Data Compression by the Recursive Algorithm of Exponential Bidirectional Associative Memory

Chua-Chin Wang and Chang-Rong Tsai

***Abstract*— A novel data compression algorithm utilizing the histogram and the high-capacity exponential bidirectional associative memory (eBAM) is presented. Since eBAM has been proved to possess high capacity and fault tolerance, it is suitable to be utilized in the data compression using the table-lookup scheme. The histogram approach is employed to extract the feature vectors in the given data. The result of the simulation of the proposed algorithm turns out to be better than the traditional methods.**

***Index Terms*—Bidirectional associative memory (BAM), exponential BAM, histogram, SNR, vector quantization (VQ).**

I. Introduction

THE objective of data compression—a high compression rate using a small amount of space with low distortion of the decompressed data—has long been a problem. Since the advent of multimedia systems, data compression has been an unresolved problem mainly due to the exchange of tremendous amounts of image, video, speech, and text data. Many approaches have been proposed for data compression, e.g., vector quantization (VQ), JPEG [14], MPEG [3], [10], CD-I [12], and DVI [11]. As to real-time video data compression, Frost [2] and Huang [4], respectively, have detailed their methods. Most of the approaches are prone to generate errors if the transmission channels are not flawless. Besides, neural networks have also been involved in the data compression application. The effect, however, might not be outstanding [9]. The viewpoint of the prior neural network applications for the data compression is that the researchers deemed the data compression and decompression are learning tasks [1], [8], [13]. Even though this viewpoint is correct in some cases, the current neural networks cannot have satisfactory performance.

Fig. 1 is a prototype of a transmission channel for data compression; the number or the dimension of the vectors traveling in the transmission channel must be less than that of the input and output vectors. Both the compression processor and decompression processor are two associative memories which are encoded with desired feature vectors of given data. Hence, how to encode the feature vectors efficiently becomes an important issue. Thanks to the evolution concept of associative memories [5], [7], [15], it is feasible to apply the similar methodology to retrieve the vector table in data compression/decompression. We therefore develop the prototype of the exponential bidirectional associative memory (eBAM) algorithm integrated with histogram approach to examine the feasibility.

Manuscript received April 1, 1994; revised February 26, 1995, April 15, 1996, and February 1, 1997. This work was supported in part by the National Science Council under Grant NSC 83-0408-E-110-014.

The authors are with the Department of Electrical Engineering, National Sun Yat-Sen University, Kaohsiung, Taiwan 80424, R.O.C. (e-mail: ccwang@ee.nsysu.edu.tw).

Publisher Item Identifier S 1083-4419(98)02184-0.

The eBAM has been proposed by Jeng *et al*. [5] and proved with high capacity by Wang [15], [16]. Its intrinsic high capacity encourages the encoding of pattern pairs for feature vectors of given data. The compression processor and the decompression processor are symmetric and have n neurons attached to the transmission channel and p neurons on the I/P and O/P, respectively, while $n < p$. We will use the histogram method to extract the feature vectors of the given data, and then encode these feature vectors in the eBAM's. Thus, the data compression and decompression can be achieved. The most attractive reason for using the eBAM as the processors is that this associative memory has a high capacity to encode vector pairs and fault tolerance ability to retrieve the encoded vector pairs [15], even in a lossy transmission channel.

In this paper, we adopt the histogram approach to build a so-called ordered histogram feature vectors table and then use the eBAM to compress/decompress the given data. Detailed simulations have been conducted showing impressive performance.

II. Framework of the eBAM Algorithm

Our data compression/decompression algorithm can be divided into two parts: the histogram preprocessor and the eBAM recursive table lookup. The former one is dedicated to extract the feature vectors of the given data, while the later is designed to compress the feature vectors into tag vectors which will be transmitted in the channel, and decompress the tag vectors at the receiving end by recursive table lookup.

A. Ordered Histogram

Although the histogram approach is a well known skill, we have to redefine it more clearly so as to integrate with the eBAM recursive algorithm. Suppose we are given a set of original data vectors, $A = \{a_1, a_2, \cdots, a_D\}$ where $a_j \in R^k, R = \{-1, +1\}$. Note that the components of the data vectors are bipolar, because the neurons are assumed to be in either **on** or **off** state. Some of these vectors are identical, which are deemed as vectors repeatedly showing up in the entire set. The goal of the histogram approach is to extract all

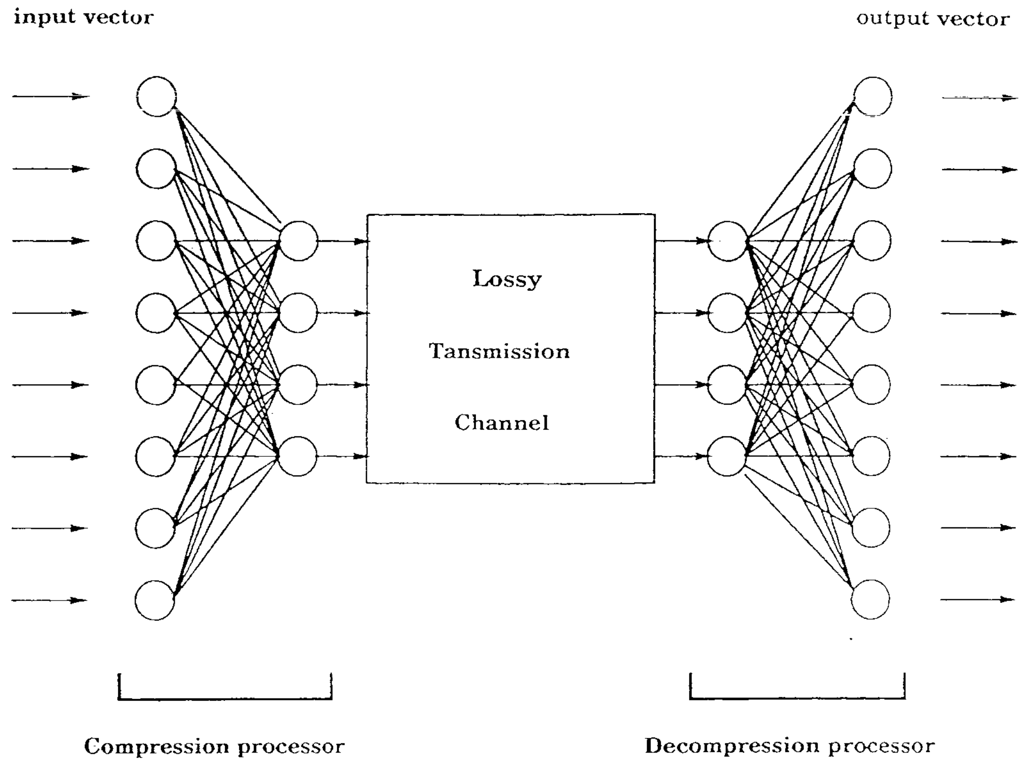

Fig. 1. A prototype of the data compression/decompression system.

of the unique vectors of A and count the number of repetition of each unique vector. Thus, a histogram mapping H can be defined as

$$H\colon A \rightarrow B, \quad \text{where } B = \{b_1, b_2, \cdots, b_N\}$$

where B is called the histogram table, N is the size of the table, and $b_i, b_j \in A, b_i \neq b_j$ for $i \neq j$. According to the above definition, the H operator partitions the A into N groups, while each group is represented by a vector b_i and a count number associated with this group

$$\text{count}(b_i) = \text{no. of } a_k \in A, k = 1, \cdots, D, \quad \text{where} \quad a_k = b_i.$$

For the sake of simplicity, we use c_i to denote $\text{count}(b_i)$ in the following text.

