Simple computation, high utilization of neurons, robustness and adaptive clustering capability make it ideally suited for the adaptive codebook generation.

   (d) It reduces the entropy of the codebook so that Huffman coding of the VQ indices will provide additional compression.

   (e) It allows the option of either minimizing rate subject to a distortion constraint or minimizing coding distortion subject to a rate constraint.

   (f) A VLSI or optical implementation is feasible [27 -30].

2. Discussions:

   - Memory capacity The jmajor feature of ART neural networks is their capability of on-line learning new input patterns without washing away the previously learned patterns. But we can fully realize this feature only under the assumption that such a neural network architecture has as much memory capacity as required. In the real world, we only have limited memory capacity. When all the neurons in the network have been used (full capacity) and a novel input vector has arrived, we can adopt least frequently used replacement or least recently used replacement policy to replace the idling neuron.

- the relationship between distortion threshold and the codebook size There is no accurate relationship between the distortion threshold for generating new codevectors and the codebook size. Generally speaking, the higher distortion threshold is, the smaller the codebook size is. This allows the option of either minimizing rate subject to a distortion constraint or minimizing coding distortion subject to a rate constraint.

REFERENCES

1. Y. Linde, A. Buzo and R. M. Gray, "An algorithm for vector quantizer design," IEEE Trans. Commun., Vol. COM-28, pp.84 -95, Jan. 1980.
2. N. M. Nasrabadi and R. A. King, "Image Coding Using Vector Quantization: A review," IEEE Trans. Commun., Vol. 36, No. 8, pp.957 -971, Aug. 1988.
3. A. Gerso and R. M. Gray, *Vector Quantization and Signal Compression*, Kluwer Academic Publishers, 1991.
4. X. M. Wang and S. M. Shende, "Online Compression of Vide sequences using adaptive VQ codebooks," IEEE Data compression Conference '94, pp185 - 194, 1994.
5. C. -M. Huang and R. W. Harris, " A comparison of several Vector Quantization Codebook Generation Approaches, " IEEE Transactions on Image Processing, Vol. 2, No. 1 , Jan. 1993.
6. J. A. Freeman and D. M. Skapurn, *Neural Networks – Algorithms, Applications, and Programming Techniques*, Addison - Wesley Pulishing Company, 1991.
7. Y. H. Pao, *Adaptive Pattern Recognition and Neural Networks,* Addison - Wesley Publishing Company, 1989.
8. J. M. Zurada, *Introduction to Artificial Neural Systems,*West Publishing company, 1992.
9. S. Haykin, *Neural Networks - A comprehensive foundation,* Macmillan College Publishing Company, 1994.
10. S.C. Ahalt, A. K. Krishnamurthy, P. Chen, and D. E. Melton, "Competitive Learning Algorithms for Vector Quantization," Neural Networks, Vol. 3, pp.277-290, 1990.
11. A. K. Krishnamurthy, S. C. Ahalt, D. Melton and P. Chen, "Neural Networks for Vector Quantization of Speech and Images," IEEE Journal on Selected Areas on Communications, Vol. 8, p.1449-1457, Oct, 1990.
12. S. C. Ahalt, P. Chen and A. K. Krishnamurthy, "Performance Analysis of Two Image Vector Quantization Techniques," Proceedings of the International Joint Conference on Neural Networks, Vol. I, (Washington D. C), pp. 169 - 175, June 18-22, 1989.
13. N. Mohsenian, S. A. Rizvi, and N. M. Nasrabadi, "Predictive vector quantization using a neural network approach," Optical Engineering, Vol. 32, No 7, pp.1503 - 1513, July 1993.
14. R. H. Nielsen, "Counterpropagation networks," Applied Optics, Vol. 26, pp.4979 - 4984, 1987.
15. R. H. Nielsen, " Applications of Counterpropagation networks," IEEE Transactions on Neural Networks, Vol. 1, pp.131-131, 1988.
16. W. Chang, H. S. Soliman, and A. H. Sung,"Image data compression using counterpropagation network," Conference Proceedings of IEEE systems, Man, and Cybernetics, Vol. 1, pp.405-409, Chicago, Illinois, Oct. 1992.
17. T. Kohonen, "Self-organized formation of topologically correct feature maps"' Biological Cybernetics, Vol. 43, pp.59-69, 1982.
18. T. Kohonen, "Self-Organization and associative memory , *2nd* ed, New York, NY: Springer - Verlog, 1988.
19. N. M. Nasrabadi and Y. Feng,"Vector Quantization of images based upon the Kohonen self-organization feature maps," IEEE Int. Conf. on Neural Networks I, pp.101-108, June, 1988.
20. T. Kohonen, "Some practical Aspects of the Self-organizing Maps," IJCNN, Washington, DC, Vol. II, pp.253-256, Jan 15- 19, 1990.
21. J. E. Fowler and S. C. Ahalt, "Robust, high-fidelity coding technique based on entropy-biased ANN codebooks", SPIE Vol 1966 Science of Artificial Neural Network II, pp108 - pp117, 1993.
22. S. Grossberg, "Competitive learning: from interactive activation to adaptive resonance," Cognitive Science, No. 11, pp.23-63, 1987.

23. G. A. Carpenter and S. Grossberg,"A massive parallel architecture for a self-organizing neural pattern recognition machine," Computer Vision, Graphics, and Image Processing, No. 37, pp.54-115, 1987.

