DISTORTION SENSITIVE COMPETITIVE LEARNING FOR VECTOR QUANTIZER DESIGN

Clifford Sze-Tsan CHOY *Wan-Chi SIU*

Department of Electronic Engineering, The Hong Kong Polytechnic University,Hong Kong

ABSTRACT

In this paper, we propose the Distortion Sensitive Competitive Learning (DSCL) algorithm for codebook design in image vector quantization. The algorithm is based on the equidistortion principle for asymptotically optimal vector quantizer after Gersho (1979) and recently from Ueda and Nakano (1994). The DSCL is simple and efficient in that a single weight vector update is performed per training vector, and the processing speed of the DSCL on sequential or multiprocessor environment can further be improved by applying a modified partial distance elimination (MPDE) method. Simulations indicate that the DSCL outperforms some recently proposed neural algorithms, including the "Neural-Gas" from Martinetz *et al.* (1993) and the DEFCL from Butler and Jiang (1996). In combining with the MPDE, the DSCL is faster than the "Neural-Gas" up to a factor of 45 times on a sequential machine, and yet arrives at better codebooks with the same number of iterations.

1. INTRODUCTION

Vector quantization (VQ) [1] has been applied successfully in image and speech compression for years. The signal to be quantized is divided into blocks, and each block is represented by a vector. A vector quantizer maps each of these vectors into a finite set of represetative vectors called the codevectors. These codevectors are collectively called the codebook, and according to a distortion measure $D\left(\vec{x}, \vec{c}\right)$ which measures the error introduced by replacing $\vec{x}$ with $\vec{c}$, they are chosen so as to minimize the average distortion introduced. Since the probability density function of the source is generally unknown, the codebook is usually designed with a training set sampled from the source, and the average distortion is approximated by the distortion introduced onto this training set.

Let us recall the well-known approach in codebook design, the generalized Lloyd algorithm (GLA) or the LBG, which was proposed by Linde *et al.* [2]. The GLA is an iterative descent algorithm which could be easily trapped at local optima corresponding to a large average distortion, especially when the initial codevectors are inappropriately assigned. Hence, a splitting method was proposed in the same paper to resolve this problem. Recently, neural networks have been applied to codebook design with success. These algorithms are characterized by being massively parallel in nature, adapting the codebook with each presentation of training vector instead of in batch (as in the GLA), and less sensitive to initialization than that of the GLA. Approaches like the soft competition scheme (SCS) [3] and the "Neural-Gas" [4] have also been proven theoretically for their global optimality. Generally speaking, the learning of each training vector in each of these algorithms consists of some form of competition for winner(s) which then follows by weight vector(s) update. In situations like massively parallel realizations on parallel machines or VLSI implementations, the competition process dominates the overall processing time while the weight(s) updating process is done in full parallelism. In more practical situations like sequential or multi-processing environments (where the number of processors is much less than the number of neurons), the competition process scales linearly (for the best case) with the number of neurons, while the weight(s) updating has to be done sequentially on each processor. Hence, the larger the number of weight vectors update, the longer will be the overall processing time. In fact, most of them require the updating of more than one weight vector per training vector, making their realizations in these environments less attractive than the GLA, especially when the number of neurons is large.

In this paper, we propose a new algorithm called the distortion sensitive competitive learning (DSCL) which is based on the equidistortion principle from Gersho [5] and recently from Ueda and Nakano [6]. The algorithm is characterized by single weight vector update per training vector, hence it is simple and efficient. To speed up the DSCL further, we suggest a modified partial distance elimination (MPDE) method which is an extension of the work from Bei and Gray [7]. Then we compare the performance of the DSCL with some existing neural algorithms, and finally conclude the paper.

2. THE EQUIDISTORTION PRINCIPLE AND THE DISTORTION SENSITIVE COMPETITIVE LEARNING

Let us recall the theoretical analysis given by Gersho [5] on asymtotically optimal vector quantizer. It is assumed that the probability density function of the source, $p\left(\vec{v}\right)$, is sufficiently smooth and the r-th power of Euclidean distance

$$D\left(\vec{x}, \vec{w}\right) = \left\| \vec{x} - \vec{w} \right\|^r \tag{1}$$

is used, where vectors are in K-dimensional Euclidean space. Let $\left\{\vec{w}_i\right\}$ be the codebook, then $\Re^K$ is being partitioned into $\{\Omega_i\}$ such that $\forall i \neq j,\ \Omega_i \bigcap \Omega_j = \emptyset,\ \bigcup_i \Omega_i = \Re^K$ and $\Omega_i = \left\{\vec{x} : D\left(\vec{x}, \vec{w}_i\right) < D\left(\vec{x}, \vec{w}_j\right),\ \forall i \neq j\right\}$. The subdistortion of the i-th codevector is given by

