A Competitive Learning Algorithm with Controlling Maximum Distortion

Takeshi Miura Kentaro Sano[†] Kenichi Suzuki Tadao Nakamura

Graduate School of Information Sciences, Tohoku University
01 Aramaki Aza Aoba, Aoba-ku, Sendai, 980-8579, Japan
email:{john, kentah, suzuki, nakamura}@archi.is.tohoku.ac.jp

Abstract— **Vector quantization with an optimal codebook is attractive for lossy data compression. So far, a number of codebook design algorithms have been proposed to minimize a mean square error, MSE. However, these algorithms have a problem that MSE minimization sometimes causes a severe maximum-distortion, which is very important in several applications. This paper proposes a competitive learning algorithm with controlling maximum distortion that designs a codebook giving a maximum distortion less than a given error bound. The proposed algorithm assigns a code vector to an input vector with a too large distortion. Experimental results showed that the algorithm has maximum-distortion control capability with an MSE minimized.**

Keywords – vector quantization, optimal codebook design, competitive learning, maximum-distortion control

I. Introduction

Vector quantization(VQ)[1] is one of the effectual techniques for lossy data compression, and therefore, many researchers have so far applied VQ to compression of audio, image, movie and volume data [2]–[8]. According to Shannon's rate distortion theory, VQ can achieve better rate-distortion performance than other approaches using scalar quantization(SQ). VQ exploits a larger amount of coherence in data than SQ by binding scalars into a vector.

A vector quantizer maps vectors in k-dimensional Euclidean space $\boldsymbol{R}^k$ into a finite number of representative vectors in $\boldsymbol{R}^k$. Such representative vectors are referred to as *code vectors*, and a set of code vectors is referred to as a *codebook*. A vector quantizer is specified by a codebook $\boldsymbol{Y} = \{\boldsymbol{y}_1, \boldsymbol{y}_2, ..., \boldsymbol{y}_N\}$ and its associated partition $\boldsymbol{S} = \{S_1, S_2, ..., S_N\}$ that divides the input vector space into N disjoint regions. Given a set of input vectors, it is quantized by replacing all input vectors in S_i with $\boldsymbol{y}_i$. This replacement is called *reproduction*, and expressed as:

$$Q(\boldsymbol{x}) = \boldsymbol{y}_i \quad \text{if} \;\; \boldsymbol{x} \in S_i \quad \text{for} \;\; i = 1, 2, ..., N \tag{1}$$

where $\boldsymbol{x} \in \boldsymbol{R}^k$ is a k-dimensional input vector.

Reproduction causes a quantization error, called *distortion*. The distortion caused by reproducing $\boldsymbol{x}$ with $\boldsymbol{y}$, $d(\boldsymbol{x}, \boldsymbol{y})$, is commonly defined as the r-th power of the Euclidean norm, $\|\boldsymbol{x} - \boldsymbol{y}\|^r$. In this paper, we use distortions with r = 2. So far, most of VQ studies have aimed at minimization of a mean square error (MSE). Let $p(\boldsymbol{x})$ be probability distribution of $\boldsymbol{x}$. The MSE is defined as:

$$MSE(Q) = \sum_{i=1}^{N} \int_{S_i} d(\boldsymbol{x}, \boldsymbol{y}_i) p(\boldsymbol{x}) d\boldsymbol{x} \tag{2}$$

To obtain a vector quantizer minimizing the MSE, we have to find an optimal codebook $\boldsymbol{Y}$ and partition $\boldsymbol{S}$ in regard to the MSE. There are the following two conditions necessary to obtain such an optimal codebook[9].

Conditoin 1. Voronoi partition: For given code vectors, the partition has to be the Voronoi partition so that:

$$\boldsymbol{x} \in S_i, \quad \text{if} \;\; d(\boldsymbol{x}, \boldsymbol{y}_i) \leq d(\boldsymbol{x}, \boldsymbol{y}_j), \quad \text{for all} \;\; j. \tag{3}$$

Condition 2. Generalized centroid assignment: For a given partition, the code vectors have to be the generalized centroid so that:

$$E\left[d(\boldsymbol{x}, \boldsymbol{y}_i) | \boldsymbol{x} \in S_i\right] = \inf_{u} E\left[d(\boldsymbol{x}, \boldsymbol{u}) | \boldsymbol{x} \in S_i\right]. \tag{4}$$

For $r = 2$, the centroid of S_i is calculated as:

$$\boldsymbol{y}_i = E\left[\boldsymbol{x} | \boldsymbol{x} \in S_i\right]. \tag{5}$$

So far, codebook design methods have been proposed to obtain a codebook satisfying these conditions.

