

An Improved Vector Quantizer Design Method: the Codebook Reorganization Algorithm

Ting-Wei Hou¹, Yuan-Tsung Chen^{1,2} and Houng-Kuo Ku^{1,3}

¹Department of Engineering Science, National Cheng Kung University, Tainan 701 Taiwan, R.O.C.

²South Branch, Bureau of National Healthcare Insurance, Tainan, Taiwan, R.O.C

³Department of Management Information System, Fortune Institute of Technology, Kaohsiung 831 Taiwan, R.O.C.

Abstract

Generalized Lloyd Algorithm (GLA) is important in vector quantizer design. It runs fast, but it is sensitive to initial conditions and it may find a local optimum. We propose an improved approach based on GLA, named vector quantized codebook reorganization algorithm (VQCRA). VQCRA finds better codebooks in less time, and is insensitive to initial conditions as compared with GLA. The experiments are performed on the 10 images extracted from USC-SIPI Image Database. The performance improvement ranges from 7.3% (the size of the codebook is 32) to 46.4% (the size of the codebook is 512).

Keywords: Vector Quantizer, Codebook design, Clustering algorithm, Data compression

1. Introduction

Among data clustering algorithms Generalized Lloyd Algorithm (GLA) [1] is one of the most famous algorithms. One issue of GLA algorithm is that it often converges toward one local optimum. A few researches attempted to overcome the issue by meta-heuristic methods of Tabu search [2], genetic algorithm [3], simulated annealing [4], stochastic relaxation [5], etc. to search toward one global optimum.

Data clustering is essentially to find each partition in the data set to have a minimum of SSE (sum of square error). The complexity of the algorithms for solving the data clustering problem has been proved to be NP-complete [6].

A typical vector quantization has three characteristics: a large amount of training vectors, high dimensional vector and a large number of partitions. We propose an improved algorithm for vector quantized codebook design, called vector-quantized codebook reorganization algorithm (VQCRA). It overcomes the local-optimal issue of GLA by codebook reorganization. It is not sensitive to the initial condition and the experiment results show that its performance is better than GLA.

The paper is organized as follows. Section II is background and design of VQCRA. Section III is the experimental results and discussions, and Section IV, the conclusion.

2. Vector Quantized Codebook Reorganization Algorithm

A worse initial codebook of GLA generates the result in Figure 1(a). where black data points are 2D fictitious data with 49 groups and bold black points are codewords found by GLA. Some codewords are not in appropriate positions enclosed by the dotted-line ellipses.

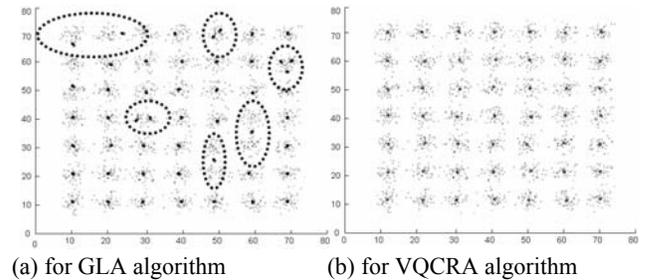


Fig. 1: Observation of codebook

Given the same initial codebook, by codebook reorganization, VQCRA generates Figure 1(b), and the codewords are totally on the rational positions.

```
variable: Ni, Nj // size of cluster i, j
rt // current reorganization times
rc // current reorganization coefficient
SSE_MERGE // array of the SSE increment value for cluster merging
removed_cluster_list // keep the cluster list which
// will be removed from codebook
splitted_cluster_list // keep the cluster list which will be splitted
cluster_size // array for keep the size of each cluster
cluster_se // array for keep the square error of each cluster
sse // sum of square error, that is sum of cluster_se
partition // array for keep the partition of each training vectors
old_codebook, new_codebook // set of the codewords
best_sse // keep the best sse found so far
best_codebook // keep the best codebook found so far
```