Repetitions of a vector imply the degree of the importance of this vector. In other words, repetitions imply feature vectors. In the procedure of data compression/decompression, we always vow to keep feature vectors as much as possible such that the distortion of the decompressed data will be as small as possible. Therefore, it is necessary to sort the above histogram table B according to the count associated to each vector in B

$$OH\colon B \rightarrow G, \quad \text{where} \quad G = \{g_1, g_2, \cdots, g_N\}$$

where $\text{count}(g_1) \geq \text{count}(g_2) \geq \cdots \text{count}(g_N)$ and $g_i \in B$. The entire operation of the OH operator is listed as the following pseudo codes:

$$\begin{aligned}
&\textbf{initialization: } k = 1, G = \emptyset, B \\
&\textbf{while } (B \neq \emptyset)\{ \\
&\quad \text{find } \exists\, i, c_i \geq c_j, \forall j \neq i; \\
&\quad g_k = b_i; \\
&\quad \text{count}(g_k) = c_i; \\
&\quad G = G \cup \{g_k\}; \\
&\quad B = B - \{b_i\}; \\
&\quad k = k + 1; \\
&\}.
\end{aligned}$$

We call the G is the ordered histogram table of data set $Ag_i \in G, i = 1, 2, \cdots, N$ the ordered feature vectors.

Therefore, we have an ordered histogram in which the feature vectors of the given data are listed according to their individual importance, i.e., the number of the repetitions.

B. eBAM algorithm

Before the discussion of the whole data compression/decompression system, it is necessary to discuss the eBAM and its characteristics. Suppose we are given M bipolar pattern pairs to be encoded, which are

$$\{(X_1, Y_1), (X_2, Y_2), \cdots, (X_M, Y_M)\} \tag{1}$$

where

$$X_i = (x_{i1}, x_{i2}, \cdots, x_{in}), \quad Y_i = (y_{i1}, y_{i2}, \cdots, y_{ip})$$

$X_i \neq X_j, i \neq j$ and $Y_i \neq Y_j, i \neq j$. Instead of using Kosko's approach [6], we use the following evolution equations in the recall process of the eBAM

$$y_k = \begin{cases} 1, & \text{if } \sum_{i=1}^{M} y_{ik} b^{X_i \cdot X} \geq 0 \\ -1, & \text{if } \sum_{i=1}^{M} y_{ik} b^{X_i \cdot X} < 0 \end{cases}$$

$$x_k = \begin{cases} 1, & \text{if } \sum_{i=1}^{M} x_{ik} b^{Y_i \cdot Y} \geq 0 \\ -1, & \text{if } \sum_{i=1}^{M} x_{ik} b^{Y_i \cdot Y} < 0 \end{cases} \tag{2}$$

where X and Y are input retrieval pattern vectors. b is a positive number, $b > 1$ while "$\cdot$" represents the inner product operator. x_k and x_{ik} are the kth bits of X and the X_i, respectively, and y_k and y_{ik} are for Y and the Y_i, respectively. The reasons for using an exponential scheme are to enlarge the attraction radius of every stored pattern pair and to augment the desired pattern in the recall reverberation process. The energy function of this network is defined as

$$E(X,Y) = -\sum_{i=1}^{M} b^{X_i \cdot X} - \sum_{i=1}^{M} b^{Y_i \cdot Y}. \tag{3}$$

The above energy function places every stored pattern pair in its own local minimum on the entire energy surface if b is large enough [15]. During the recall reverberation process, the X and Y vectors will recall each other recursively and alternatively until the $E(X,Y)$ reaches a final value which is supposed to be a local minimum. The convergence of this process has been proved in [5] and [15].

We have computed the capacity of the exponential BAM [15]

$$M < 1 + \frac{2^{n-2} b^4}{(1 + b^{-4})^{n-1}} \tag{4}$$

where n is assumed to the $\min(n,p)$ without any loss of generality.

In the above equation, the given pattern, say X, to recall the corresponding pattern pair is exactly the same as one of the stored X_i's so that the corresponding Y_i will be recalled. Suppose we initialize the eBAM with an input pattern, X which is r bits away from the nearest pattern X_h where $r < (n/2)$. (If $r > (n/2)$ then X will be closer to $\overline{X}_h$ than X_h.) And this input pattern, X can still recall the nearest pattern, X_h and its corresponding pattern Y_h. In Appendix I, we have proved the following conclusion.

Theorem 1: Given a attraction radius, r where $r < (n/2)$, the maximal capacity for an eBAM to store pattern pairs is

$$M < M_{\max} = 1 + \frac{2^{(n-2)}}{(n-r-1) \cdot b^{4(r-2)}}. \tag{5}$$

Theorem 2: The bit-error probability of the eBAM is

$$P_e \approx \frac{1}{\sqrt{2\pi}} \cdot \frac{(M-1)^{1/2} 2^{(-(n/2)+1)} (n-r-1)^{1/2}}{b^{(4-2r)}} \cdot \exp\left[-\frac{b^{(8-4r)}}{2(M-1) \cdot 2^{(-n+2)} \cdot (n-r-1)}\right]. \tag{6}$$

C. Ordered Histogram Combined with eBAM

Since an ordered histogram can be extracted from any given data, it is convenient to utilize the histogram to proceed the data compression. The basic idea is to encode the important feature vectors of the histogram table in the eBAM, because eBAM is proved to possess high capacity and low bit-error probability. The details of the eBAM recursive algorithm, hence, can be tabulated as follows.

1) Select the first N' important feature vectors from the ordered histogram as the vectors to be encoded and transmitted, where $N' \leq N$. Hence, N' determines the compression rate and the quality of the decompressed data. If the N' is large, then the compression rate will be low but the quality of the decompressed data will be high, and vice versa.
2) Generate a unique **tag vector** for each selected feature vector randomly. The dimension of the tag vectors, p must be smaller than that of the feature vectors, n. If not so, then the data compression would be meaningless. Note that the dimension of the tag vectors depends on N'. In order to represent N' vectors, then the dimension of the tag vectors must be at least $\log_2 N'$. For instance, if first 64 feature vectors are chosen, then at least 64 tag vectors are needed. That implies the dimension of the tag vectors is $\log_2\ 64 = 6$.
3) Then the vector pairs encoded in eBAM, i.e., the feature vectors table, are transmitted. Every vector of the given data is present to the compression eBAM such that a corresponding tag vector will be recalled and transmitted.
4) The decompression eBAM receives a tag vector and then recall a corresponding feature vector.

Basically, the eBAM data compression/decompression method utilizes the characteristics of the eBAM, high capacity, low error probability, and fault tolerance, which can be concluded from Theorems 1 and 2. In order to show its performance, several indexes for multimedia and image processing are defined as follows and then compared with other methods.

Compression Rate: The ratio of the size of the original data to the size of the compressed data.

$$CDR = \frac{D \times n}{D \log_2 N' + N' \times (n + \log_2 N')}. \tag{7}$$

Signal-to-Noise Ratio: The similarity of the decompressed data relative to the original data

$$\mathrm{SNR} = 10 \times \log_{10} \left(\frac{255^2}{\frac{\sum_i (x_i - \hat{x}_i)(x_i - \hat{x}_i)^t}{D}} \right) \tag{8}$$

where x_i denotes the original vectors, $\hat{x}_i$ is the decompressed vector, and 255 is a heuristic constant. Note that the denominator of the argument of the $\log_{10}\ (\cdot)$ is the average of the difference, or called error, between the original data set and the decompressed data given by the eBAM method.