Although the conventional methods minimizing the MSE are good at preserving feature of data on average, they are likely to give several quantized vectors with severe distortions. Even though there are a few input vectors that have very characteristic information of the data set, these input vectors are not particularly preserved, resulting in quantized data losing their original characteristics. In the case of image compression application based on VQ, an image is divided into n^2-pixel blocks to form n^2-dimensional input vectors, which are quantized with a designed codebook. The conventional method spoils characteristic blocks, e.g. blocks containing letters and/or distictive patterns like edges, because a small number of these block do not influence an MSE much. Thus, MSE minimization only restricts degradation of overall image quality, not local features concerning perceptual errors.

In a codebook with a minimized MSE, each code vector has an expected value of distortions that is in inverse proportion to the probability density of input vectors around the code vector. Therefore, sparse input vectors,

[†]Corresponding author.

which are in regions of low probability density, are likely to have large distortions in minimizing an MSE. WCNN algorithm[10] was proposd to even out distortions of input vectors, instead of minimizing an MSE. WCNN algorithm is an extended version of competitive learning algorithms for uniform distortions. Within this algorithm, probability density of input vectors are approximately estimated by obtaining statistical distribution of input-vector's norms. Then, the WCNN algorihm reduces large distortions due to low probability by competition using distortions weighted with the reciprocal of the estimated probability.

However, WCNN algorithm has the following disadvantages. Firstly, the approximated estimation based on norms gives inaccurate probability of input vectors, resulting in imperfectly uniformed distortions. Secondly, distortions are unnecessarily evened out giving undesirable increase of MSE. Users would sometimes like just to reduce the maximum distortion, instead of evening out the whole distortions.

This paper proposes a competitive lerning algorithm with controlling maximum distortion that designs a codebook with the maximum distortion less than a given error bound. This algorithm is an extension of a competitive learning algorithm that is introduced a rearrangement mechanism of code vectors. The algorithm also minimizes an MSE while the maximum distortion is controlled.

This paper is organized as follows. Section II briefly reviews conventional competitive learning algorithms and their problem . Section III describes the proposed algorithm. Section IV discusses the performance of the proposed algorithm through experimental results. Finally, Section V gives conclusions and future work.

II. Conventional algorithms for codebook design

A. Basic competitive learning algorithms

A competitive learning algorithm is one of the most famous and widely used algorithms for codebook design minimizing an MSE. A competitive learning algorithm basically repeats a competition process to update a codebook. A winner code vector of competition for a given input vector is moved toward the input. Kohonen competitive learning algorithm is a basic one with a *winner-takes-all* rule[11].

Within Kohonen learning algorithm, learning proceeds as follows. Before learning, codebook $\boldsymbol{Y} = \{\boldsymbol{y}_1, \boldsymbol{y}_2, ..., \boldsymbol{y}_N\}$ is initialized with some distribution of vectors. Then the following learning steps are repeated for input vectors. For input vector $\boldsymbol{x}_k$, such code vector $\boldsymbol{y}_{i^*}$ is found that $\boldsymbol{x}_k \in S_{i^*}$ where S_{i^*} is the Voronoi region of $\boldsymbol{y}_{i^*}$. This code vector with the smallest distortion for the input vector is referred to as a *winner*. The process to find a winner is called *competition* or *winner selection*. Next, winner $\boldsymbol{y}_{i^*}$ is updated based on:

$$\boldsymbol{y}_{i^*} := \boldsymbol{y}_{i^*} + \alpha(\boldsymbol{x}_k - \boldsymbol{y}_{i^*}) \tag{6}$$

where α is a learning rate, which is a positive constant for codebook adaptability to inputs.

