Novel K-means Algorithm for Compressing Images

K. Somasundaram
Professor
Department of Computer Science & Applications
Gandhigram Rural Institute – Deemed University, Tamil Nadu, India

M. Mary Shanthi Rani
Assistant Professor
Department of Computer Science & Applications
Gandhigram Rural Institute – Deemed University, Tamil Nadu, India

ABSTRACT
Our proposed method is a two phase scheme that enhances the performance of K-means vector quantization algorithm for compressing images**.** In the proposed method, we have explored the possibility of application of statistical parameters for choosing the initial seeds for K-means algorithm. The selection of initial seeds depends on the statistical features of input data set. The novelty in our approach is the judicious selection of initial seeds based on variance, mean, median and mode parameters. Considering mode value of each dimension of the data adds uniqueness to our method. Our approach shows better performance yielding good PSNR and variable bit rate at a very low time complexity. This method is best suited for online web applications that involve massive and rapid image and video transmission.

Keywords
Vector Quantization, K-means, variance, mode, Break Even Point, Rate-distortion

1. INTRODUCTION
K-means [1] is a widely used Vector Quantization technique known for its efficiency and speed. It has great number of applications in the fields of image and video compression, watermarking, speech and face recognition.

The k-means algorithm is an algorithm to cluster objects based on attributes into k partitions. The objective is to minimize total intra-cluster variance, or the squared error function

$$V = \sum_{i=1}^{k} \sum_{x_j \in S_i} |x_j - \mu_i|^2 \quad (1)$$

where there are k clusters S_i, i = 1, 2,..., k and μ_i is the centroid or mean point of all the points in cluster S_i.

The algorithm starts by partitioning the input points into k initial sets, either at random or using some heuristic data. It then calculates the centroid of each set and constructs a new partition by associating each point with the closest centroid. Then the centroids of each set are recalculated and the algorithm is repeated by alternate application of these two steps until convergence, which is obtained when the points no longer switch clusters (or alternatively centroids are no longer changed).

Despite its efficiency, it has drawbacks like apriori fixation of number of clusters and random selection of initial seeds. Inappropriate choice of initial seeds may yield poor results and leads to increase in computation time for convergence.

A pioneering work on seed initialization was proposed by Ball and Hall (BH) [2]. A similar approach, named as Simple Cluster Seeking (SCS), was proposed by Tou and Gonzales [13]. The SCS method chooses the first input vector as the first seed and the rest of the seeds are selected if they are'd' distance apart from all selected seeds.

The SCS and BH methods are sensitive to the parameter d and the order of the inputs. Astrahan [3] suggested a method using two distance parameters, d1 and d2 which first computes the density of each point in the dataset. The highest density point is chosen as the first seed and subsequent seed points are chosen in the order of decreasing density subject to the condition that each new seed point be at least at a distance of 'd2' from all other previously chosen seed points. The drawback in this approach is that it is very sensitive to the values d1 and d2 and requires hierarchical clustering.

Kaufman and Rousseau [4] introduced a method that estimates the density through pair wise distance comparison and initializes the seed clusters using the input samples from the areas with high local density. A significant drawback of this method is its computational complexity which may sometimes be higher than k-means when 'n' is large.

Katsavounidis et al (KKZ) [6] suggested a parameter less approach, which chooses the first seed near the "edge" of the data, by choosing the vector with the highest norm. Then, it chooses the next seed to be the point that is farthest from the nearest seed in the set chosen so far. This method is very inexpensive (O (kn)) and is easy to implement but is sensitive to outliers.

Bradley and Fayyad (BF) [7] proposed an initialization method that is suitable for large datasets. The main idea of their algorithm is to select 'm' subsamples from the data set, apply the k-means on each subsample independently, keep the final k centers from each subsample and produce a set that contains mk points. They apply the k-means on this set m times: for the first time, the first k points as initial centers and for the second time, the second k points and so on. And the algorithm returns the best k centers from this set. They use 10 subsamples from the data set, each of size 1% of the full dataset size.

Finally, a last round of k-means is performed on this dataset and the cluster centers of this round are returned as the initial seeds for the entire dataset. This method generally performs better than k-means and converges to

the local optimum faster. However, it still depends on the random choice of the subsamples and hence, can lead to poor clustering in an unlucky session.

