4 PEREZ, F., and KOCH, C.: 'Toward color image segmentation in analog VLSI: Algorithm and hardware', *Int. J. Comput. Vis.*, 1994, **12**, (1), pp. 17–42

5 BALLARD, D.H. and BROWN, C.M.: 'Computer vision' (Prentice-Hall Inc., 1982), pp. 334–340

Reallocation of GLA codevectors for evading local minima

T. Kaukoranta, P. Fränti and O. Nevalainen

Indexing terms: Data compression, Vector quantisation, Image processing

The performance of the generalised Lloyd algorithm (GLA) is improved by reallocating the codevectors every time the GLA reaches a local optimum. This is achieved by splitting the largest partition and by merging two small neighbouring partitions, thereby preserving the size of the codebook. The whole procedure is repeated until no improvement is achieved.

Introduction: Vector quantisation is a technique that has been researched and applied widely in the compression of images and speech data. By using this method we can code the original signal by the codewords of a fixed codebook. The main problem is the construction of a good codebook on the basis of a set T of known training vectors which is supposed to form a representative sample of the signal to be processed. We suppose that the training vectors are K-dimensional and denote the size of T by N. Vector quantisation partitions the input space into M non-overlapping regions, called partitions or clusters, so that the input space is completely covered. A representative (codevector) is then assigned to each partition. The aim of the codebook construction is to minimise the average distance (distortion) between the training vectors and their representatives. The question of what is a proper choice for T is not an issue here. The motivation is merely to select the best possible codebook for the given training set despite how it has been chosen. The distance between two vectors X and Y is measured by the square error

$$d(X,Y) = \sum_{k=1}^{K}(X[k] - Y[k])^2 \tag{1}$$

The total distortion caused by a codebook is calculated by

$$D = \sum_{i=1}^{N} d(X_i, f(X_i)) \tag{2}$$

where $f(X)$ is the mapping function from T to the codevectors in the codebook. Thus, the mapping function $f(X)$ defines the partitioning of the training set.

GLA: There are several known methods for generating a codebook [1]. The most cited and widely used is the generalised Lloyd algorithm (GLA) [2]. It tries to improve an existing initial codebook by iterating the following two steps. In the partition step the training set is partitioned according to the existing codebook. The optimal partitioning is obtained by mapping each training vector to the nearest codevector as defined by eqn. 1. In the codebook step a new codebook is constructed by calculating the centroids of the partitions defined in the partition step. These two steps guarantee that the next codebook is always equal to or better than the previous one.

GLA makes local changes in the codebook and finally ends to a local optimum. This raises a question as to whether the local optimum could be evaded and thus a better codebook obtained. Simulated annealing [3, 4] tries to do this by adding random noise to the codevectors. A better codebook can be reached but the changes are still local. Here we propose an alternative approach in which we make non-local changes in the codebook. The GLA algorithm is augmented by a codevector reallocation step that makes a single change to the codebook without decreasing its size or quality. The main motivation of the reallocation is to create a new starting point to the GLA that would be relatively far from the previous solution. If the codebook can be improved by the new step, it can possibly be improved further by the GLA.

Reallocation of codevectors: Consider the distortion of a codebook from the point of view of a single codevector. Denote by C_j the set of training vectors that are mapped to the codevector Y_j:

$$C_j = \{X | X \in T, f(X) = Y_j\} \tag{3}$$

The total distortion caused by the codevector Y_j is

$$D_j = \sum_{X_i \in C_j} d(X_i, Y_j) \tag{4}$$

and the total distortion caused by the codebook for the training set

$$D = \sum_{j=1}^{M} D_j \tag{5}$$

where M is the size of the codebook. The motivation of the new algorithm is based on the following hypothesis: in an efficient codebook, each codevector contributes to the total distortion D by an equal amount:

$$D_i = D_j \quad \forall\, i, j \tag{6}$$

In this case the codevectors are used with equal effectiveness and they are spread out into the vector space evenly, with respect to the distribution of the training vectors. Consider the distortion caused by the individual codevectors in a codebook generated by the GLA. This is illustrated in Fig. 1 where the frequencies of the D_j values are shown. The distortion is not distributed equally, so the previous hypothesis is not fulfilled in codebooks generated by the GLA.