By repeating the above learning steps, code vectors are generally approximating the probability distribution of input vectors with small distortion. However, Kohonen learning has an inherent problem, i.e., the under-utilization problem. Inadequate distribution of initial code vectors gives several code vectors extremely distant from any inputs, which cannot be utilized for quantization. Such code vectors have no opportunity to be moved by input vectors. As a result, MSE is not reduced due to the local minima given by under/never-utilized code vectors. So far, various extended competitive learning algorithms have been proposed to avoid the local minima[12][13].

B. Issues accompanying MSE minimization

According to the partial distortion theorem[13], each region S_i makes an equal contribution to the distortion for an optinal quantizer with a large number of code vectors.

$$D_i = E[d(\boldsymbol{x}, \boldsymbol{y}_i)|\boldsymbol{x} \in S_i] \cdot P_i \approx constant \tag{7}$$

where D_i is called the partial distortion in region S_i for code vector $\boldsymbol{y}_i$. P_i is the probability that $\boldsymbol{x} \in S_i$. Eq.(7) means that the expected value of distortions in S_i, that is D_i/P_i, is in inverse proportion to P_i in the case of an optimal codebook minimizing an MSE. Since input vector distribution is generally not uniform, there exists differrence in the expected distortion among code vectors. Naturally, there also exists difference in a distortion of each input vector. The region with a large expected distortion is likely to contain input vectors with large distortions, and vice versa.

In the codebook design minimizing an MSE, the smaller probability density of input vectors results in the larger distortion. The region with the lowest probability density is likely to give an input vector the maximum distortion. Thus, a distortion of each input vector is not controlled at all, but just given according to its probability density. However, the maximum distortion is also important in some applications, and one of the measures of VQ quality.

Several applications require low mamixum-distortion. Let's suppose that VQ is applied to image compression. In this case, firstly the original image is divided into n^2-pixel blocks to form n^2-dimensional input vectors, which are quantized with a designed codebook. If a codebook designed with MSE minimization is used, each block (input vector) has a different distortion. While some blocks are very similar to their original blocks, and some blocks are very different from the original ones. This feature of MSE minimization can cause extreme corruption for a few but very characteristic and important blocks, e.g. blocks containing letters and/or distictive patterns in an image. The larger maximum-distortion an image has, the more serious perceptual error we often feel as shown in Fig.1.

These characteristic blocks (or input vectors) are likely to have maximum distortions because of their small probability. In order to guarantee acceptable quality for these characteristic blocks, the maximum distortion should be controlled less than an error bound given by users. This is

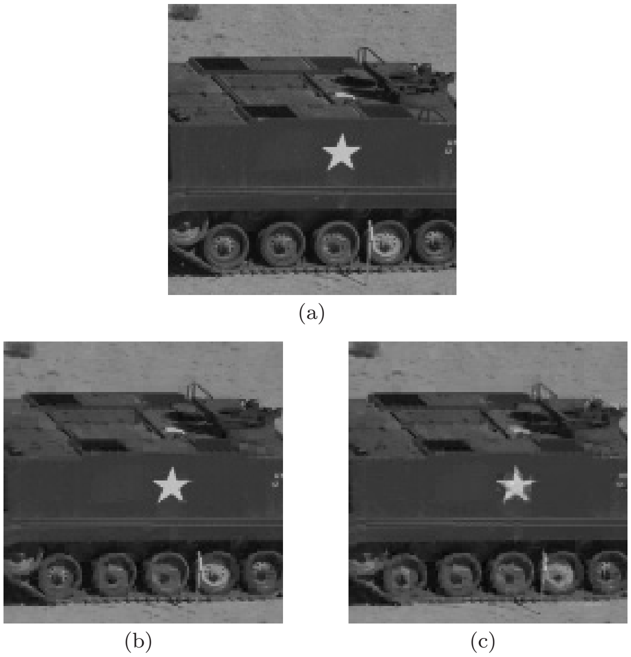


Fig. 1. (a) Orignal image (APC). (b) Reconstructed image with $MSE = 285.87$, maximum distortion $= 2637.48$. (c) Reconstructed image with $MSE = 272.52$, maximum distortion $= 17024.10$.

the motivation of our work. We extend a competitive learning algorithm so that the maximum distortion is controlled by users. To reduce the maximum distortion, several code vectors have to particularly be assigned to input vectors with distortions larger than the given error bound. On the other hand, an MSE should also be reduced in addition to the maximum-distortion control.