24. G. A. Carpenter and S. Grossberg,"The ART of adaptive pattern recognition by a self-organizing neural network," Computer, pp.77-88, March 1988

25. G. A. Carpenter and S. Grossberg," ART2: Self-organization of stable category recognition codes for analog input patterns," Appl. Opt., Vol. 22, pp. 4919- 4930, 1987.

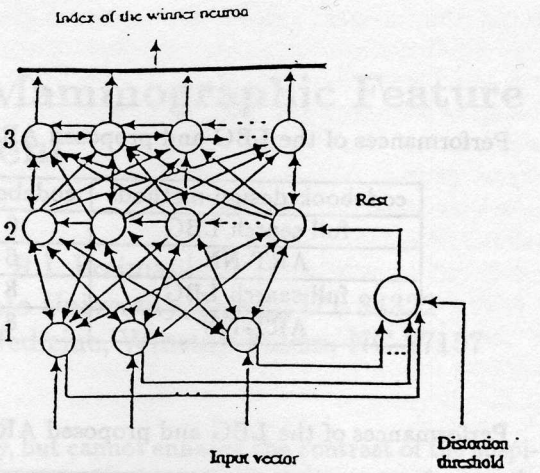
26. G. A. Carpenter, S. Grossberg, and D. B. Rosen,"ART2-A: An adaptive resonance algorithm for rapid category learning and recognition," Neural Networks, Vol. 4, 1991.

27. W. C. Fang, B. J. Shen, O. T. C. Chen and J. Choi, " A VLSI neural processor for image data compression using self-organization networks," IEEE Transactions on Neural Networks, Vol 3, pp.506-518, 1992.

28. J. S. Kane and M. J. Paquin, "POPART: Partial Optical Implementation of A daptive Resonance Theory 2," IEEE Transactions on Neural Networks, Vol. 4, No. 4, pp.695 - 702, July 1993.

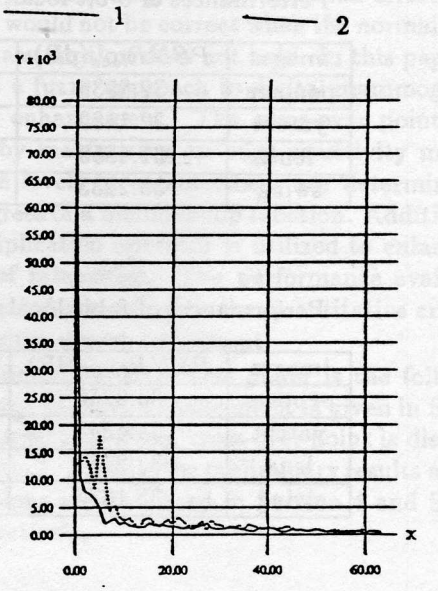
29. D. C. Wunsch II, T. P. Caudell, C. David Capps, R. J. Marks II, and R. A. Falk, "An optoelectronic Implementation of the Adaptive REsonance Neural Networks," IEEE Transaction on Neural Networks, Vol. 4, No. 4, pp.673-683, July 1993.

30. K. W. Przytul, and V. K. Prasann, *Parallel Digital Implementations of Neural Networks*, Prentice-Hall, 1993.



1. represents input layer
2. distortion calculation layer
3. winner-take-all layer

Fig. 1. The proposed ART based neural network for codebook generation.



1. results by LBG
2. results by proposed method

Fig. 2. The codevector distribution of 6-bit codebooks generated by LBG and proposed approaches for the finger print images.

TABLE I

Performances of the LBG and proposed ART-NN based algorithms for training images using MRVQ

codebook design methods	codebook's size	PSNR of coded training images(dB)
full search LBG	6 bits	30.9454
ART-NN	6 bits	30.4582
full search LBG	8 bits	32.3049
ART-NN	8 bits	31.7889

TABLE II

Performances of the LBG and proposed ART-NN based algorithms for training images using MSVQ

codebook design methods	codebook's size	PSNR of coded training images(dB)
full search LBG	6 bits-6 bits	31.5987
ART-NN	6 bits-6 bits	30.9985
full search LBG	4 bits-4 bits-4 bits	30.5131
ART-NN	4 bits-4 bits-4 bits	30.5831

TABLE III

Performances of 6-bit local, non-adaptive and adaptive MRVQ codebooks

images	$PSNR_{loca}$(dB)	$PSNR_{na}$(dB)	$PSNR_a$(dB)	N_{num}	U_{num}
peppers	32.8539	28.1353	32.5660	11	15
yatcht	32.1189	30.5359	31.2806	8	10
lena	31.4855	30.8331	31.1420	4	12
swing	33.2233	32.5756	32.6407	0	7

TABLE IV

Performances of 8-bit local, non-adaptive and adaptive MRVQ codebooks

images	$PSNR_{local}$(dB)	$PSNR_{na}$(dB)	$PSNR_a$(dB)	N_{num}	U_{num}
peppers	35.0259	29.3325	32.7851	46	40
yatcht	34.4241	32.2279	33.4955	44	48
lena	33.2170	32.4360	32.7851	32	40
swing	35.1376	34.0297	34.1654	0	28

TABLE V

6 bit codebook entropies of LBG/ART-NN

Images	LBG	ART-NN
human figure	4.34	2.25
fingerprint	5.22	4.05

TABLE VI

8 bit codebook entropies of LBG/ART-NN

Images	LBG	ART-NN
human figure	6.27	4.11
fingerprint	7.10	5.71