$$S_i = \int_{\Omega_i} D\left(\vec{x}, \vec{w}_i\right) p\left(\vec{x}\right) d\vec{x}, \tag{2}$$

and the overall expected distortion is given by $S = \sum_i S_i$. Then, according to ref. [5], in the limit of large N, a necessary condition for an optimal vector quantizer is that $\forall i \neq j,\ S_i = S_j$. This is usually referred to as the equidistortion principle in optimal vector quantizer design. In the work of Yamada *et al.* [8], they considered a more general difference distortion measure which has the general form

$$D\left(\vec{x}, \vec{y}\right) = L\left(\vec{x} - \vec{y}\right), \tag{3}$$

where $L(\cdot)$ is an arbitrary function satisfying $L\left(\vec{0}\right) = 0$ and $L\left(\vec{x}\right) \leq L\left(\vec{y}\right)$ iff $\left\|\vec{x}\right\| \leq \left\|\vec{y}\right\|$, where $\|\cdot\|$ is an arbitrary seminorm on $\Re^K$. They proved that the equidistortion principle is also applicable to this difference distortion measure. According to the work of Ueda and Nakano [6], the proof of Gersho [5] was extended so that the equidistortion prinicple holds even when $p\left(\vec{x}\right)$ composes of multiply disjoint distributions.

Recently, a number of approaches based on the equidistortion principle were proposed including the Competitive and Selective Learning (CSL) [6], the Partial Distortion Equivalent Competitive Learning (PDECL) [9], the Partial Distortion Weighted Fuzzy Competitive Learning (PDWFCL) [10] and the Distortion Equalized Fuzzy Competitive Learning (DEFCL) [11].

In this paper, we propose a new algorithm called the distortion sensitive competitive learning (DSCL) which is also based on the equidistortion principle. Let us denote N as the number of neurons, then the i-th neuron is associated with a weight vector $\vec{w}_i$ and a scalar b_i. With a training set available and the distortion measure defined according to (1), our DSCL is described as follows.

Algorithm 1: DSCL

1. Initialize randomly the weight vectors $\vec{w}_i(0)$. Set $b_i(0) \leftarrow 1\ \forall i$.
2. Do until some termination criteria
   (a) Randomly draw a training vector $\vec{v}(t)$ from the training set.
   (b) Find the winner, k, such that

$$k = \arg\min_i b_i(t) \cdot D\left(\vec{v}(t), \vec{w}_i(t)\right). \tag{4}$$

   (c) Update the weight vector by

$$\vec{w}_k(t+1) = \vec{w}_k(t) - \epsilon(t)\ \frac{\partial D\left(\vec{v}, \vec{w}\right)}{\partial \vec{w}}\bigg|_{\vec{w} = \vec{w}_k(t)}. \tag{5}$$

   (d) Update the value of b_k by

$$b_k(t+1) = b_k(t) + D\left(\vec{v}(t), \vec{w}_k(t)\right). \tag{6}$$

   (e) Set $t \leftarrow t+1$.

Note that Algorithm 1 is quite general in that by changing (6), we can arrive at some existing competitive learning algorithms. If b_k is not updated, we have the simple competitive learning (SCL). If (6) is replaced by $b_k(t+1) = b_k(t) + 1$, we have the Frequency Sensitive Competitive Learning (FSCL) [12]. In the work of Butler and Jiang [11], the Distortion Equalized Competitive Learning (DECL) was suggested as a non-fuzzy version of the DEFCL. The DECL is also bsed on the equidistortion principle, with with $b_k(t+1) = b_k(t) + \frac{b_k(t)}{\sum_i b_i(t)} D\left(\vec{v}(t), \vec{w}_k(t)\right)$. While both the DECL and our approach are based on the equidistortion principle, our approach has the advantage of being simpler, such that less computation is required to calculate the quantity $\triangle b_k(t) = b_k(t+1) - b_k(t)$. This simplicity also favours the hardware realization of the DSCL over the DECL, since in the DECL, additional circuitries have to be included to evaluate the global sum $\sum_j b_j(t)$ and the division operation. Another difference between the DSCL and the DECL is the magnitude of $\triangle b_k(t)$. For the DECL, this quantity is actually scaled by the factor $\frac{b_i(t)}{\sum_j b_j(t)}$ which can be very small when the number of neurons is large. Hence, the magnitude of $\triangle b_i(t)$ is generally larger for the DSCL. Owing to this fact, the performance of the DSCL is less affected by the values of $b_i(0)$ than that of the DECL, especially when the number iterations is small. We will present experimental results to demonstrate this point in Section 3..