In this paper, we propose a competitive learning algorithm with maximum-distortion control. This algorithm performs the Kohonen learning with rearrangement of code vectors. When an input vector with a distortion larger than a given error bound is detected, a code vector is arranged at the input vector location to avoid the unacceptable distortion. The arranged code vector is selected so that the expected increase in MSE is smallest. As a result, users can control the maximum distortion with minimum MSE increase. The next section describes the proposed algorithm in detail.

III. Competitive learning algorithm with maximum-distortion control

A. Approach

As mentioned in Section II-B, our goal is to control the maximum distortion less than a given error bound for optimal codebook design. Simultaneously, we also have to consider MSE minimization while the maximum distortion is controlled. To achieve this goal, we extend the Kohonen competitive learning algorithm. Within the Kohonen learning, a winner of competition is moved toward the input vector. At the same time that the winner is found for the input vector, we can know the current distortion of the input vector. We use this online distortion of each input vector to control the maximum distortion.

When we detect an input vector with a distortion larger than a given error bound, such a distortion should be reduced to less than the bound. For this reduction, we take a means of specially assigning a code vector to the input vector. In order to keep the number of code vectors constant, one of the code vectors has to be selected and deleted simultaneously for the special assignment. This procedure of deletion and new assignment can be regarded as rearrangement of the selected code vector

Although the new assignment is effective for maximum-distortion control, MSE can be increased by deleting another code vector. To minimize MSE increase, the deleted code vector should carefully be selected. In the following subsections, we describe a method to select a code vector that has small contribution to MSE.

A.1 Code-vector selection for rearrangement

To restrict unnecessary increase of MSE, we delete a code vector with small contribution to MSE in the rearrangement process. We need a metric to evaluate the MSE contribution of each input vector.

When a code vector is removed, an MSE increases because input vectors to be approximated by the code vector have larger distortions to their secondly closest code vectors. This increase of MSE can be used to evaluate the MSE contribution of the code vector assumed to be removed. We define MSE contribution of a code vector as the degree of MSE increase assuming that the code vector is removed.

Now we give mathematical definition of MSE contribution. $Q(\boldsymbol{x})$ is a code vector that approximates input vector $\boldsymbol{x}$. Let $Q^*(\boldsymbol{x})$ denote a code vector that is secondly closest to $\boldsymbol{x}$. In other words, $Q^*(\boldsymbol{x})$ approximates $\boldsymbol{x}$ when $Q(\boldsymbol{x})$ is removed. MSE contribution of the i-th code vector, MC_i, is defined as the sum of distortion increases due to deletion of the code vector:

$$MC_i = \sum_{j=1}^{M} (d(\boldsymbol{x}_j, Q^*(\boldsymbol{x}_j)) - d(\boldsymbol{x}_j, Q(\boldsymbol{x}_j)))\delta_{i,\arg Q(\boldsymbol{x}_j)} \quad (8)$$

where $\delta_{i,j}$ is Kronecker's delta (i.e., $\delta_{i,j} = 1$ for $i = j$, and otherwise $\delta_{i,j} = 0$), and $\arg Q(\boldsymbol{x}_j)$ means the identification number of code vector $Q(\boldsymbol{x}_j)$. We select a code vector with the smallest MC for rearrangement.

To decide if rearrangement is actually performed by using the selected code vector, we have to take care of the maximum-distortion emerging when the selected code vector is deleted. Here, we also define maximum-distortion contribution of the i-th code vector, MDC_i, which is the biggest distortion concerning the code vector to be deleted.

$$MDC_i = \max_j d(\boldsymbol{x}_j, Q^*(\boldsymbol{x}_j))\delta_{i,\arg Q(\boldsymbol{x}_j)} \quad (9)$$

If the MDC of the selected code vector is larger than the distortion of the target input vector for rearrangement, the maximum distortion is not reduced by this rearrangement. On the contrary, the maximum distortion becomes larger. Therefore, we do not perform the rearrangement in this case.

