Journal of Information Hiding and Multimedia Signal Processing ©2017 ISSN 2073-4212
Ubiquitous International Volume **8**, Number **3**, May **2017**

Edge and Contrast Classified K-means Algorithm for Image Vector Quantizer Design

Xiao-Dan Jiang

College of Electrical and Information Engineering, Quzhou University
QuZhou, 324000, P. R.China
16282409@qq.com

Zhe-Ming Lu*

School of Aeronautics and Astronautics, Zhejiang University
Hangzhou, 310027, P. R.China
*Author Corresponding: zheminglu@zju.edu.cn

Hui Li

College of Information Science and Technology, Hainan University
Haikou 570228, P. R.China
lihui@hainu.edu.cn

Received September, 2016; revised December, 2016

Abstract. *Vector Quantization (VQ) is one of effective image compression techniques. High compression quality depends on designing a good VQ codebook. Based on two optimality conditions, the conventional K-means algorithm is a widely used scheme to generate a codebook. However, the initial codebook greatly affects both the convergence speed and the quality of the generated codebook. Therefore, many researchers have presented all kinds of methods to obtain a good initial codebook, including splitting-based, pruning-based, pairwise nearest neighbor-based, random initialization-based and maximum distance initialization-based schemes. However, all these techniques do not take into account the features of the training vectors and some of them need high extra computational load. This paper presents a simple classification based technique to obtain a better initial codebook. During the initial codebook generation process, we classify the training vectors according to their edge orientations (eight templates) and their contrast information (binary decision), and thus we can obtain sixteen subsets. Then we just randomly select several codewords from each subset proportionally to the number of training vectors in the subset. Experimental results demonstrate that, compared with the conventional and modified K-means algorithms under the random initialization strategy, our scheme converges to a better codebook with fewer iterations.*
Keywords: Vector quantization, Image compression, Codebook design, K-means algorithm, Initial codebook generation.

1. **Introduction.** Vector quantization (VQ) [1] has been been widely applied in data compression [2] and data clustering [3]. A vector quantizer Q of level K and dimension n can be viewed as a mapping from the n-dimensional Euclidean space R^n into a finite set C, i.e.,

$$Q : R^n \to C = \{\boldsymbol{y}_1, \boldsymbol{y}_2, \ldots, \boldsymbol{y}_K\} \tag{1}$$

where the set C is called the codebook, K is the codebook size, and $\boldsymbol{y}_i = (y_{i1}, y_{i2}, \cdots, y_{in})^{\mathrm{T}}$, $1 \leq i \leq K$, are called the codewords. After quantization, any n-dimensional input vector $\boldsymbol{v} = (v_1, v_2, \ldots, v_n)^T$ can be quantized into a codeword in C, that is,

$$Q(\boldsymbol{v}) \in \{\boldsymbol{y}_1, \boldsymbol{y}_2, \ldots, \boldsymbol{y}_K\} \tag{2}$$

A suitable distortion metric $d(\cdot,\cdot)$ is often used in vector quantization, and therefore the quantization result $Q(\boldsymbol{v})$ of $\boldsymbol{v}$ should be the codeword $\boldsymbol{y}_j$ with the smallest distortion to $\boldsymbol{v}$. The squared error is widely used in VQ, assume that Q is designed based on the training set $X = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_M\}$, where $\boldsymbol{x}_i = (x_{i1}, x_{i2}, \ldots, x_{in})^{\mathrm{T}}$, $1 \leq i \leq M$, then we can evaluate the quality of Q by the following average distortion

$$D = E[d(\cdot, Q(\cdot))] = \frac{1}{M}\sum_{i=1}^{M} \|\boldsymbol{x}_i - Q(\boldsymbol{x}_i)\|^2 \tag{3}$$

VQ has simple decoding structure, and it can achieve high compression ratio while maintaining acceptable reconstruction quality, and therefore becomes a popular data compression technique. In general, the compression performance of VQ is dependent on the quality of the codebook C. Thus, how to design an optimal codebook is very important for a VQ-based compression system. Based on a training set X, the aim of optimal vector quantizer design is to seek the codebook C of size K that minimizes the average distortion over all possible codebooksof size K. As we know, there are two famous conditions that an optimal vector quantizer should satisfy. One is called nearest neighbor condition which assigns each training vector to the codeword that is closest to it. The other is called the best codebook condition which requires that each codeword should be the centroid of the training vectors that are mapped to it. The above two conditions can be easily used to develop an efficient scheme named K-means algorithm for locally optimal codebook design iteratively. This algorithm is also called the generalized Lloyd algorithm (GLA) or the Linde-Buzo-Gray (LBG) algorithm [4].