Fig. 2: Variables declaration in the VQCRA

We discovered that once the disposition of GLA codevectors was unsatisfactory, it leads to unbalanced SSE values in groups. Therefore, VQCRA is to balance the distribution of data set and average of SSE values in each group. The parameters of VQCRA, and VQCRA are described in Figure 2 and Figure 3.

```

Procedure VQCRA (Vector-Quantized Codebook Reorganization Algorithm)
Input: initial reorganization coefficient ( $R_{\text{init}}$ ), reorganization max times ( $L_r$ ),
training vectors (size N), codebook size (K), merge ratio ( $\tau$ )
Output:  $\text{best\_codebook}$ ,  $\text{best\_sse}$ 
Begin
1.  $rt = 0$ ; //variable initialization
2.  $rc = R_{\text{init}}$ ;
3.  $\text{best\_sse} = \infty$ ;
4.  $\text{best\_codebook} = \Phi$ ;
5. generate  $\text{old\_codebook}$  from training vectors by random;
6. while ( $rt < L_r$ )
do
7. ( $\text{new\_codebook}$ ,  $\text{partition\_cluster\_se}$ ,  $\text{sse}$ ,  $\text{cluster\_size}$ )
 $\leftarrow$  GLA( $\text{old\_codebook}$ ,  $c = 0.01$ );
8. update  $\text{best\_codebook}$  and  $\text{best\_sse}$ ;
9. for each codeword  $M_i$  in  $\text{new\_codebook}$  find its nearest codeword  $M_j$ 
and compute  $\text{SSE\_MERGE}(M_i) = (\tau * N_i * N_j) / (\tau * N_i + N_j) * (M_i - M_j)^2$ 
10.  $\text{removed\_cluster\_list} \leftarrow$  sort  $\text{SSE\_MERGE}$ , find lowest  $rc$  clusters
11.  $\text{splitted\_cluster\_list} \leftarrow$  sort  $\text{old\_cluster\_se}$ , find highest  $rc$  clusters
12.  $\text{old\_codebook} \leftarrow$  remove codewords from  $\text{new\_codebook}$  which in
 $\text{removed\_cluster\_list}$  and add new codewords into
 $\text{new\_codebook}$  based on  $\text{splitted\_cluster\_list}$ 
13.  $rc = R_{\text{init}}(1 - r) / L_r$ ;
14.  $rc = \lfloor rc \rfloor$ ;
15.  $rt \leftarrow rt + 1$ ;
end while;
16. output  $\text{best\_codebook}$  and  $\text{best\_sse}$ ;

```

Fig. 3: VQCRA algorithm

2.1. Groups Merging

Suppose two groups C_i and C_j , $C_i = (N_i, M_i, \text{SSE}_i)$, $C_j = (N_j, M_j, \text{SSE}_j)$. N_i and N_j are the numbers of members of C_i and C_j , respectively. M_i, M_j are the codevectors of C_i and C_j . $\text{SSE}_i, \text{SSE}_j$ is the summation of square error of C_i, C_j . Suppose that new group $C'_i = (N_i + N_j, M'_i, \text{SSE}'_i)$ is generated after the mergence of C_i and C_j satisfy the formula (1) [7].

$$\text{SSE}'_i = \text{SSE}_i + \text{SSE}_j + \frac{N_i * N_j}{N_i + N_j} (M_i - M_j)^2 \quad (1)$$

SSE'_i is the summation of square errors of C'_i . From the formula (1), the incremental value, $\Delta \text{SSE}_{\text{merge}}$ satisfies formula (2) after the mergence of C_i and C_j . Namely it expresses the variation of SSE values when two groups are completely merged.