TABLE I
COMPETITION HISTORY TABLE

iteration	winner index	1st distortion	2nd distortion
T	$\arg Q(\boldsymbol{x}(T))$	$d(T)$	$d^*(T)$
$T-1$	$\arg Q(\boldsymbol{x}(T-1))$	$d(T-1)$	$d^*(T-1)$
$T-2$	$\arg Q(\boldsymbol{x}(T-2))$	$d(T-2)$	$d^*(T-2)$
$\vdots$	$\vdots$	$\vdots$	$\vdots$
$T-L+1$	$\arg Q(\boldsymbol{x}(T-L+1))$	$d(T-L+1)$	$d^*(T-L+1)$

A.2 Online estimation method

Offline calculation of MC_i and MDC_i has a large overhead, and therefore requires long computing time. In order to avoid the overhead to obtain MC_i and MDC_i for each competition, we propose their online estimation method. For the online estimation, we use a competition history table shown in Table I. The competition history table has L lines, each of which contains three elements: a winner index, a distortion for the winner (1st distortion), and a distortion for the secondly closest code vector (2nd distortion). The t-th line corresponds to the input vector of the $(T-t+1)$-th iteration, $\boldsymbol{x}(T-t+1)$, where the T-th one is the current iteration. We estimate MC_i and MDC_i by using the history of the latest L iterations.

For concise description, let $d(t)$ and $d^*(t)$ denote $d(\boldsymbol{x}(t), Q(\boldsymbol{x}(t)))$ and $d(\boldsymbol{x}(t), Q^*(\boldsymbol{x}(t)))$, respectively. The recent MC_i can approximately be calculated by:

$$MC_i(T) = \sum_{t=1}^{L} (d^*(t) - d(t))\delta(i, \arg Q(\boldsymbol{x}(t))). \qquad (10)$$

Similarly, the recent MDC_i can approximately be calculated by:

$$MDC_i(T) = \max_{t=1}^{L} d^*(t)\delta(i, \arg Q(\boldsymbol{x}(t))). \qquad (11)$$

The competition history table can be implemented as a cyclic buffer. Since the sum for $MC_i(T)$ and the max-search for $MDC_i(T)$ can incrementally be performed, the overhead of this online estimation is very low.

B. Procedure of the proposed algorithm

Our proposed algorithm performs competitive learning with code-vector rearrangement based on the online estimation of MC_i and MDC_i. We present the procedure of the proposed algorithm as follows.

Step 1: Initialize all the code vectors with randomly selected vectors from the set of input vectors, and set all the entries of the competition history table zero.
Step 2: At iteration T, choose an input vector $\boldsymbol{x}(T)$ and calculate distortions for each code vector. Let $d_i = d(\boldsymbol{x}(T), \boldsymbol{y}_i)$.
Step 3: Find $Q(\boldsymbol{x}(T))$ and $Q^*(\boldsymbol{x}(T))$.
Step 4: Delete the $(T-L+1)$-th line of the competition history table, and write $\arg Q(\boldsymbol{x}(T))$, $d(\boldsymbol{x}(T), Q(\boldsymbol{x}(T)))$ and $d(\boldsymbol{x}(T), Q^*(\boldsymbol{x}(T)))$ to the table.
Step 5: Incrementally calculate $MC_i(T)$ and $MDC_i(T)$ according to Eqs.(10) and (11), respectively.
Step 6: Find a code vector with the smallest $MC(T)$, $\boldsymbol{y}_{deleted}$.
Step 7: If $MDC(T)$ of $\boldsymbol{y}_{deleted}$ is less than a given error bound, and $d(\boldsymbol{x}(T), Q(\boldsymbol{x}(T)))$ exceeds the error bound, then jump $\boldsymbol{y}_{deleted}$ to the location of $\boldsymbol{x}(T)$. Moreover, replace entries concerning the $\boldsymbol{y}_{deleted}$ with null. Write an index of $\boldsymbol{y}_{deleted}$, 0, $d(\boldsymbol{x}(T), Q(\boldsymbol{x}(T)))$ to the current line of the competition history table.
If not, then update $Q(\boldsymbol{x}(T))$ based on the Kohonen competitive learning rule:

$$\boldsymbol{y}_{win} := \boldsymbol{y}_{win} - \alpha(T)(\boldsymbol{x}(T) - \boldsymbol{y}_{win}) \qquad (12)$$

where $\boldsymbol{y}_{win}$ denotes $Q(\boldsymbol{x}(T))$ and $\alpha(T)$ is a leaning rate at iteration T.
Step 8: If T is less than a predetermined value, $T = T+1$ and goto Step 2.