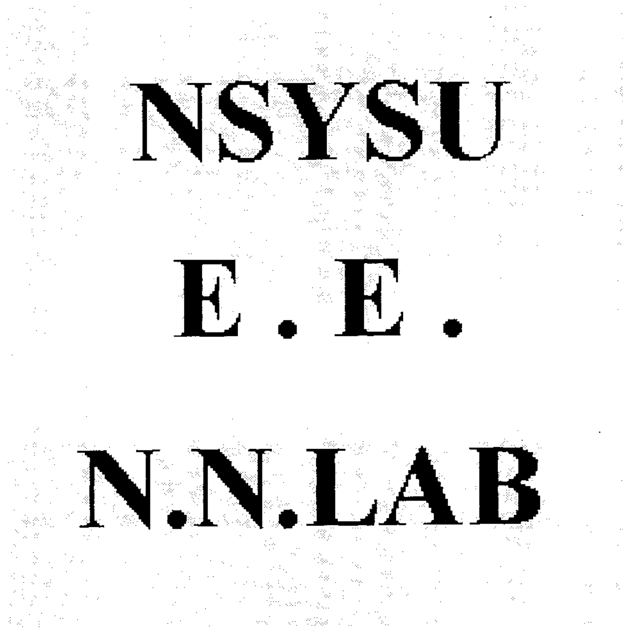

Fig. 2. The original picture.

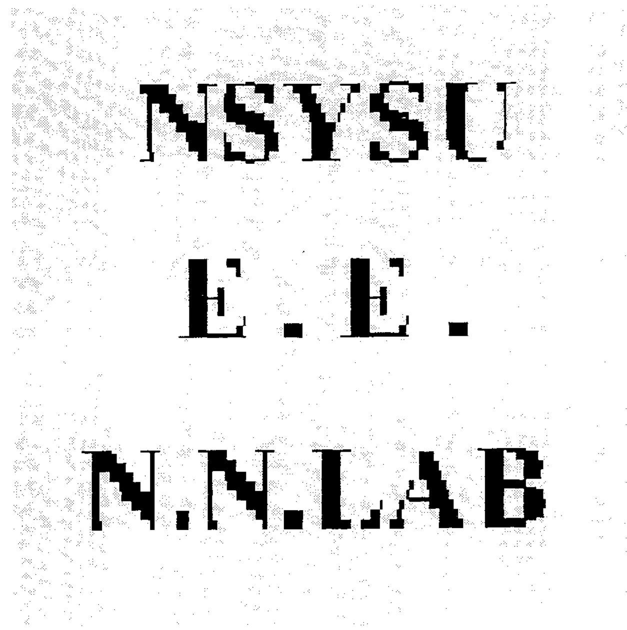

Fig. 3. eBAM method ($BK = 4,\ N' = 8$).

Hence, if $\hat{x}_i = x_i, \forall\, i$ it means a perfect recall of the original data after the compression–decompression. Then, the SNR is approaching 100. Otherwise, if the average of the difference is large, the SNR will be dramatically reduced. It is a nice index to compare the performance of compression/decompression methods.

III. Simulation Analysis

In order to show the performance of the eBAM recursive compression/decompression algorithm, we have done several simulations which will be compared with a standard VQ algorithm [8], [9]. In the following, we have conducted a series of simulation to illustrate the effect of the eBAM algorithm.

Example 1: A 256 $\times$ 256 original picture is shown in Fig. 2. Since the picture occupies a lot of memory, it is necessary to quantize the picture into small blocks which then will be compressed and transmitted. We define BK as the number of pixels of one side of the small blocks and L as the number of pixels of one side of the picture. Thus, the CDR and SNR defined, respectively, in (7) and (8) can be rewritten as

$$\mathrm{CDR} = \frac{BK \times BK}{\log_2\ N' + \dfrac{N' \times (BK^2 + \log_2\ N')}{L^2/BK^2}} \tag{9}$$

$$\mathrm{SNR} = 10 \times \log_{10}\left(\frac{255^2}{\dfrac{\sum_i (x_i - \hat{x}_i)(x_i - \hat{x}_i)^t}{D}}\right). \tag{10}$$

Note that the function of the codebook in the standard VQ method is similar to that of the feature vectors table. The size of the codebook and the vector table will affect the performance of data compression. Figs. 3–6 show the results of using eBAM algorithm with $N' = 8$, 32, 64, 128, respectively. In these pictures, the chosen block size is $BK = 4$ which means the dimension of the feature vectors and the data vectors are 16. Thus, $n = 16$ and $p \geq \log_2\ N'$. In our simulations, $p = \log_2\ N'$ is chosen. It is clear that when the number of the chosen feature vectors increases, the quality of the picture improves. However, Figs. 7 and 8 show a disappointing results of using the VQ method even though the size of the code book is as large as 128 and 256, respectively.

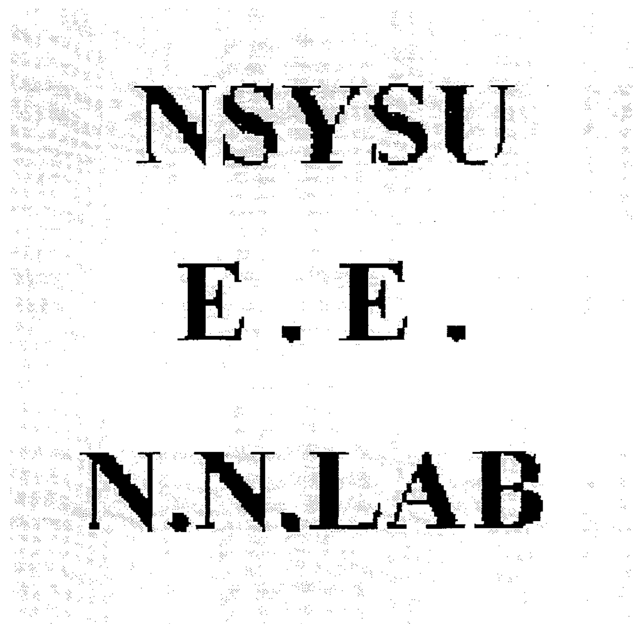

Fig. 4. eBAM method ($BK = 4,\ N' = 32$).

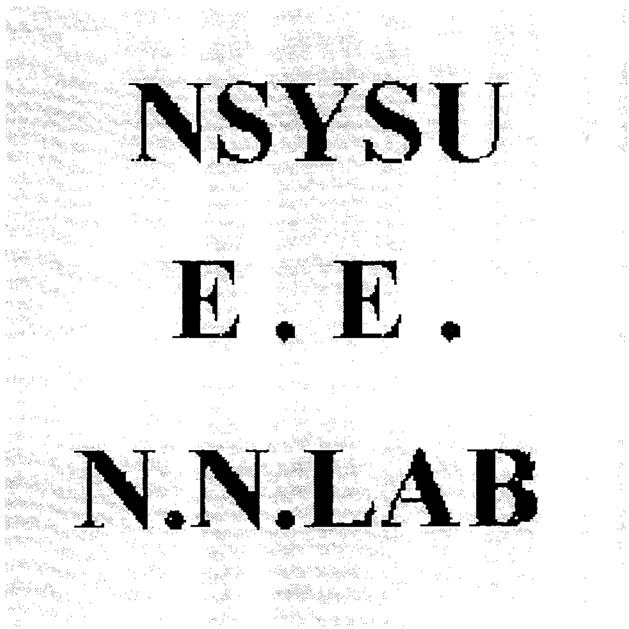

Fig. 5. eBAM method ($BK = 4,\ N' = 64$).