Let the size of codebook be K ($\gg 2$). If a codevector M_i is to delete from a specific group C_i , the members of this group is to be absorbed by neighboring groups, which then triggers the merging of many groups at the same time. It is very difficult to estimate accurately for the incremental value, $\Delta \text{SSE}_{\text{merge}}$.

$$\begin{aligned} \Delta \text{SSE}_{\text{merge}} &= \text{SSE}'_i - \text{SSE}_i - \text{SSE}_j \\ &= \frac{N_i * N_j}{N_i + N_j} (M_i - M_j)^2 \end{aligned} \quad (2)$$

Therefore, we introduce the approximate formula (3):

$$\Delta \text{SSE}_{\text{merge}} = \frac{\tau * N_i * N_j}{\tau * N_i + N_j} (M_i - M_j)^2 \quad (3)$$

,where τ indicates the ratio of the members of C_i being absorbed by C_j after the mergence. It is obviously $0 < \tau \leq 1$.

2.2. Group Splitting

After carrying out GLA, group C_i may be chosen to split into two groups of $C_{i1} = (N_{i1}, M_{i1}, \text{SSE}_{i1})$ and $C_{i2} = (N_{i2}, M_{i2}, \text{SSE}_{i2})$, where $N_i = N_{i1} + N_{i2}$ and SSE values will be reduced at the same time. After splitting one group, the size of one codebook is increased by 1. Group splitting can be regarded as the reverse of the group merging. The variation of SSE values after splitting is shown in formula (4):

$$\begin{aligned} \Delta \text{SSE}_{\text{split}} &= \text{SSE}_{i1} + \text{SSE}_{i2} - \text{SSE}_i \\ &= -\frac{N_{i1} * N_{i2}}{N_{i1} + N_{i2}} (M_{i1} - M_{i2})^2 \end{aligned} \quad (4)$$

2.3. Codebook Reorganization

The most important idea of VQCRA is codebook reorganization. The operations of the codebook reorganization include: (1) (group merging) deleting R codevectors from the codebook and (2) (group splitting) inserting R codevectors into the codebook, while the codebook size remains unchanged and R is reorganization coefficient. The hinge of codebook reorganization is to determine the number of groups and which groups to merge and to split. The quality of codebook reorganization influences the efficiency of VQCRA.

VQCRA uses codebook reorganization to escape the local optimal solution and moves toward a global search. After escaping the local optimal, it uses GLA to perform a local search (a fine tune). The improvements on the codebook after the reorganization is performed until the maximum times (L_r) of reorganization is satisfied.

The disposition of codevectors is then achieved. It is in accordance with the distribution of the data set, and it also reduces the SSE values of the codebook. The physical meaning of codebook reorganization is summarized in the formula (5).

$$\Delta \text{SSE} = \Delta \text{SSE}_{\text{merge}} + \Delta \text{SSE}_{\text{split}} + \Delta \text{SSE}_{\text{GLA}} \quad (5)$$

In the steps 4-11 of Figure 3, ΔSSE is the variation of SSE value after VQCRA operation for one time. $\Delta \text{SSE}_{\text{merge}}$ is the variation of SSE value after the codebook has merged R groups, while $\Delta \text{SSE}_{\text{split}}$ is the variation of SSE value after R groups have been split. $\Delta \text{SSE}_{\text{GLA}}$ is the change of SSE values after fine adjustment of GLA. (GLA is used in the VQCRA to finely adjust codebook). It is obvious $\Delta \text{SSE} < 0$ and SSE value is decreased if $\Delta \text{SSE}_{\text{merge}} > 0$, $\Delta \text{SSE}_{\text{split}} < 0$, $\Delta \text{SSE}_{\text{GLA}} \leq 0$. and $|\Delta \text{SSE}_{\text{split}} + \Delta \text{SSE}_{\text{GLA}}| > \Delta \text{SSE}_{\text{merge}}$. This indicates a better solution is found. .

While performing codebook reorganization, VQCRA evaluates each codevector to determine which to delete and to make $\Delta \text{SSE}_{\text{merge}}$ as small as possible. Each codevector's SSE value in the

codebook is evaluated. And the codevectors are sorted in ascending order according to their Δ SSEs. The first R members of the sorted codevectors are deleted. Consequently, the SSE values in all groups are sorted by a descending order to generate new R codevectors. The first R members of the sorted list are split.