Fig. 2. Lena test image

Step 7 is a key of the proposed algorithm to perform rearrangement of a code vector.

IV. Performance evaluation

A. Experiments

We conducted experiments by applying the proposed algorithm to image compression based on VQ. For experiments, we used Lena test image shown in Fig. 2. The size of the image is 512×512 pixels, each of which has 8-bit depth. The image is divided into 4^2-pixel blocks to form 16384 16-dimensional input vectors. By subtracting pixel values with an average pixel value of each block, mean residual vectors were generated as input vectors. Image compression based on VQ of mean residual vectors reduces block noise and edge degradation compared with direct VQ.

The number of code vectors, N, was set 128 or 512. The initial code voctors were generated by randomly selecting from input vectors. All of the experiments were executed ten times and the results presented are the average value of

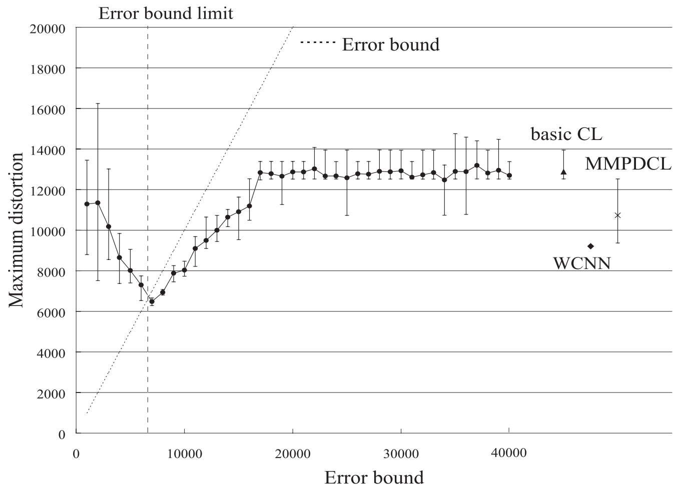


Fig. 3. Error bound and maximum distortion for $N = 128$

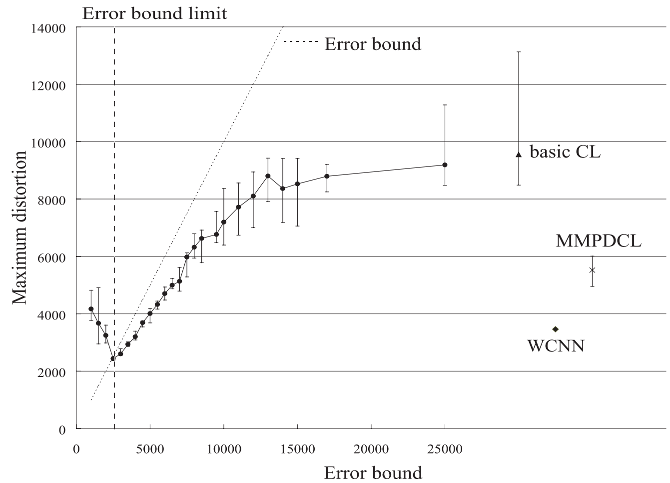


Fig. 4. Error bound and maximum distortion for $N = 512$

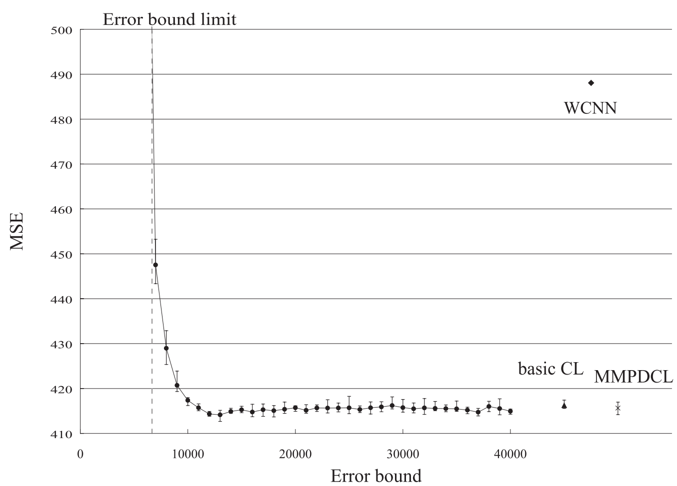


Fig. 5. Error bound and MSE for $N = 128$

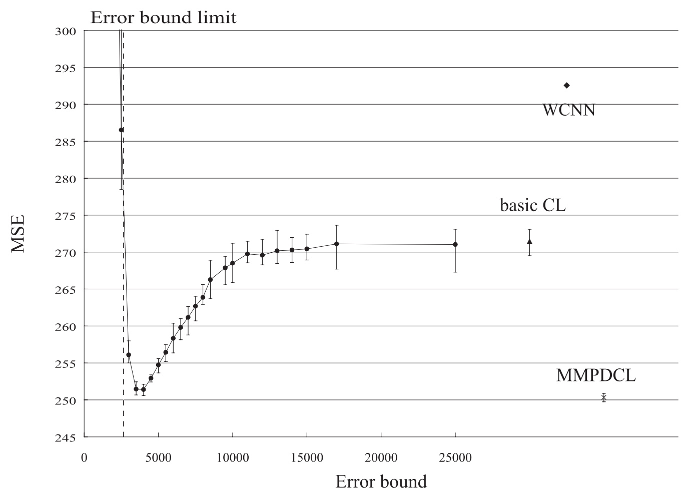


Fig. 6. Error bound and MSE for $N = 512$

the ten runs. In the following subsections, we will compare the proposed algorithm with Kohonen competitive learning algorithm (basic CL) and MMPDCL algorithm.

B. Results and discussion

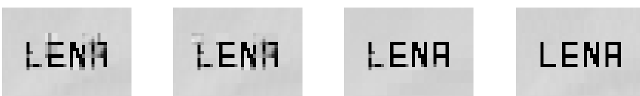
Figs. 3 and 4 show relationship between an error bound and a maximum distortion for $N = 128$ and 512, respectively. Each mark shows the average of ten experimental results, and a vertical line crossing the mark shows the maximum and minimum results. The proposed algorithm made the maximum distortion less than the given error bound when the error bound was not a too small value. The maximum distortion controlled by the proposed algorithm is always less than that by the Kohonen algorithm, and can be a smaller value than that by the MMPDCL algorithm.