In order to speed up the DSCL (and SCL, FSCL and DECL as well), we suggest a modified partial distance elimination (MPDE) method extended from the work of Bei and Gray [7]. With the distance measure as defined in (1) for the case with $r = 2$ and the vectors in K-dimensional space, our MPDE is described in Algorithm 2. Note that for the r-th power of Euclidean distance (for positive r), the MPDE is still applicable by using $b_i^{2/r}$ instead of b_i.

Algorithm 2: MPDE

1. Set $D_{min} \leftarrow b_1 \|\vec{w}_1 - \vec{v}\|^2$.
2. Set $Index \leftarrow 1$.
3. For i from 2 to N, do
   (a) Set $D \leftarrow 0$.
   (b) Set $Limit \leftarrow \frac{D_{min}}{b_i}$.
   (c) Let $\vec{w}_i$ be represented by $(y_1, \ldots, y_K)$, and $\vec{v}$ by $(v_1, \ldots, v_K)$
   For j from 1 to K, do

i. $D \leftarrow D + (y_j - v_j)^2$.

ii. If $Limit \leq D$,
then break this loop and continue with next value of i.

(d) Set $D_{min} \leftarrow D \cdot b_i$. Set $Index \leftarrow i$.

4. Terminate the algorithm and return $Index$ as the index of the best match codevector.

3. RESULTS

We demonstrate the performance of our Distortion Sensitive Competitive Learning in image vector quantization with $r = 2$ in (1), using a sub-block size of 4×4 pixels. Each image has 512×512 pixels, hence there are $L = 16384$ vectors in the training set of each image. The DSCL is compared with the FSCL [12], the "Neural-Gas" [4], the DEFCL and the DECL [11]. The quality of the codebook so generated is compared based on the signal-to-noise ratio (SNR),

$$\mathrm{SNR} = 10 \log \frac{\sum_{\vec{v} \in S} \left\| \vec{v} \right\|^2}{\sum_{\vec{v} \in S} \left\| \vec{v} - Q\left(\vec{v}\right) \right\|^2}, \qquad (7)$$

where $Q\left(\vec{v}\right)$ is the reconstructed vector of $\vec{v}$, and S is the training set.

The learning rate decreases linearly with number of learning steps t, i.e. $\epsilon(t) = 1 - \frac{t}{kL}$, where k is a constant specifying the learning time. The sizes of the codebook N in our experiments are $32, 64, 128, 256, 512, 1024$. For the "Neural-Gas", the neighbourhood function is chosen as $h_\lambda(k) = e^{-\frac{k}{\lambda}}$ where $\lambda(t) = \lambda_i (\frac{\lambda_f}{\lambda_i})^{\frac{t}{kL}}$, according to [4]. According to our analysis, we observe that the case with $\lambda_i = \frac{N}{8}$ and $\lambda_f = 0.01$ gives the best performance, hence these values are adopted. For the DEFCL, we choose the value of $m = 1.2$ since our analysis suggested that this gives the best codebook in all cases.

In our experiments, we chose $k = 10$, and the SNR of the resulting codebooks from each of the algorithm for each value of N with different images together with the corresponding processing times are tabulated in Table 1. It is obvious that for a small codebook size, the codebooks have similar quality, while for large ones, the proposed DSCL and the "Neural-Gas" outperform others. Note that in our experiments, it has been demonstrated that the DECL performs better than that of the DEFCL, although the contrary was concluded in ref. [11]. Although the DECL differs from the DSCL by a factor in the updating equation of b_i in (6), it is observed that this difference could lead to a degraded performance. In terms of processing speed, our DSCL is fastest while the DEFCL is the slowest. Note that in the implementations of the DSCL, the DECL and the FSCL, the MPDE is applied. On the other hand, the idea of the MPDE cannot be applied trivially in the implementation of the DEFCL and the "Neural-Gas". Experimental results indicate that the MPDE can improve the processing time by three times, while the DSCL is up to 45 times faster than that of the "Neural-Gas". Hence, even when the effect of the MPDE is removed, we still have a speed difference of 15 times.