The main disadvantages of GLA lie in two aspects, one is that it can only converge to a locally optimal codebook, the other is that both the convergence speed and the quality of the obtained codebook depend on the initial codebook. To overcome these problems, many researchers [5] have been devoted to enhancing the performance of GLA in terms of quality or speed. In order to improve the codebook quality, most scholars have tried to use global optimal schemes to find a better solution for the codebook generation problem [6, 7, 8]. However, the cost of these kinds of population-based schemes is too high to be utilized in online codebook generation. With the increase of the image size and the demand for online processing, the speed has become a severe problem in codebook design. To improve the speed, many scholars have paid particular attention to speeding up the codebook generation. Some approaches use more efficient codebook structures to reduce the time required to assign training vectors to codewords such as tree-structured vector quantization (TSVQ) [9]. Some methods reduce the number of comparisons required to assign training vectors to codewords to which they belong [10, 11]. Some schemes use new codeword updating steps rather than the conventional centroid-based updating step[12, 13].

As we have known, not only the convergence speed but also the quality of the converged codebook depend on the initial codebook, this paper focuses on generating a better initial codebook. In the past several decades, many algorithms have been put forward to generate a good initial codebook, including splitting-based, pruning-based, pairwise nearest neighbor design (PNN)-based, random initialization based and maximum distance initialization based schemes[14]. However, all these techniques do not take into account the features of each training vector. Therefore, in this paper, we propose a simple and efficient initialization technique for the K-means algorithm used in image vector quantization by classifying the input vectors into sixteen subsets by a simple edge classifier together with a contrast classifier, then we randomly select several initial codewords from each subset with the number of codewords being proportional to the number of training vectors in the subset. Experimental results demonstrate that, compared with the conventional and modified K-means algorithms with random selection initialization, the new initialization technique converges to a better codebook with a relatively faster convergence speed.

The rest of the paper is organized as follows. Section 2 introduces conventional and modified K-means algorithms in detail. The proposed edge and contrast classified K-means scheme is presented in Section 3. The experimental results are given in Section 4. Finally, we draw a conclusion in Section 5.

2. **Related Works.**

Before describing our scheme, we introduce the conventional K-means scheme and two modified K-means schemes that we will compare in the experiments.

2.1. **Original K-Means Scheme.**

The original K-means algorithm divides the set of training vectors $X = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_M\}$ into K clusters V_i in such a way that the two necessary conditions for optimality are satisfied. In the algorithm description, m is the iteration index and $V_i^{(m)}$ is the i-th cluster at iteration m, with $\boldsymbol{y}_i^{(m)}$ its centroid. The algorithm can be described as follows.

Step 1. Initialization Step: Load the training set $X = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_M\}$ into memory. Set $m = 0$. Choose a set of initial codewords $\boldsymbol{y}_i^{(0)}, 1 \leq i \leq K$.

Step 2. Partitioning Step: Partition the set of training vectors $X = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_M\}$ into the clusters $V_i, 1 \leq i \leq K$ based on the nearest neighbor condition as follows

$$V_i^{(m)} = \{\boldsymbol{v} \in X | Q^{(m)}(\boldsymbol{v}) = \boldsymbol{y}_i^{(m)}\} = \{\boldsymbol{v} \in X | d(\boldsymbol{v}, \boldsymbol{y}_i^{(m)}) \leq d(\boldsymbol{v}, \boldsymbol{y}_j^{(m)}), \texttt{all} j \neq i, 1 \leq j \leq K\} \tag{4}$$

Step 3. Codeword Updating Step: Update the codeword of each cluster by computing the centroid of the corresponding training vectors in each cluster as follows

$$\boldsymbol{y}_i^{(m+1)} = \frac{1}{|V_i^{(m)}|} \sum_{\boldsymbol{v} \in V_i^{(m)}} \boldsymbol{v} \tag{5}$$

Step 4. Termination Test Step: If the decrease in the overall distortion d_{m+1} at iteration $m+1$ relative to d_m is below the threshold e, stop the algorithm, i.e., stop if $|d_{m+1} - d_m|/d_{m+1} \leq e$, where d_{m+1} is the average distortion after $m+1$ iterations defined as

$$d_{m+1} = \frac{1}{M} \sum_{i=1}^{M} \|\boldsymbol{x}_i - Q^{(m+1)}(\boldsymbol{x}_i)\|^2 \tag{6}$$

Otherwise, replace m by $m+1$ and go to Step 2.