Fahim *et al*. [8] proposed a method to minimize the number of distance calculations required for convergence. Arthur and Vassilvitski [10] proposed the k-means++ approach, which is similar to the KKZ method. However, while choosing the seeds, they choose a point with a probability proportional to its distance from already chosen seeds. In k-means++, the next point chosen will be the one with the probability proportional to the minimum distance of this point from already chosen seeds. Due to the random selection of first seed and probabilistic selection of remaining seeds, different runs have to be performed to obtain a good clustering.

Single Pass Seed Selection (SPSS) [11] algorithm initializes first seed and the minimum distance that separates the centroids based on the point which is close to more number of other points in the data set.

An efficient fast algorithm to generate VQ codebook was proposed by Kekre et al [12] which use sorting method to generate codebook and the code vectors are obtained using median approach.

In this paper we propose a method Novel K-means which is a variant of Bradley and Fayyad's method [7]. Our method achieves two goals. First, by adopting appropriate seed selection method for edge blocks (high variant blocks) and non-edge blocks (low variant blocks), our method preserves image quality. Second, by using divide and conquer strategy significant reduction in computation time is achieved.

The rest of the paper is organized as follows: Section 2 briefly describes the method, Section 3 presents the performance analysis of the proposed method and Section 4 concludes our work.

2. NOVEL K-MEANS

The proposed method is an innovative approach to enhance the performance of K-means algorithm. It adopts divide and conquer strategy by dividing the input data space into classes and apply K-means clustering. Novel K-means segments input image into *K* clusters and works in two phases as follows. In the first phase, the input vectors are divided into *N* classes based on the mean value of each vector. As the range of the value of a pixel *R* in image data is 0-255, the range of the domain of each class is set as the ratio of R to the number of classes (*N*). A training vector is assigned to a class C_I, if its mean value is less than or equal to the domain of C_I **.** This mean based grouping of input data speeds up convergence resulting in reduction of computation time.

K-Means algorithm is performed on each class of vectors with initial seeds. The number of initial seeds *M* in each class is proportional to its size. The initial seeds are selected with the right blend of statistical features (variance, mean, median and mode) of the class population.

If high variant vectors outnumber low variant vectors in the class, 50% of the initial seeds are chosen from the vectors having high variance .The remaining seeds are the vectors whose values are mode values of each dimension of the class vectors, in descending order. On the other hand, if low variant vectors dominate the class population (high correlation), the ratio of high variant vectors and high mode vectors would be 1:3. The correlation among data determines the ratio of contribution of high variance and high mode vectors. If the class population is equal to 2, the mean of the two vectors is taken as the representative seed of the class.

The code vectors generated from each class are subjected to pruning based on a minimum (Euclidean) distance threshold "D" with the code vectors of the partially generated code book. A new code vector V_{1i} of a class $C_{i,}$ will be added to the codebook only if its Euclidean distance with all of the code vectors of classes $C_1..C_{i-1}$ is greater than a predefined threshold *D*. Thus, pruning of code vectors are done at the construction phase which significantly reduces the code book size and computation time as well.

The second phase runs a final round of K-means with the codebook constructed in the first phase as the initial seeds.

2.1 Algorithm

The input image data is divided into 4x4 vectors and the number of classes *N* should be set before the seed selection and pruning phases. The choice of *N* plays a crucial role in the performance of Novel K-means.

2.1.1 Seed selection phase

1. Divide the 512x512 input image data into 4x4 image blocks and convert the blocks into training vectors (16-vecctor) of dimension 16.

2. Calculate the range of the domain of each class which is equal to 256 / *N* * *I*, where *I* is the Class_index between 1 and N.

3. Calculate the mean value of each training vector

4. Assign each training vector to one of the *N* classes say C_{Ii}, if its mean value is less than or equal to the domain of the class C_I

5. Calculate *T* which is equal to the ratio of total number of training vectors and number of initial seeds *K*.

6. Determine the number of initial seeds *M[I]* for each class C_I as follows :

$$M[I] = Class\text{-}size[I] / T$$

so that $\sum M[I] = K$, $I = 1 \ldots N$.

7. Calculate the variance of each vector and find the number of vectors with variance higher than the mean variance of the class population.