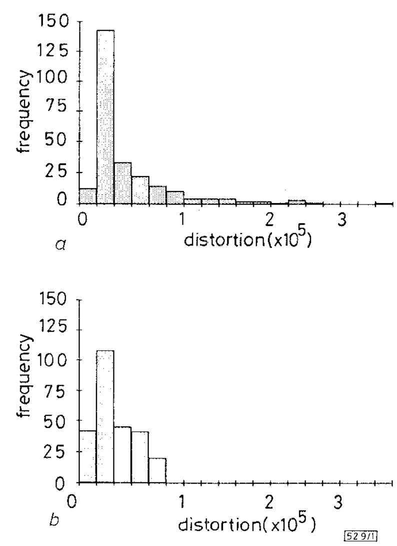

Fig. 1 *Histogram of distortion values of individual codevectors (for training set Camera)*

a GLA; standard deviation = 51574
b GLA with reallocation; standard deviation 21870

The aim of the reallocation phase is to decrease the variance of the D_j-values. Thus, we consider the codevectors that generate either very high or very low distortion values. Large distortion originates typically from two kinds of clusters. If the number of training vectors in a cluster is large their contribution to the total distortion may be significant even if individual distances are moderate. In the second type, the cluster is heterogeneous in the sense that vector-to-centroid distances are long. The cluster itself may contain only a few vectors.

Our reallocation strategy is based on the combination of merge and split operations. Two clusters with small distortion values are selected. The clusters are merged and the centroid of the new cluster is calculated. This operation releases one codevector that can be allocated elsewhere. A third cluster, that with the highest distortion value, is then selected. It is split so that two codevectors are used to code the training vectors belonging to the selected cluster.

The two operations have an opposite effect; the merge operation increases the distortion and the split operation decreases the distortion. The main (and only) benefit of these operations, however, is not the overall decrease in the distortion of the codebook. It is more important that the new codebook can be input to the GLA as a different starting point, which is far away from the current local minimum.

New algorithm: Generate an initial codebook by choosing M random vectors from the training set. Iterate the codebook by the GLA (until no improvement achieved).

Repeat
  Select clusters C_{m1} and C_{m2} to be merged
  Select cluster C_s to be split
  Merge C_{m1} and C_{m2}
  Split C_s
  Iterate the codebook by GLA.
Until Total distortion D of the codebook does not decrease.

Merge phase: The clusters C_{m1} and C_{m2} can be selected by considering the distance of the centroids, or the increase of the distortion caused by the merge operation [5]. Because the goal is to decrease the variance of the distortion values D_j, it is natural to choose clusters that produce the smallest distortion when merged.

Split phase: There are several different ways to select the cluster for splitting [6]. Here we use a greedy heuristic by selecting the cluster with the highest distortion value. The actual split is performed by assigning two codevectors to the cluster, and then letting the GLA iterations take care of the partitioning. The chosen codevectors are the original one (the centroid of the cluster), and the training vector that has the longest distance from the centroid. This choice guarantees that the distortion is decreased at least by the amount of the distance between the furthest vector and the centroid. It may not produce an optimal splitting but the clusters will be modified by the GLA anyway. Therefore the assignment of the new codevector is not crucial for the performance of the algorithm overall.

Details: Despite the fact that the merge and split operations modify three clusters only, the effect of these operations will be more global. As noted before, merging increases the distortion because only one codevector is used to represent a large number of training vectors. However, it is likely that some of the training vectors of the merged cluster become nearer to the codevectors of the adjacent clusters than the codevector of the merged cluster.

Also the splitting has an influence on the adjacent clusters. In particular, the selection of the new codevector causes some of the training vectors from the adjacent clusters to be mapped to this new one. Thus, the merge and split operations make rough changes to the codebook, whereas the GLA performs the fine tuning.