2.4. Reorganization Coefficient

Theoretically, the value of R should not exceed K/2. If R exceeds K/2, some groups are both merged and split, which is not desired.

R is suggested to decrease in the reorganization process to save computation time, because each time the codebook reorganization is performed, more codewords are approaching or at their final positions. Formula (6) is suggested to adjust R.

$$R = R_{ini} \left(1 - \frac{m}{L_c}\right) \quad (6)$$

In formula (6), R_{ini} is the initial reorganization coefficient, L_c is the maximum reorganization number, m is the number of reorganization performed at the time, $m \in [0, L_c - 1]$. As $m = L_c$, $R = 0$, codebook reorganization is completed and the algorithm terminates.

3. Experiments and Discussions

The algorithms was implemented in MATLAB 6.1 on a PC with Pentium 4 2.4GHz CPU and 512M memory. The ten data set of 256X256 images (couple, camera, moon, aerial, clock, tree, beans, girl, girl2, and girl3) was extracted from USC-SIPI Image Database [8].

The parameters of VQCRA were tuned to have better performance. VQCRA and GLA were fed the data set. The parameters of each algorithm were set up as follows:

- The size of codebook was 32, 64, 128, 256 and 512.
- The initial codebooks of each algorithm were all generated randomly.
- The final conditions of GLA were $\varepsilon = 0.01$, $\varepsilon = 0.001$ and $\varepsilon = 0$.

3.1. Performance for Initial Reorganization Coefficient

Several initial reorganization coefficients were tried to investigate their effects on the performance of VQCRA. They were K/2, K/4, K/8, K/16 and K/32, respectively. The decay of reorganization coefficient followed the formula (6). The relative percentage of distortion was based on VQCRA (K/4, 15, 0.5). The notation VQCRA (X, Y, τ), X represented the initial reorganization coefficient, Y represented the

maximum reorganization number and τ represented the ratio of group merging, respectively. The results are shown in Figures 4 and 5.

The results show if the initial reorganization coefficient is small, the speed of convergence is slow. Therefore, the optimal initial reorganization coefficient should be between K/4 and K/2. When the size of the codebook is 128 and 256, the optimal initial reorganization coefficient is K/2 as in Figures 4. If the size of the codebook is 512, the optimal initial reorganization coefficient is K/4 as in Figure 5.

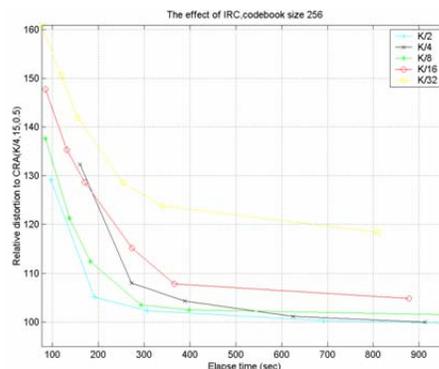


Fig. 4: Comparison of performance for initial reorganization coefficient in VQCRA (codebook size:256)

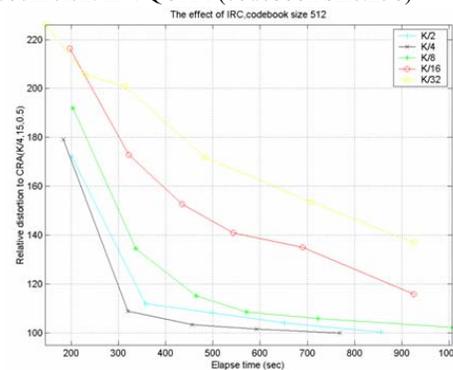


Fig. 5: Comparison of performance for initial reorganization coefficient in VQCRA(codebook size:512)

3.2. Performance

Figure 6 shows vector quantization of the ten test images and the percentage of global distortion of GLA and VQCRA. The percentage of global distortion is a simplified average value which is related to the relative distortion of ten images, and it serves as a metric of efficiency. The improvements of VQCRA over the GLA in Figure 6 are summarized in Table 1. The improved percentage of distortion of VQCRA over GLA (when reorganization number is 10) is increased while increasing the size of the codebook, from the value of 7.3% (the size of the codebook is 32) to 46.4% (the size of the codebook is 512). When the size of codebook is lower than 32, VQCRA's

improvement is not obvious. Therefore, VQCRA is especially suitable for the codebooks of larger sizes, such as 512.