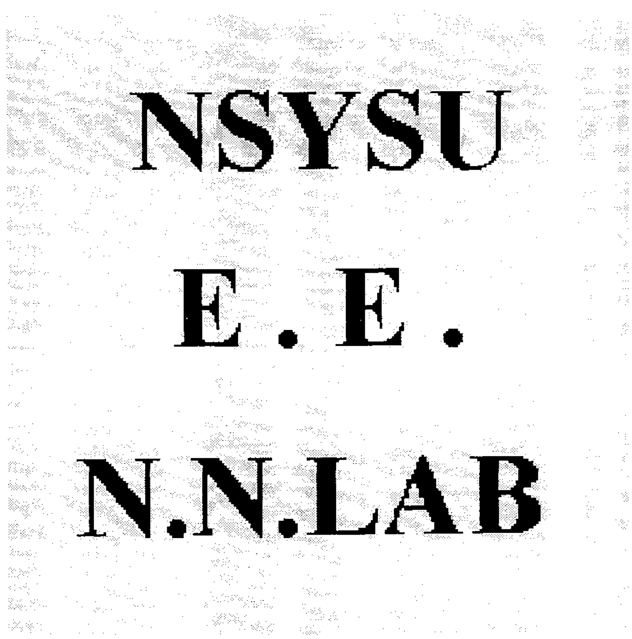

Fig. 6. eBAM method ($BK = 4,\ N' = 128$).

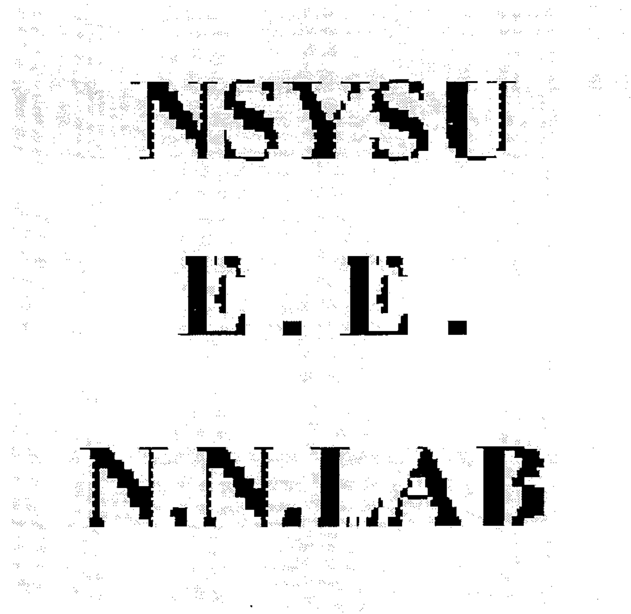

Fig. 7. VQ method ($BK = 4,\ codebook = 128$).

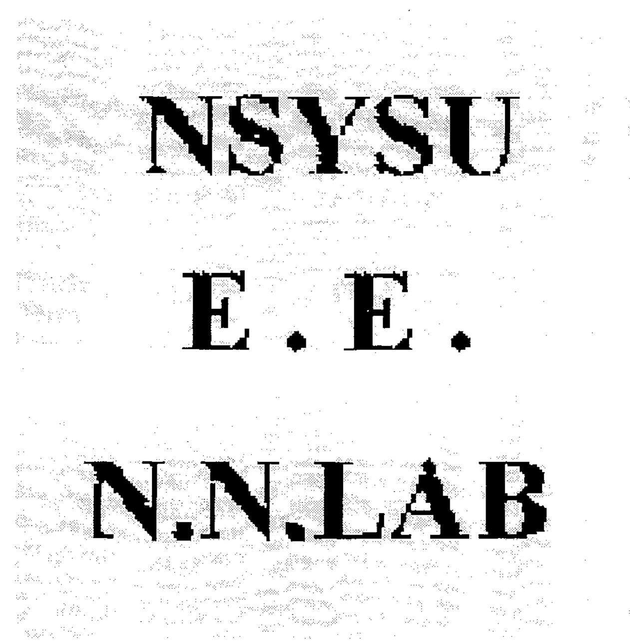

Fig. 8. VQ method ($BK = 4,\ codebook = 256$).

Now we enlarge the BK to be 8, which means $n = 64$ and repeat the same procedures. The results of the eBAM method are shown in Figs. 9–12. In contrast, the results of the VQ method are shown in Figs. 13 and 14. For the sake of clarity, the SNR of the eBAM methods are graphically illustrated in Fig. 15, while the VQ's is in Fig. 16. The overall SNR comparison of the mentioned methods is given in Fig. 17.

According to the above results, the eBAM algorithm not only shows a better SNR, but also uses a smaller table size. The detailed numerical data of CDR and SNR are listed in Tables I and II.

Note that if the eBAM approach is used, then the overall N is 150. That is the reason why there are no results for the cases $N' = 256$ and $N' = 512$. In short, the eBAM data compression/decompression has been proved to be feasible and powerful.

Example 2: We all understand that there is a possibility that the transmission channels are not noise free. It causes that the received bits might be wrong. Let's use the VQ to be an example, as shown in Fig. 18, the codebook $\text{size} = 128, BK = 4$. One bit error in every received tag vector will make the decompressed picture totally unreadable. On the contrary, if the eBAM method is used, a very noisy picture is

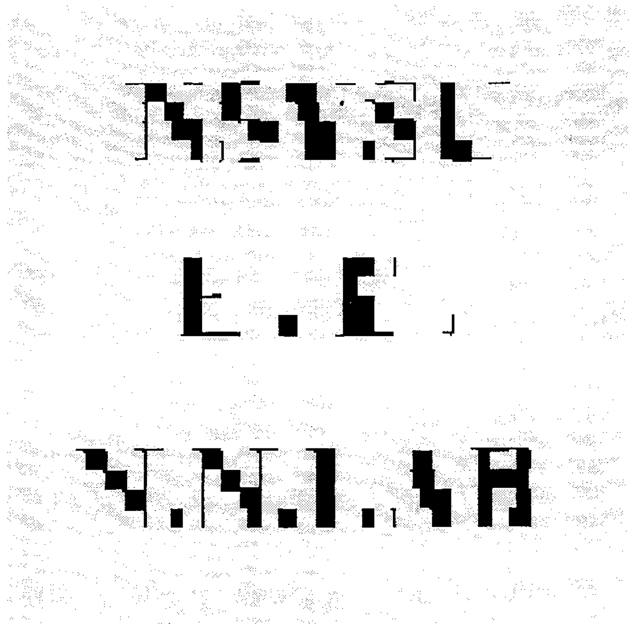

Fig. 9. eBAM method ($BK = 8,\ N' = 8$).