Note that the proposed reallocation phase does not guarantee an improved codebook. One reason is that the clusters with small distortion value may be located wide apart so that their merging will increase the distortion more than the saving due to the splitting phase. It is therefore difficult to predict the overall effect of the reallocation phase. We just have to observe if the distortion of the new codebook is decreased. If we succeed, the overall iteration step is then repeated. On the other hand, if the codebook is not improved (or even made worse), the algorithm will terminate and the final codebook will be the previous one.

Table 1: Average MSE-values of 100 codebooks with random initial codebook

Training set	Initial (Random)	GLA	GLA with reallocation
Bridge	261.39	179.87	178.63
Camera	236.55	123.17	90.86
Couple	70.00	40.20	30.98
Lena	103.58	59.56	58.78

Performance evaluation: Four test images (Bridge, Couple, Camera, Lena) were divided into 4×4 blocks giving 16-dimensional vectors. From these vectors, four training sets were constructed. The first three have 4096 training vectors, and the last one (Lena) 16384. From all training sets 100 random initial codebooks of 256 codevectors were generated. The GLA and the method of this Letter were applied to improve all the initial codebooks.

The proposed method was able to decrease the variance of the distortion values D_j for all tested training sets. This is illustrated in Fig. 1 where the frequencies of the distortion values D_j are shown for training set Camera. The distortion values of the codebooks are then summarised in Table 1. The proposed method was able to improve the distortion by 23–26% for training sets Camera and Couple. However, the improvement for Bridge and Lena was only ~1%.

In cases where the proposed method improved the codebook considerably, we observed an increase in the number of codevectors causing sero distortion. This is because the split clusters are often located in sparse areas in the vector space. When we split a cluster of this type the new codevector is located so far away from the other vectors that no other training vector will be mapped to it despite the GLA iterations. Thus, the modification did not have the desired cumulative effect on the other neighbouring clusters in these cases.

Conclusion: The performance of the GLA was improved by reallocating the codevectors using merge and split operations. The new method was able to decrease the distortion of the codebook in all cases. The improvement was 23–26% for Camera and Couple, but remained only marginal (~1%) for Bridge and Camera. A more sophisticated technique for the selection of the merged clusters might improve the results of the two latter cases.

© IEE 1996
3 June 1996
Electronics Letters Online No: 19961075s

T. Kaukoranta and O. Nevalainen (*Department of Computer Science, University of Turku, Lemminkäisenkatu 14 A, FIN-20520 Turku, Finland*)

P. Fränti (*Department of Computer Science, University of Joensuu, PB 111, FIN-80101 Joensuu, Finland*)

References

1 AKROUT, N., PROST, R., and GOUTTE, R.: 'Image compression by vector quantization: a review focused on codebook generation', *Image and Vision Computing*, 1994, **12**, (10), pp. 627–637

2 LINDE, Y., BUSO, A., and GRAY, R.M.: 'An algorithm for vector quantizer design', *IEEE Trans.*, 1980, **COM-28**, (1), pp. 84–95

3 VAISEY, J., and GERSHO, A.: 'Simulated annealing and codebook design'. Proc. ICASSP, April 1988, **2**, pp. 1176–1179

4 ZEGER, K., and GERSHO, A.: 'Stochastic relaxation algorithm for improved vector quantiser design', *Electron. Lett.*, 1989, **25**, (14), pp. 896–898

5 EQUITZ, W.H.: 'A new vector quantization clustering algorithm', *IEEE Trans.*, 1989, **ASSP-37**, (10), pp. 1568–1575

6 GERSHO, A., and GRAY, R.M.: 'Vector quantization and signal compression' (Kluwer Academic Publishers, 1992)

Recursive splitting of active contours in multiple clump segmentation

H.H.S. Ip and R.P.K. Yu

Indexing terms: Image processing, Segmentation

A new technique is presented for clump decomposition based on the recursive splitting of active contours. The approach does not require prior knowledge of the number of objects and the sizes of the objects to be segmented.

Introduction: Techniques exist to perform clump decomposition but can only be used for silhouette images [5 – 7]. Hand active contours [3], or snakes, have been applied to segment objects