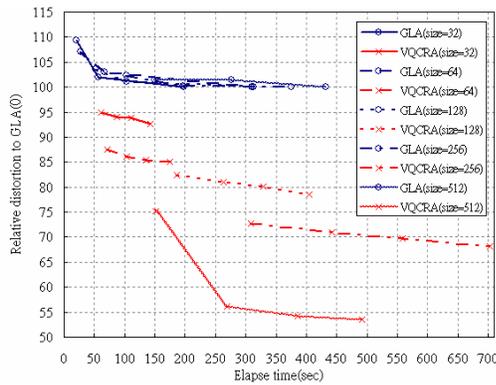


Fig. 6: Comparison of performance between GLA and VQCRA (codebook size:32, 64, 128, 256, and 512)

Table 1. The improved percentage of distortion of VQCRA related to GLA

| Size of Codebook | rt=3 | rt=5 | rt=7 | rt=10 |
|------------------|------|------|------|-------|
| 32 | 5 | 5.9 | 6.2 | 7.3 |
| 64 | 12.4 | 13.9 | 14.6 | 14.9 |
| 128 | 17.6 | 19 | 19.8 | 21.4 |
| 256 | 27.2 | 29 | 30.3 | 31.9 |
| 512 | 24.8 | 43.9 | 45.8 | 46.4 |

4. Conclusions

The proposed algorithm addresses the idea of codebook reorganization to overcome the shortcoming of Generalized Lloyd Algorithm. The experimental results show that the proposed algorithm generates a better vector quantizer design within a shorter time and, the improved percentage of distortion of the proposed algorithm to that of the Generalized Lloyd algorithm is gradually increasing from 7.3% (with codebook size 32) the highest value of 46.4% (with codebook size 512). The improved percentage of distortion of the proposed algorithm is gradually increased along with the increase of the size of the codebook. The proposed algorithm is especially suitable for the codebooks of more than 32 codevectors.

5. References

[1] Y. Linde, A. Buzo and R.M. Gray, “ An algorithm for vector quantizer design ”, IEEE Trans. Comm. , pp.84-95, vol.28, no.1,1980.
 [2] P. Franti, J. Kivijarvi, and O. Nevalainen, “ Tabu search algorithm for codebook generation in vector quantization ”, Pattern Recognition, pp.1139-1148, vol.31,no.8, 1998.

[3] J. Holland, “ Adaption in natural and artificial systems: an introductory analysis with applications to biology, control and artificial intelligence”, University of Michigan Press,1975.
 [4] Ngoc-Ai Lu and Darryl R. Morrell, “ VQ codebook design using improved simulated annealing algorithms ”, IEEE Trans. on Acoustics, Speech, and Signal Processing, pp.673-676,vol.1, Apr. 1991.
 [5] K. Zeger and A. Gersho, “ Stochastic relaxation algorithm for improved vector quantization design ”, Electron Lett., pp.896-898, vol.25, no.14, Jul 1989.
 [6] P. Brucker, “ On the complexity of clustering problems ”, Lectures Notes in Economics and Mathematical Systems , pp.45-54, 1978.
 [7] W.H. Equitz, “ A new vector quantization clustering algorithm ”, IEEE Trans. on acoustics, speech ,and signal processing. vol.37,no.10,Oct. 1989.
 [8] The USC-SIPI Image Database, <http://sipi.usc.edu/services/database/Database.html>