2.2. **Modified K-Means Schemes.**

Lee et al. [12] proposed a modified K-means algorithm, which results in a better locally optimal codebook than K-means algorithm with the same initial codebook. The modified K-means algorithm is almost the same as the conventional K-means algorithm except for an improvement at the codebook updating step. They update the current codeword $\boldsymbol{y}_i^{(m)}$ at iteration m to the new codeword $\boldsymbol{y}_i^{(m+1)}$ at iteration $m+1$ as

$$\boldsymbol{y}_i^{(m+1)} = \boldsymbol{y}_i^{(m)} + s \times [\frac{1}{|V_i^{(m)}|} \sum_{\boldsymbol{v} \in V_i^{(m)}} \boldsymbol{v} - \boldsymbol{y}_i^{(m)}] \tag{7}$$

where $s > 0$ is a scale factor. Based on the squared-error distance measure, it has been shown experimentally that the modified K-means algorithm converges slower in comparison to the conventional K-means algorithm when $s < 1$. When $1 < s < 2$, it converges faster and results in better performance. When $s > 2$, the algorithm either does not converge, or converges very slowly with poor performance. When $s = 1$, the modified K-means algorithm is just the same as the conventional K-means algorithm. Based on experiments, the best results lie in $s = 1.8$.

The use of a "fixed" scaling for the entire range of iterations results in the use of step sizes larger than the corresponding centroid-update at iterations closer to convergence and causes undesirably high perturbations of the codewords which are otherwise converging to some optimal configuration. This in turn has the effect of increasing the number of iterations required to converge as well as perturbing the codebook convergence to a poorer local optimum. Thus, Paliwal and Ramasubramanian [13] proposed the use of a variable scale factor s in the codeword updating step which varies as a function of the iteration m and is inversely proportional to m as follows:

$$s = 1 + \frac{x}{x+m} \tag{8}$$

where $x > 0$. In this equation, $s = 2$ when $m = 0$, and $s = 1$ when $m = \infty$. thus, it satisfies the aforementioned conditions. To see the effect of variable x used in the scale factor equation, Paliwal and Ramasubramanian have studied the algorithm with various values of x. According to their results, they finally adopt $x = 9$.

3. **Proposed Edge and Contrast Classified K-Means Algorithm.** In image vector quantizer design, the feature distribution of the training set is very important for the initial codebook generation. In this paper, we consider two types of features, i.e., edge and contrast. During the initialization step, our algorithm classify the training vectors into sixteen classes based on their edge orientations and contrast information. Assume that the training set is composed of M image blocks of size 4×4, i.e., $X = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_M\}$, the proposed scheme can be illustrated in Fig.1. The initial codebook generation process can be described as follows:

Step 1. Each training vector $\boldsymbol{x}_i (i = 1, 2, \ldots, M)$ is input into the edge classifier and contrast classifier, and an overall index $t_i \in \{1, 2, \ldots, 16\}$ is output to denote which class the training vector belongs to.

Step 2. We collect all the training vectors belonging to the same class to generate a subset, and thus we have 16 subsets P_j of sizes $s_j (j = 1, 2, \ldots, 16)$ respectively, where $s_1 + s_2 + \ldots + s_{16} = M$.

Step 3. We initialize $K \times s_j / M$ initial codewords from each subset P_j based on random selection, thus we can in total obtain K initial codewords.

Step 4. The modified K-means algorithm in [13] is used to generate the final codebook. During the iteration steps, if an empty cell occurs, we just judge the classes that all current codewords belong to and find the class with the least number of codewords, and randomly select a training vector from this class as a new centroid.

Now, we turn to describing how to classify the training vectors based on our edge classifier and contrast classifier. With regards to edge classification, inspired by the Structured Local Binary Kirsch Pattern (SLBKP) in [15] that adopts eight 3×3 Kirsch templates to denote eight edge directions, we propose eight 4×4 templates for edge classification as shown in Fig.2. Assume the input image X is segmented into non-overlapping 4×4 blocks, the edge classification can be described as follows: First, we perform eight 4×4 edge orientation templates on each 4×4 block $x(p, q)$, $1 \le p \le 4$, $1 \le q \le 4$, obtaining