8. Set *m1[I]* and *m2[I]* to be the number of high variant vectors and mode vectors respectively that form the initial seeds for class C_I. If we let *h_no* to be the number of high variant vectors and *Class_size[I]* to be the total number of vectors in the class C_I, then

$$if\ h\_no \geq Class\_size[I]/2$$

$$m1[i] = M[I]/2$$

$$else \quad m1[I] = M[I]/4$$

$$m2[I] = M[I] - m1[I]$$

9. Construct the remaining 'm2' seeds based on the statistical parameters mean, median and mode as follows.

    - Take the mean vector of the class as the first seed.
    - Form the median vector with the median value of each column in the class which is taken as the second seed.
    - Construct mode vectors whose values are the mode value (value with highest frequency of occurrence) of its respective dimension in descending order until we get desired number of seeds (m2-2). If mode_vector[i,j] is the i^{th} mode vector, then

    mode_vector[i,j] = pixel value that occupies i^{th} position in descending order of frequency of occurrence(mode) in j^{th} dimension of all the vectors of a class.

10. Perform K-means algorithm on each class with *M* initial seeds.

2.1.2 Pruning and Construction Phase

11. Calculate the Euclidean distance between each of the newly generated code vectors of the current class C_I and the code vectors in the partial codebook constructed from classes $C_1 \ldots C_{I-1}$ so far.
12. Add only those code vectors whose Euclidean distance with all of the partial codebook vectors is greater than a predefined threshold *D* .to the codebook.
13. Repeat this process for all the classes C_I, I=1.. N
14. The final code book resulting after step 13 is the initial codebook for the entire data set.
15. Perform final run of K-means on the the entire data set with this final codebook as initial seeds.

3. PERFORMANCE ANALYSIS

We evaluate the performance of the proposed method with several test images of size 512 x 512 on a machine with core2 duo processor at 2 GHZ using MATLAB.

An objective measure of reconstructed picture quality is the Peak Signal to Noise Ratio (PSNR) defined by

$$PSNR = 20 \log_{10}\left(\frac{255}{\sqrt{MSE}}\right)$$

where *MSE* is the mean-square error measuring the deviation of the reconstructed image from the original image. Bit rate is measured in bits per pixel (BPP).

The results obtained for Lena image with D=4 and K=256 are given in Table 1. From Table 1, we observe that there is significant drop in computation time with increase in the number of classes without loss in reconstructed image quality and Bit rate. Novel K-means on Lena image achieves 70% drop in computation time than K-means with random initialization of seeds. It is observed that it accomplishes abrupt drop in computation time for a particular value of 'N' which can be considered as the Break-Even Point (BEP). Table 2 shows comparative performance of K-means and Novel K-means with different test images . Experimental results demonstrate that Novel K-means shows optimal performance in terms of Computation time, Bit rate and PSNR at this BEP and it has also been observed that there is only a small decrease in computation time with increase in value of *N* above BEP. We note that BEP is image-specific and is equal to $c(\log_2 K)$ for some constant c which lies in the range between 0.5 and 1 where K is the number of initial seeds. Several runs of Novel k-means on different test images show that value of c depends on the number of frequently occurring pixel values (hit-no). The value of c is equal to 0.5 for images whose hit-no is less than 200 and 1 for images with hit_no >200.

Besides, as the pruning phase prunes redundant code vectors, the size of the final codebook is found to be less than K thereby enhancing compression rate. The threshold D used for the pruning process should be chosen so as to achieve better rate-distortion performance. We learn from our experiments that this value of *D* is expected to be in the range 4-6 to get good rate-distortion performance.

Regarding the memory requirements, Novel K-means requires storage for the codebook and code indices as well. If *S* is the number of code vectors (16-vector) in the final code book, the size *P* of the codebook in bits , is given by

$$P = S \times 16 \times 8 \text{ bits}$$

Each training vector requires an 8 bit index to find its matching code vector in the decoding process.

Table 1. Performance analysis of Lena Image for D=4 and K=256

N (No. of classes)	Computation time in sec	BPP	PSNR
(K-means)	1500	0.125	33.03
2	1200	0.125	32.9
4 (BEP)	286	0.122	32.966
8	200	0.121	32.85
16	134	0.120	32.842
32	124	0.120	32.811