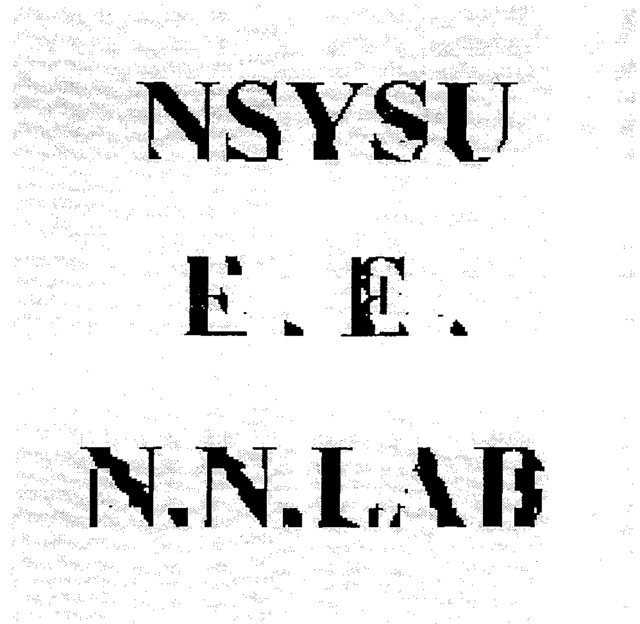

Fig. 11. eBAM method ($BK = 8,\ N' = 64$).

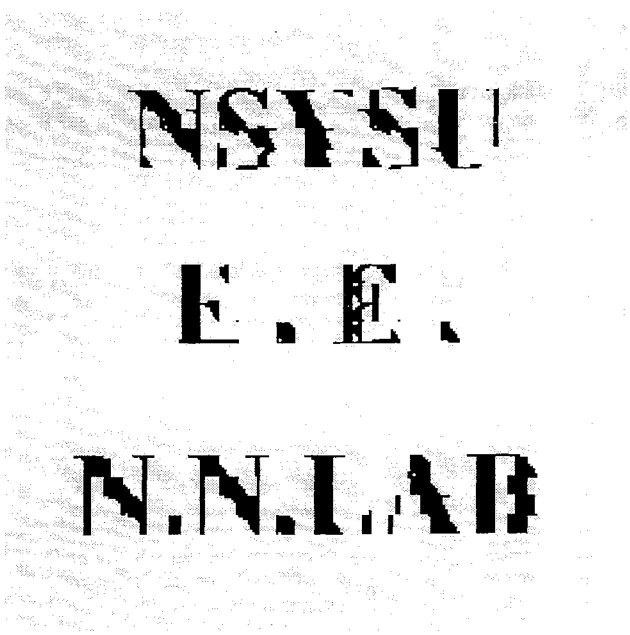

Fig. 10. eBAM method ($BK = 8,\ N' = 32$).

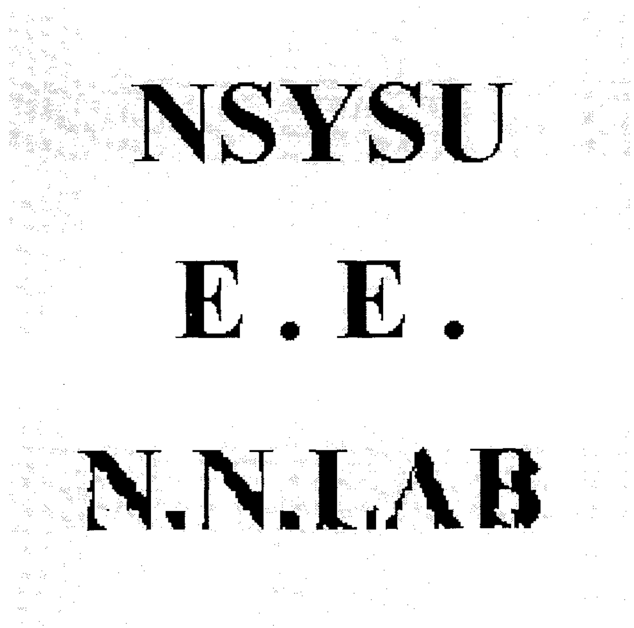

Fig. 12. eBAM method ($BK = 8,\ N' = 128$).

generated, which is barely readable, as shown in Fig. 19. This is because the fault tolerance ability, or called self-adjusting, of the eBAM. The remedy to get a clear picture without much loss of the fidelity is to increase the dimension of the tag vectors. In other words, add extra bits to the tag vectors to enhance the fault tolerance ability. A convincing demonstration is shown in Fig. 20. The fault tolerance ability of the eBAM method is tabulated in Table II.

The relationship between the number of added bits and the SNR is shown in Fig. 21. Note that the increase of number of bits of the tag vectors certainly enhance the fault tolerance ability and consequently the SNR. But on the other hand, the CDR will be drastically reduced.

IV. Conclusion

We have shown the superior SNR and CDR of the eBAM data compression algorithm. The algorithm is suitable for various kinds of data compression/decompression applications, though we use the image compression as an illustration. It is noted that the SNR mainly depends on the choice of N' the number of chosen feature vectors. In addition to the better CDR and SNR, the eBAM algorithm also shows the ability to overcome the bit error problem caused in a lossy channel. The eBAM algorithm will be extended to colored pictures and grey-level pictures by using a multivalued eBAM (MV-eBAM) [17] as the compression mechanism.

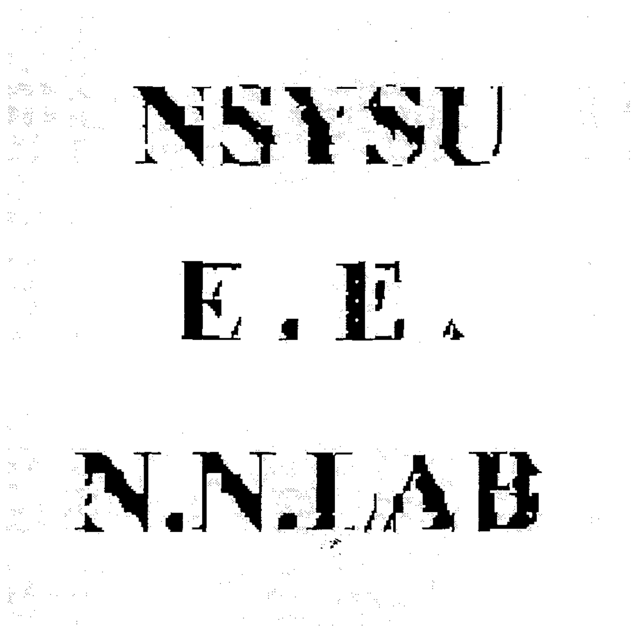

Fig. 13. VQ method ($BK = 8,\ codebook = 128$).

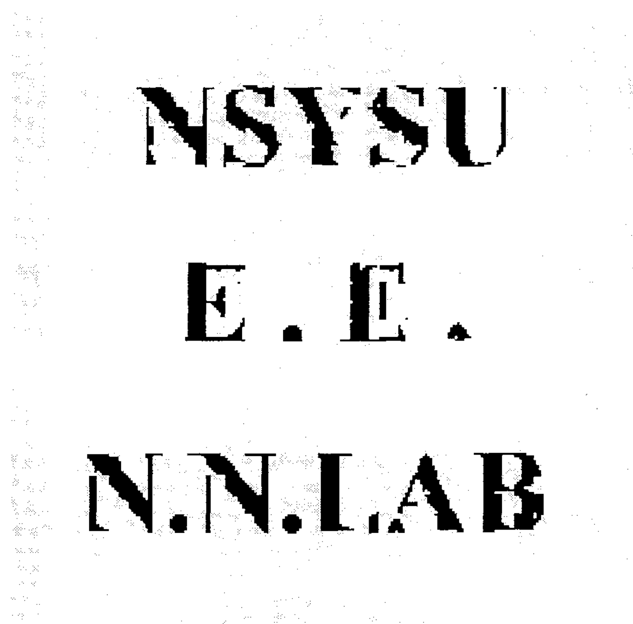

Fig. 14. VQ method ($BK = 8,\ codebook = 256$).