Furthermore, we compare the adaptation speed of the three algorithms – the DSCL, the DECL and the "Neural-Gas". We consider an algorithm having higher adaptation speed than another one when it produces higher quality codebook given the same number of training steps. We compare the case when $N = 1024$, and for the two images "Lena" and "Tiffany". For each image, each of the algorithms was executed for the value of k ranging from 1 to 20, and the resulting plots are shown in Figure 1. Obviously, the DSCL has higher adaptation speed than the other two, except when k is small (≤ 2) and in this case the "Neural-Gas" has resulted in better codebooks. From the plots it is clear that in order to obtain a codebook with the same SNR, the DECL requires at least twice the number of steps than our proposed DSCL. For example, in the case of "Lena", the codebook quality for $k = 3$ from the DSCL is better than that from the DECL when $k < 10$. Hence, it is obvious that an extra factor of the updating equation of the DECL leads to a reduction of its adaptation speed.

4. CONCLUSIONS

In this paper, we propose the Distortion Sensitive Competitive Learning (DSCL) algorithm based on the equidistortion principle for asymptotically optimal vector quantizer. Comparing results with existing neural algorithms indicate that the DSCL is simple, fast and efficient in sequential environment. In combining with a modified partial distance elimination (MPDE) method for speeding up the competitive process, the DSCL is faster than the "Neural-Gas" algorithm by up to 45 times, and yet with better codebook quality.

Acknowledgments:This work was supported by the Research Grant Council of the University Grant Committee under the Grant HKP152/92E (PolyU340/939)

REFERENCES

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

[2] Yoseph Linde, Andrés Buzo, and Robert M. Gray. "An Algorithm for Vector Quantizer Design". *IEEE Transactions on Communications*, 28(1):84–95, January 1980.

[3] Eyal Yair, Kenneth Zeger, and Allen Gersho. "Competitive Learning and Soft Competition for Vector Quantizer Design". *IEEE Transactions on Signal Processing*, 40(2):294–309, February 1992.

[4] Thomas M. Martinetz, Stanislav G. Berkovich, and Klaus J. Schulten. "Neural-Gas Network for Vector Quantization and its Application to Time-Series Prediction". *IEEE Transactions on Neural Networks*, 4(4):558–569, July 1993.

[5] Allen Gersho. "Asymptotically Optimal Block Quantization". *IEEE Transactions on Information Theory*, 25(4):373–380, July 1979.

[6] Naonori Ueda and Ryohei Nakano. "A New Competitive Learning Approach Based on an Equidistortion Principle for Designing Optimal Vector Quantizers". *Neural Networks*, 7(8):1211–1227, 1994.

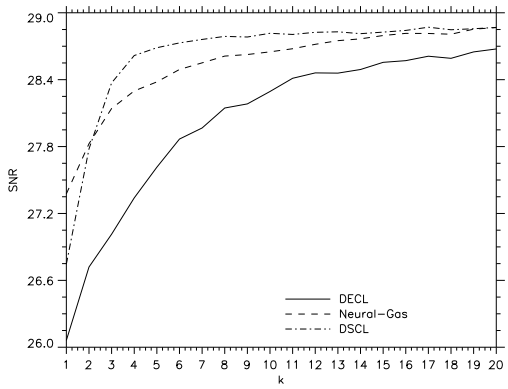

Figure 1-a. Plots of the SNR versus k for the three algorithms DSCL, DECL and "Neural-Gas" with the image "Lena" where k is related to the training steps

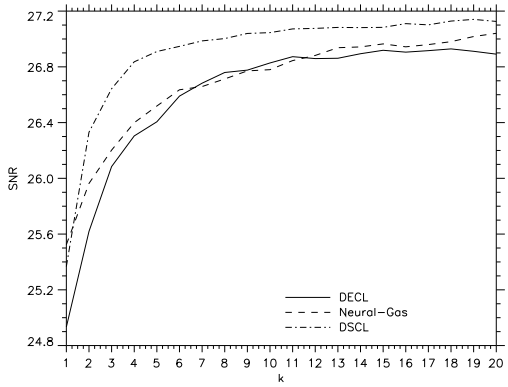

Figure 1-b. Plots of the SNR versus k for the three algorithms DSCL, DECL and "Neural-Gas" with the image "Tiffany"where k is related to the training steps

[7] Chang-Da Bei and Robert M. Gray. "An Improvement of the Minimum Distortion Encoding Algorithm for Vector Quantization". *IEEE Transactions on Communications*, 33(10):1132–1133, October 1985.

[8] Yoshio Yamada, Saburo Tazaki, and Robert M. Gray. "Asymptotic Performance of Block Quantizers with Difference Distortion Measures". *IEEE Transactions on Information Theory*, 26(1):6–14, January 1980.