Figure 1. Original Lena image

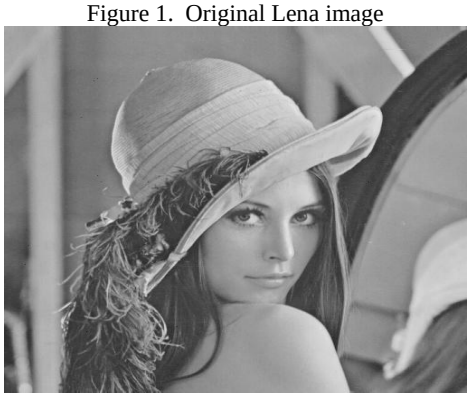

Figure 2. Novel K-means Lena

Table 2 . Comparative Performance of K-means and Novel K-means

Image	Method	BEP (N)	Computation time	Bit rate	PSNR
Barbara	Kmeans		2075	0.125	28.2
	Novel K-means	4	322	0.123	28.18
Boat	K-means	-	362	0.125	31.4
	Novel K-means	8	150	0.121	31.3
Peppers	K-means	-	3160	0.125	31.4
	Novel K-means	8	762.59	0.119	31.322
Couple	K-means	-	1196	0.125	28.898
	Novel K-means	8	699	0.124	28.908

Also Novel K-means yields a unique solution for any number of runs whereas K-means with random initialization does not give consistent results. The divide and conquer strategy reduces the computation complexity and eventually time complexity to a great extent without loss of quality and bit rate.

4. CONCLUSION

In this paper, we have proposed a two phase method in which the use of mode parameter for initial seed selection has been investigated. Novel K-means outperforms traditional K-means with random choice of initial seeds significantly in terms of computation time with comparable PSNR and bit rate. This would be an ideal choice for time-bound applications of image compression like video conferencing, online search engines of image and multimedia databases.

5. REFERENCES

[1] Lloyd, S.P., 1982. Least square quantization in PCM, IEEE Trans. Inform. Theory, vol 28: pp 129-136

[2] Ball, G.H. and D.J. Hall, 1967. PROMENADE-an online pattern recognition system, Stanford Research Inst. Memo, Stanford University

[3] Astrahan, M.M., 1970. Speech Analysis by Clustering, or the Hyperphoneme Method, Stanford A. I. Project Memo, Stanford University .

[4] Kaufman, L. and Rousseeuw, 1990. Finding Groups in Data: An Introduction to Cluster Analysis. Wiley, New York, ISBN: 0471878766, pp: 342.

[5] Gersho, A. and R.M. Gray, 1992. Vector Quantization and Signal Compression, Kluwer Academic, Boston, ISBN: 0792391810, pp: 761.

[6] Katsavounidis, I., C.C.J. Kuo and Z. Zhen, 1994. A new initialization technique for generalized Lloyd iteration. IEEE. Sig. Process. Letters, pp 144-146

[7] Fayyad, U.M., G. Piatetsky-Shapiro, P. Smyth and R. Uthurusamy, 1996. Advances in Knowledge Discovery and Data Mining. AAAI/MIT Press, ISBN: 0262560976, pp: 611.

[8] Bradley, P.S. and U.M. Fayyad, 1998. Refining initial points for K-means clustering. Proceeding of the 15th International Conference on Machine Learning (ICML'98), July 24-27, ACM Press, Morgan Kaufmann, San Francisco, pp: 91-99.

[9] Fahim, A.M., A.M. Salem, F.A. Torkey and M. Ramadan, 2006. An efficient enhanced k-means clustering algorithm. J. Zhejiang Univ. Sci. A., 7: pp:1626-1633.

[10] Deelers, S. and S. Auwatanamongkol, 2007. Enhancing K-means algorithm with initial cluster centers derived from data partitioning along the data axis with the highest variance. Proc. World Acad. Sci. Eng. Technol., 26: pp: 323-328.

[11] Arthur, D. and S. Vassilvitskii, 2007. k-means++: The advantages of careful seeding. Proceeding of the 18th Annual ACM-SIAM Symposium of Discrete Analysis, Jan. 7-9, ACM Press, New Orleans, Louisiana, pp: 1027-1035.

[12] K. Karteeka Pavan, Allam Appa Rao, A.V. Dattatreya Rao and G.R. Sridhar, Single Pass Seed Selection Algorithm for k-Means**,** Journal of Computer Science 6 (1): pp: 60-66.

[13] Kekre, H.B. Sarode, T.K. Thadomal Shahani , An Efficient Fast Algorithm to Generate Codebook for Vector Quantization , proceedings of International Conference on Emerging Trends in Engineering and Technology, pp.62-67.

[14] Tou, J. and R. Gonzales, 1977. Pattern Recognition Principles. Addision-Wesley, Reading, MA., ISBN: 0201075873, pp: 377

[15] Feng Wu, Xiaoyan Sun, Image Compression by Visual Pattern Vector Quantization (VPVQ) , Data Compression conference, 1068-0314/08 © 2008 .