APPENDIX

Theorem 1: Given a attraction radius, r where $r < (n/2)$ the maximal capacity for an eBAM to store pattern pairs is

$$M < 1 + \frac{2^{(n-2)}}{(n-r-1) \cdot b^{4(r-2)}}. \tag{11}$$

Proof: According to (2)

$$y_k = \operatorname{sgn}\left\{ b^{(n-2r)} \cdot y_{hk} + \sum_{i \neq h}^{M} y_{ik} \cdot b^{X_i \cdot X} \right\}. \tag{12}$$

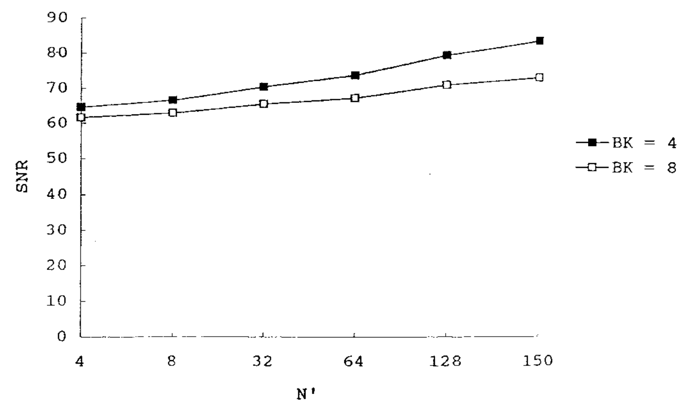

Fig. 15. The SNR of the eBAM method.

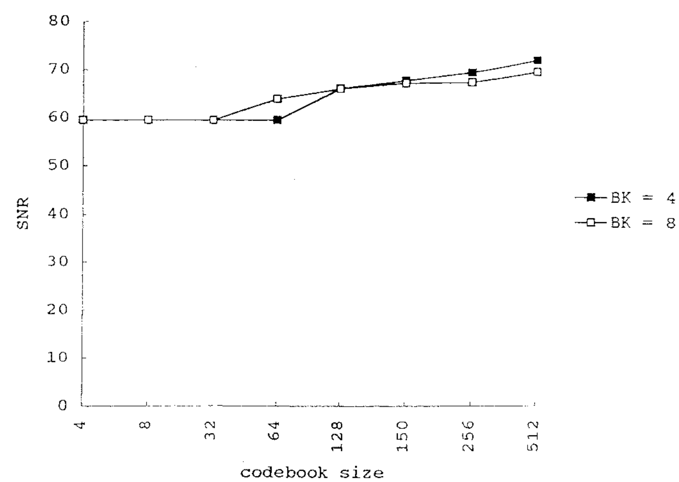

Fig. 16. The SNR of the VQ method.

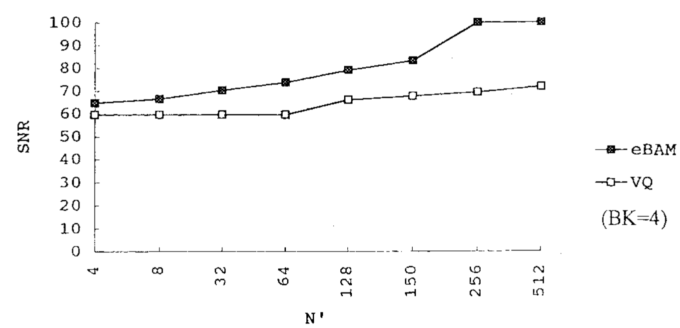

Fig. 17. The SNR of the eBAM method versus the VQ method.

TABLE I
COMPARISON OF THE SNR OF THE eBAM METHOD AND THE VQ

N' or codebook	4	8	32	64	128	150	256	512
eBAM ($BK = 4$)	64.56	66.52	70.33	73.67	79.22	83.29	100.0	100.0
eBAM ($BK = 8$)	61.67	62.94	65.47	67.13	70.83	72.91	100.0	100.0
VQ ($BK = 4$)	59.43	59.43	59.43	59.43	66.04	67.70	69.38	72.90
VQ ($BK = 8$)	59.43	59.43	59.43	63.90	65.99	67.18	67.31	69.47

We only discuss the Y part of evolution equations without any loss of robustness here. The power of the signal, i.e., the

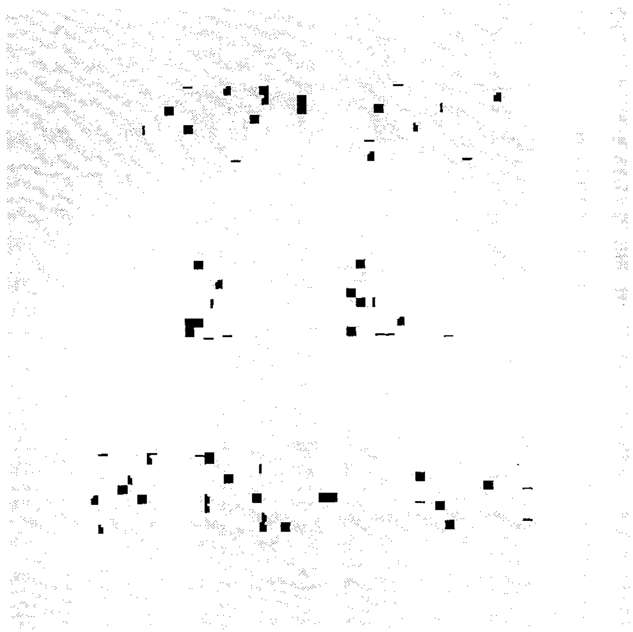

Fig. 18. The VQ method result with one-bit error ($BK = 4,\ codebook = 128$).

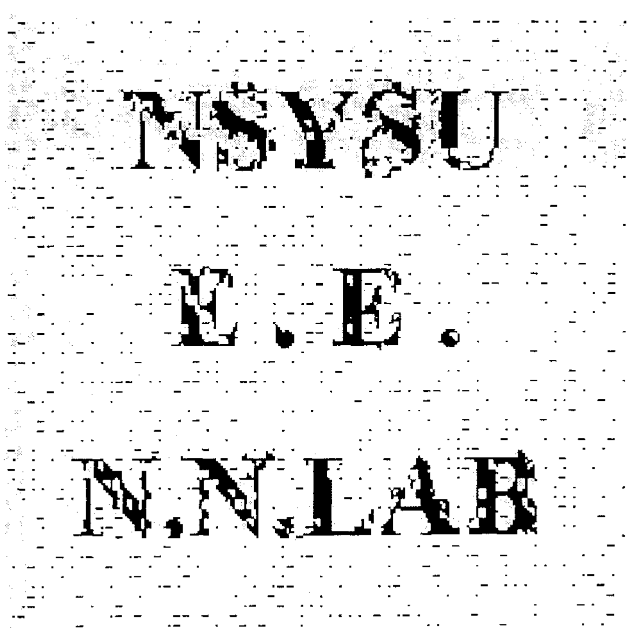

Fig. 19. The eBAM method result with one-bit error ($BK = 4,\ N' = 128$, no added bits).

first term in the right hand side of (12) is

$$S = (b^{(n-2r)})^2 = b^{(2n-4r)}. \quad (13)$$

On the other side, the noise term of (12) can be deemed as $(M - 1)$ random variables,

$$\begin{aligned} v_1 &= y_{1j} b^{X_1 \cdot X}, \\ v_2 &= y_{2j} b^{X_2 \cdot X}, \\ &\vdots \\ v_N &= y_{Nj} b^{X_N \cdot X} \end{aligned}$$