For the larger number of code vectors ($N = 512$) allows a much smaller error bound to control the maximum distortion in comparison with the smaller number of code vectors ($N = 128$). Thus, the limitation of maximum-distortion control depends on the number of code vectors. We refer to the smallest error bound that can control the maximum distortion as *an error bound limit*. In addition, when a too small value is given to the error bound, the maximum distortion cannot be controlled and becomes a large value because there are not code vectors enough to reduce the maximum distortion.

Figs. 5 and 6 show relationship between an error bound and an MSE for $N = 128$ and 512, respectively. In these figures, an MSE has different trends. In Fig. 5, the MSEs obtained by the proposed algorithm are comparable to those by Kohonen algorithm and MMPDCL algorithm for error bounds larger than the error bound limit. This is because these algorithms do not suffer from local minima in minimizing an MSE for the small number of code vectors ($N = 128$). As an error bound approaches the error bound limit, an MSE slightly increases. When an error bound is less than the error bound limit, an MSE dramatically increases.

On the other hand, in Fig. 6, the MSE obtained by the proposed algorithm for large error bounds is similar to that by Kohonen algorithm, which is much larger than that by

(a) Lena test image with "LENA"



(b) Basic CL (c) MMPDCL (d) WCNN (e) Proposed

Fig. 7. Image compression results of Lena with characters.

MMPDCL algorithm. This is because Kohonen algorithm and the proposed algorithm were trapped by local minima due to the large number of code vectors ($N = 512$), while MMPDCL algorithm avoided the local minima and minimized an MSE. However, the MSE obtained by the propsed algorithm is reduced as an error bound decreases to the error bound limit. This is another effect of the rearrangement of code vectors. The rearrangement of code vectors gets rid of code vectors with low error contribution, which are under/never-utilized code vectors causing the local minima.