[9] Ce Zhu, Lihua Li, Zhenyz He, and Jun Wang. "A New Competitive Learning Algorithm for Vector Quantization". *Proceedings of ICASSP'94*, pages 557–560, 1994.

[10] C. Zhu, L.H. Li, T.J. Wang, and Z.Y. He. "Partial-distortion-weighted fuzzy competitive learning algorithm for vector quantization". *Electronics Letters*, 30(6):505–506, March 1994.

[11] D. Butler and J. Jiang. "Distortion equalized fuzzy competitive learning for image data vector quantization". *Proceedings of ICASSP'96*, pages 3390–3393, May 1996.

[12] Stanley C. Ahalt, Ashok K. Krishnamurthy, Prakoon Chen, and Douglas E. Melton. "Competitive Learning Algorithms for Vector Quantization". *Neural Networks*, 3:277–290, 1990.

N	Approach	"Lena"	"Tiffany"	"Baboon"	"Peppers"
32	DSCL	22.85	21.90	16.74	22.32
	DECL	22.82	21.94	16.72	22.28
	FSCL	22.80	21.31	16.69	21.98
	DEFCL	22.89	21.84	16.66	22.31
	NGas	22.85	21.94	16.73	22.17
64	DSCL	23.94	22.76	17.36	23.50
	DECL	23.92	22.80	17.35	23.53
	FSCL	23.72	22.49	17.29	23.19
	DEFCL	23.96	22.80	17.25	23.53
	NGas	23.89	22.76	17.36	23.39
128	DSCL	25.06	23.71	17.95	24.59
	DECL	25.06	23.67	17.98	24.58
	FSCL	24.59	23.23	17.89	24.10
	DEFCL	25.05	23.61	17.85	24.60
	NGas	24.98	23.62	17.96	24.50
256	DSCL	26.14	24.60	18.61	25.67
	DECL	26.12	24.56	18.59	25.57
	FSCL	25.59	24.08	18.49	25.13
	DEFCL	26.12	24.51	18.44	25.56
	NGas	26.06	24.51	18.60	25.67
512	DSCL	27.30	25.67	19.32	26.83
	DECL	27.26	25.57	19.31	26.72
	FSCL	26.61	24.92	19.14	26.00
	DEFCL	26.93	25.32	19.09	26.39
	NGas	27.24	25.58	19.29	26.78
1024	DSCL	28.75	27.04	20.22	28.17
	DECL	28.29	26.83	20.18	27.73
	FSCL	27.65	26.03	19.96	27.14
	DEFCL	27.17	25.89	19.79	26.58
	NGas	28.65	26.78	20.18	28.08

Table 1-a. SNR of the codebooks (NGas = "Neural-Gas")

N	Approach	"Lena"	"Tiffany"	"Baboon"	"Peppers"
32	DSCL	20.67	23.92	27.82	20.72
	DECL	20.28	21.19	24.76	21.35
	FSCL	20.70	21.89	24.89	22.02
	DEFCL	421.93	422.16	452.83	421.99
	NGas	365.88	364.35	372.54	363.81
64	DSCL	31.75	30.92	40.80	30.18
	DECL	28.79	31.73	38.13	29.72
	FSCL	30.05	31.91	37.61	29.23
	DEFCL	831.87	830.26	839.99	840.39
	NGas	730.25	734.04	741.61	733.92
128	DSCL	46.69	51.30	66.71	46.13
	DECL	50.58	49.60	62.95	46.28
	FSCL	46.53	53.82	65.90	45.90
	DEFCL	1666.02	1668.64	1684.58	1665.81
	NGas	1428.58	1406.47	1445.70	1409.65
256	DSCL	78.57	85.88	111.23	78.57
	DECL	79.94	85.95	126.19	78.86
	FSCL	82.24	90.84	112.24	80.21
	DEFCL	3340.49	3347.92	3387.43	3356.42
	NGas	3035.65	2993.41	3042.21	2975.51
512	DSCL	144.51	151.63	202.19	139.53
	DECL	146.22	165.69	199.32	150.22
	FSCL	151.14	164.06	201.24	149.26
	DEFCL	6950.41	6655.87	6702.02	6703.80
	NGas	5817.67	5891.50	5882.94	5971.84
1024	DSCL	265.15	298.77	384.31	266.94
	DECL	283.48	290.50	376.66	293.07
	FSCL	294.80	324.80	379.60	301.28
	DEFCL	13432.70	13470.40	13657.40	13407.60
	NGas	12186.70	12962.90	12604.40	12220.40

Table 1-b. Processing Time in seconds (NGas = "Neural-Gas")