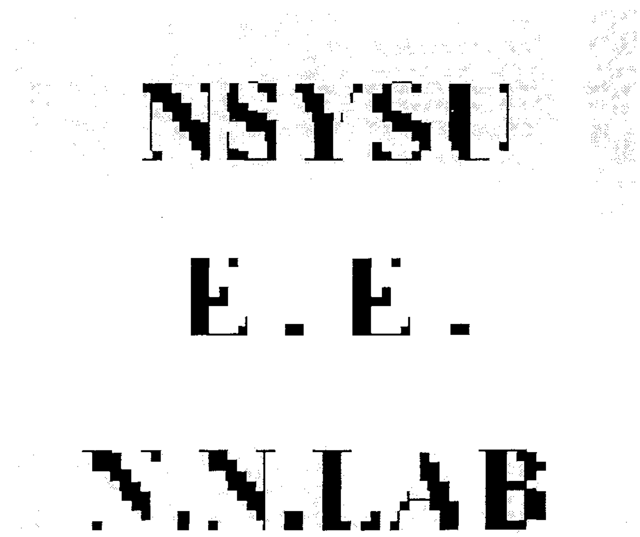

Fig. 20. The eBAM method result with one-bit error ($BK = 4,\ N' = 128$, added bits $= 4$).

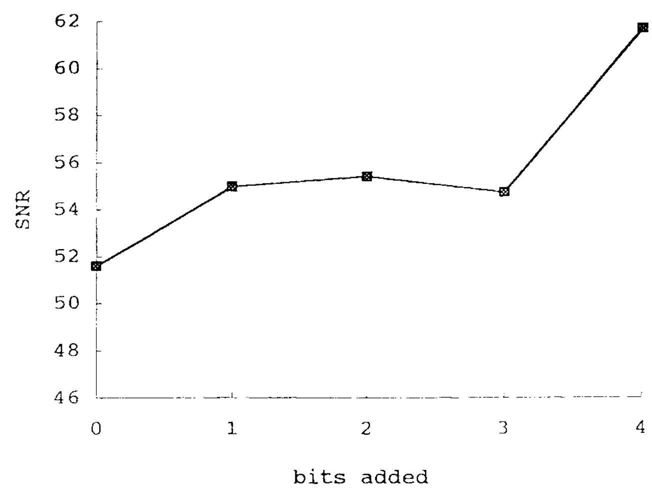

Fig. 21. The relationship of SNR and the added bits ($BK = 4, N' = 128$).

TABLE II
THE FAULT TOLERANCE ABILITY OF THE eBAM METHOD ($BK = 4N' = 128$)

bits added	0	1	2	3	4
CDR	2.000	1.778	1.600	1.455	1.333
SNR	51.58	54.96	55.40	54.72	61.67

where all of the $v_i, i \neq h$ are the identical random variables. We can take v_1 to compute the mean and the variance

$$Pr(v_1 = +1 \cdot b^{n-2-2k}) = (\tfrac{1}{2})^{n-1} C_k^{n-1} \quad (14)$$

$$Pr(v_1 = -1 \cdot b^{n-2-2k}) = (\tfrac{1}{2})^{n-1} C_k^{n-1} \quad (15)$$

where $k \geq r + 1$ and k is the Hamming distance between X and X_1. According to (15), we certainly get

$$E\{v_1\} = 0. \quad (16)$$

Then, the variance can be derived as

$$\begin{aligned} E[v_1^2] &= 2 \sum_{k=r+1}^{n-1} b^{2(n-2k-2)}(\tfrac{1}{2})^{n-1}C_k^{n-1} \\ &= 2 \sum_{k=r+1}^{m} b^{2(m-2k-1)}(\tfrac{1}{2})^{m}C_k^{m}, \\ &\quad \text{where} \quad m = n-1 \\ &= 2\, (\tfrac{1}{2})^m b^{2(m-1)} \sum_{k=r+1}^{m} (b^{-4})^k 1^{(m-k)} C_k^m. \end{aligned} \tag{17}$$

However, the above equation will not have a close form solution mainly because of the summation term. We can try to find the upper bound of this summation term. Let $a = b^{-4}$ to make the equations easy to read. Then, the summation term in (17) can be rewritten as shown at the bottom of the page.

Hence, (17) can be derived as

$$\begin{aligned} E[v_1^2] &< 2(\tfrac{1}{2})^m b^{2(m-1)}[(1+b^{-4})^m - (1+b^{-4})^r] \\ &= 2\,(\tfrac{1}{2})^{(n-1)} b^{2(n-2)}[(1+b^{-4})^{(n-1)} - (1+b^{-4})^r] \end{aligned} \tag{18}$$

The above upper bound is the maximal noise power, called $N_{\max}$. Then the minimal signal-to-noise-ratio $(\mathrm{STNR}_{\min})$ of the eBAM is shown in (19) at the bottom of the page.

The $\mathrm{STNR}_{\min}$ must be greater than one in order to recall the correct pattern pair. Hence,

$$\begin{aligned} \mathrm{STNR}_{\min} &> 1 \\ b^4 \cdot 2^n &> 4(M-1)\cdot b^{4r}\cdot[(1+b^{-4})^{(n-1)} - (1+b^{-4})^r] \\ &\approx 4(M-1)\cdot b^{4r}\cdot[(1+(n-1)\cdot b^{-4}) \\ &\quad - (1+r\cdot b^{-4})] \\ &= 4(M-1)(n-r-1)\cdot b^{4(r-1)} \\ 2^n &> 4(M-1)(n-r-1)\cdot b^{4(r-2)} \\ M &< 1 + \frac{2^{(n-2)}}{(n-r-1)\cdot b^{4(r-2)}} = M_{\max}. \end{aligned} \tag{20}$$

Theorem 2: The bit-error probability of the eBAM is

$$\begin{aligned} P_e &\approx \frac{1}{\sqrt{2\pi}} \cdot \frac{(M-1)^{1/2} 2^{(-(n/2)+1)}(n-r-1)^{1/2}}{b^{(4-2r)}} \\ &\quad \cdot \exp\left[-\frac{b^{(8-4r)}}{2(M-1)\cdot 2^{(-n+2)}\cdot(n-r-1)}\right]. \end{aligned} \tag{21}$$

Proof: y_{hk} in (12) can be assumed to be -1 without loss of robustness. Then, the error occurs when the argument in the sgn function of (12) turns out to greater than 0. That is,

$$\begin{aligned} P_e &= \mathrm{Prob}(V > 0), \\ V &= -b^{n-2r} + \sum_{i \neq h}^{M} y_{ik} \cdot b^{X_i \cdot X} \\ V &= -b^{n-2r} + v. \end{aligned}$$

$$\begin{aligned} \sum_{k=r+1}^{m} a^k \cdot C_k^m &= \sum_{k=0}^{m} a^k \cdot C_k^m - \sum_{k=0}^{r} a^k \cdot C_k^m \\ &= (1+a)^m - \sum_{k=0}^{r} a^k \cdot \frac{m!}{(m-k)!\cdot k!} \\ &= (1+a)^m - \sum_{k=0}^{r} a^k \cdot \frac{m(m-1)\cdots(r+1)\cdot r!}{[(m-k)(m-k-1)\cdots(r+1-k)\cdot(r-k)!]\cdot k!} \\ &= (1+a)^m - \sum_{k=0}^{r} a^k \cdot \left(\frac{m}{m-k}\right)\left(\frac{m-1}{m-k-1}\right)\cdots\left(\frac{r+1}{r+1-k}\right)\cdot\frac{r!}{(r-k)!\cdot k!} \\ &< (1+a)^m - \sum_{k=0}^{r} a^k \cdot C_k^r \\ &= (1+a)^m - (1+a)^r \end{aligned}$$

$$\begin{aligned} \mathrm{STNR}_{\min} &= \frac{S}{(M-1)N_{\max}} = \frac{b^{(2n-4r)}}{(M-1)\cdot 2(\tfrac{1}{2})^{(n-1)} b^{2(n-2)}[(1+b^{-4})^{(n-1)} - (1+b^{-4})^r} \\ &= \frac{b^4 \cdot 2^n}{4(M-1)\cdot b^{4r}\cdot[(1+b^{-4})^{(n-1)} - (1+b^{-4})^r} \end{aligned} \tag{19}$$