Through these experimental results shown in Figs. 3 to 6, WCNN algorithm achieved the maximum distortions that are comparable to those by the proposed algorithm. However, WCNN algorithm could not reduce MSEs in comparison with the proposed algorithm. This is because the inaccurate probability estimation and unnecessarily evened out distortions. Besides, we believe that the capability to control the maximum distortion is an outstanding advantage of our algorithm.

We also conducted experiments using the image with characters shown in Fig. 7(a). We inserted characters "LENA" in the Lena test image to generate blocks with large distortion. Each character consists of about 8×5 pixels. We used 128 code vectors and set an error bound $10,000$. Figs. 7(b) to (e) show the characters in the reconstructed images obtained by Kohonen algorithm, MMPDCL algorithm, WCNN algorithm and the proposed algorithm, respectively. While the former three algorithms resulted in the corrupted characters, the proposed algorithm could completely preserve them as they were in the original image.

V. Conclusions

In this paper, we have proposed a competitive learning algorithm with controlling maximum distortion. The proposed algorithm performs competitive learning with rearrangement of code vectors in order to control the maximum distortion. The rearrangement operation deletes a code vector with small MSE contribution, and assigns a new code vector to an input vector that has the maximum distortion larger than an error bound given by users. Experimental results have shown that the proposed algorithm can control the maximum distortion while minimizing an MSE.

Further performance evaluation should be conducted with various sets of input vectors for different configuration of parameters: the number of code vectors and dimension. In addition, we will introduce an additional mechanism to avoid local minima even though rearrangements do not occur so often.

Acknowledgments

This research was partially supported by Grant-in-Aid for Young Scientists(B) KAKENHI(#15700040) and Grant-in-Aid for Scientific Research(B) KAKENHI(#14380131).

References

[1] A.Gersho and R.M.Gray. *Vector Quantization and Signal Compression.* Kluwer Academic Publishers, 1992.

[2] N.Nasrabadi and Y.Feng. Vector quantization of images based upon the kohonen self-organizing feature maps. *Proceedings of the IEEE International Conference on Neural Networks*, 1:101–108, 1988.

[3] P.C.Cosman, K.L.Oehler, E.A.Riskin, and R.M.Gray. Using vector quantization for image processing. *Proceedings of the IEEE*, 81(9):1326–41, September 1993.

[4] O.T.C.Chen, B.J.Sheu, and W.C.Fang. Image compression using self-organizing networks. *IEEE Transactions on Circuits and Systems for Video Technology*, 4(5):480–489, October 1994.

[5] C.Amerijckx, M.Verleysen, P.Thissen, and J.D.Legat. Image compression by self-organized kohonen map. *IEEE Transactions on Neural Networks*, 9(3):503–507, May 1998.

[6] Paul Ning and Lambertus Hesselink. Vector quantization for volume rendering. *Proceedings of 1992 Workshop on Volume Visualization*, pages 69–74, October 1992.

[7] Boon-Lock Yeo and Bede Liu. Volume rendering of dct-based compressed 3d scalar data. *IEEE Transactions on Visualization and Computer Graphics*, 1(1):29–43, March 1995.

[8] B.Ramamurthi and A.Gersho. Classified vector quantization of images. *IEEE Transactions on Communications*, COM-34(11):1105–1115, November 1986.

[9] A.Gersho. Asymptotically optimal block quantization. *IEEE Transactions on Information Theory*, 25(4):373–380, 1980.

[10] D.-C. Park and Y.-J. Woo. Weighted centroid neural network for edge preserving image compression. *IEEE Transactions on Neural Networks*, 12(5):1134–1146, September 2001.

[11] T. Kohonen. *Self-Organization and Associative Memory.* Springer-Verlag, Berlin Heidelberg, 1989.

[12] Naonori Ueda and Ryohei Nakano. A new competitive learning approach based on an equidistortion principle for designing optimal vector quantizers. *Elsevier Journal on Neural Networks*, 7(8):1211–1227, 1994.

[13] Ce Zhu and Lai-Man Po. Minimax partial distortion competitive learning for optimal codebook design. *IEEE Transactions on Image Processing*, 7(10):1400–1409, 1998.