Again we assume the summation term in the above equation is the sum of $(M-1)$ identical random variables as described in the proof of Theorem 1. We also take v_1 as an example to compute the expectation value and the variance of each random variable. According to (16) and (18)

$$\begin{aligned}\text{Var }\{v_1\} &= E\{v_1^2\} - E^2\{v_1\}\\ \sigma_{v_1} &= \sqrt{\text{Var}\{v_1\}}\\ &< \sqrt{2(\tfrac{1}{2})^{(n-1)} b^{2(n-2)}[(1+b^{-4})^{(n-1)} - (1+b^{-4})^r]}\\ &\approx 2^{(-(n/2)+1)} b^{(n-2)} \sqrt{(n-r-1)\cdot b^{-4}}\\ &= 2^{(-(n/2)+1)} b^{(n-4)} (n-r-1)^{1/2}.\end{aligned}$$

Since the overall noise v is the sum of $(M-1)$ identical random variables, e.g., v_1 we surely have the following conclusion:

$$\begin{aligned}\text{Var}\{v\} &= (M-1)\cdot\text{Var}\{v_1\}\\ \sigma_v &= (M-1)^{1/2}\sigma_{v_1}.\end{aligned}$$

Henceforth, the error of the recall process occurs when $v > b^{n-2r}$ in (12). Basing upon the central limit theorem, we should have the following result when $n\to\infty, M\to\infty$

$$P_e = \text{Prob}(V>0) = \text{Prob}(v > b^{n-2r}) = Q\left(\frac{b^{n-2r}}{\sigma_v}\right) \tag{22}$$

where

$$\begin{aligned}Q(t) &= \frac{1}{\sqrt{2\pi}}\int_t^{\infty} e^{-(x^2/2)}\,dx\\ &\approx \frac{1}{\sqrt{2\pi}}\cdot t^{-1}\cdot e^{-(t^2/2)}.\end{aligned}$$

Hence, the bit-error probability P_e is defined as

$$\begin{aligned}P_e \approx \frac{1}{\sqrt{2\pi}}&\cdot\frac{(M-1)^{(}1/22^{(-(n/2)+1)}(n-r-1)^{1/2}}{b^{(4-2r)}}\\ &\cdot\exp\left[-\frac{b^{(8-4r)}}{2(M-1)\cdot 2^{(-n+2)}\cdot(n-r-1)}\right].\end{aligned} \tag{23}$$

REFERENCES

[1] G. R. De Haan and Ö. Eğecioğlu, "Links between self-organizing feature maps and weighted vector quantization," in *Int. Joint Conf. Neural Networks*, 1991, pp. 887–892.

[2] T. M. E. Frost and C. J. Theaker, "Real-time video data compression system," *Proc. Inst. Elect. Eng.*, vol. 137, pt. E, Sept. 1990.

[3] D. Le Gall, "MPEG: A video compression standard for multimedia applications," *Commun. ACM*, vol. 34, pp. 47–58, Apr. 1991.

[4] H.-C. Huang, J.-H. Huang, and J.-L. Wu, "Real-time software-based video coder for multimedia communication systems," in *Proc. ACM Multimedia 93*, Aug. 1993, pp. 65-73.

[5] Y.-J. Jeng, C.-C. Yeh, and T.-D. Chiueh, "Exponential bidirectional associative memories," *Electron. Lett.*, vol. 26, pp. 717–718, May 1990.

[6] B. Kosko, "Adaptive bidirectional associative memory," *Appl. Opt.*, vol. 26, pp. 4947–4960, Dec. 1987.

[7] B. Kosko, "Bidirectional associative memory," *IEEE Trans. Syst., Man, Cybern.*, vol. 18, pp. 49–60, Jan./Feb. 1988.

[8] H. Ko and R. H. Baran, "Data compression and novelty filtering in retinotopic," in *Int. Joint Conf. Neural Networks*, 1991, pp. 2502-2507.

[9] J. Liu and D. Wang, "Data compression for image recognition using neural network," in *Int. Joint Conf. Neural Networks*, vol. 4, pp. 333–338, June 1992.

[10] K. Patel, B. C. Smith, and L. A. Rowe, "Performance of a software MPEG video decoder," in *Proc. ACM Multimedia 93*, Aug. 1993, pp. 75–82.

[11] G. D. Riplay, "DVI-a digital multimedia technology," *Commun. ACM*, vol. 32, pp. 811–822, July 1989.

[12] F. Sijstermansand and J. van der Meer, "CD-I full-motion video encoding on a parallel computer," *Commun. ACM*, vol. 34, pp. 82–91, Apr. 1991.

[13] S. Thacore, V. Pang, and M. Palaniswami, "Image data compression using a self-organizing neural network with adaptive thresholds," *Int. Joint Conf. Neural Networks*, 1991, pp. 646–651.

[14] G. K. Wallace, "The JPEG still picture compression standard," *Commun. ACM*, vol. 34, pp. 31–44, Apr. 1991.

[15] C.-C. Wang and H.-S. Don, "An analysis of high-capacity discrete exponential BAM," *IEEE Trans. Neural Networks*, vol. 6, pp. 492–496, Mar. 1995.

[16] C.-C. Wang, and J.-P. Lee, "The decision making properties of discrete multiple exponential bidirectional associative memories," *IEEE Trans. Neural Networks*, vol. 6, pp. 993–999, July 1995.

[17] C.-C. Wang, S.-M. Hwang, and J.-P. Lee, "Capacity analysis of the asymptotically stable multi-valued exponential bidirectional associative memory," *IEEE Trans. Syst., Man, Cybern. B*, vol. 26, pp. 733–743, Oct. 1996.

Chua-Chin Wang was born in Taiwan, R.O.C., in 1962. He received the B.S. degree in electrical engineering from National Taiwan University, Taipei, in 1984, and the M.S. and Ph.D. degrees in electrical engineering from the State University of New York, Stony Brook, in 1988 and 1992, respectively.

He is currently an Associate Professor in the Department of Electrical Engineering, National Sun Yat-Sen University, Taiwan. His recent research interests include neural networks and implementations, low-power logic and circuit design, and VLSI design.

Chang-Rong Tsai was born in Peng-Hu, Taiwan, R.O.C., in 1971. He received the B.S. degree in information engineering from Chung-Yuan University in 1992 and the M.S. degree in electrical engineering from National Sun Yat-Sen University in 1994.

His research areas include neural networks, data compression, and LAN architectures. He is currently an R&D Engineer with Best System Inc., Taiwan.