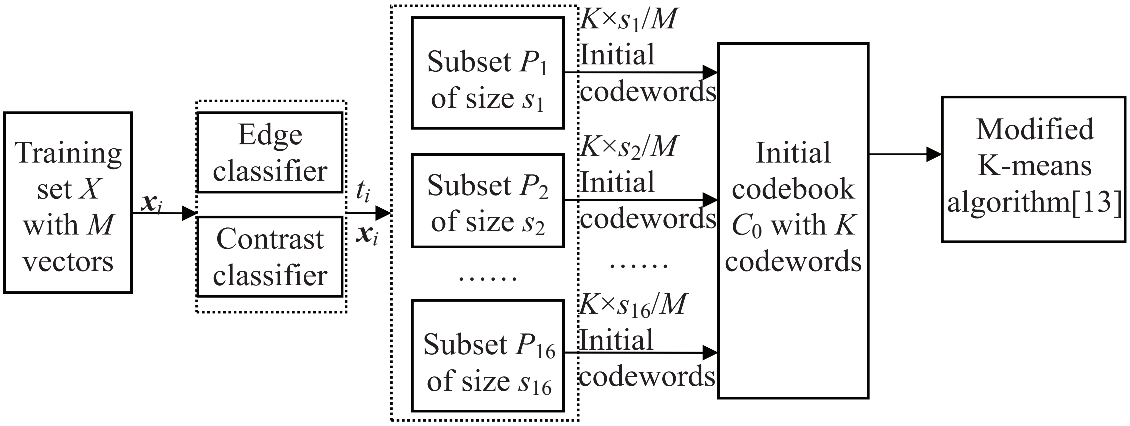

FIGURE 1. The block diagram of the proposed edge classified K-means algorithm.

$$\mathbf{E}_1=\begin{bmatrix}-4&2&2&2\\-4&0&0&2\\-4&0&0&2\\-4&2&2&2\end{bmatrix},\mathbf{E}_2=\begin{bmatrix}2&2&2&2\\0&0&2&2\\-4&-4&0&2\\-4&-4&0&2\end{bmatrix},\mathbf{E}_3=\begin{bmatrix}2&2&2&2\\2&0&0&2\\2&0&0&2\\-4&-4&-4&-4\end{bmatrix}$$

$$\mathbf{E}_4=\begin{bmatrix}2&2&2&2\\2&2&0&0\\2&0&-4&-4\\2&0&-4&-4\end{bmatrix},\mathbf{E}_5=\begin{bmatrix}2&2&2&-4\\2&0&0&-4\\2&0&0&-4\\2&2&2&-4\end{bmatrix},\mathbf{E}_6=\begin{bmatrix}2&0&-4&-4\\2&0&-4&-4\\2&2&0&0\\2&2&2&2\end{bmatrix}$$

$$\mathbf{E}_7=\begin{bmatrix}-4&-4&-4&-4\\2&0&0&2\\2&0&0&2\\2&2&2&2\end{bmatrix},\mathbf{E}_8=\begin{bmatrix}-4&-4&0&2\\-4&-4&0&2\\0&0&2&2\\2&2&2&2\end{bmatrix}$$

FIGURE 2. The eight 4×4 edge orientation templates.

an edge orientation vector $\boldsymbol{v} = (v_1, v_2, \ldots, v_8)$ with its components $v_i (1 \leq i \leq 8)$ being calculated as follows:

$$v_i = \left|\sum_{p=1}^{4}\sum_{q=1}^{4}[x(p,q)\cdot e_i(p,q)]\right| \tag{9}$$

Where $e_i(p,q)$ denotes the element at the position (p,q) of $\mathtt{E}_i$. Thus, an input block $x(p,q)$ is classified into the j-th category if

$$j = \arg\max_{1\leq i\leq 8} v_i \tag{10}$$

Thus, we can classify each training vector into one of eight categories according to its edge orientation. For the contrast classifier, we first calculate the average value μ of all components in each training vector as follows

$$\mu = \frac{1}{16}\sum_{p=1}^{4}\sum_{q=1}^{4} x(p,q) \tag{11}$$

And then we calculate the contrast σ by the following formula

$$\sigma = \frac{1}{16} \sum_{p=1}^{4} \sum_{q=1}^{4} [x(p,q) - \mu] \tag{12}$$

And then we classify the input vector into the high-contrast or smooth class based on the threshold 3.0. Thus, in total, based on above two classifiers, we can classify the training vectors into $8 \times 2 = 16$ categories.

4. **Experiments and Analysis.** To demonstrate the performance of the proposed edge classified K-means algorithm, we compared our scheme with the traditional K-means algorithm (KMeans), the modified K-means algorithm with the fixed scale value $s = 1.8$[12] (`MKM_F`) and the modified K-means algorithm with a variable scale value and $x = 9$ [13] (`MKM_V`). In our experiments, we used two 512×512 monochrome images with 256 gray levels, Lena and Peppers. We segmented each image into 16384 blocks, and each block is of size 4×4. We tested the performance for different codebook sizes of 256, 512, and 1024. The quality of the compressed images is evaluated by PSNR. Because the random selection is adopted for each algorithm, the performance is averaged over ten runs.

All the algorithms are terminated when the ratio of the mean squared error difference between two iterations to the mean squared error of the current iteration is within 0.0001 or 0.01%. In Table 1, the PSNR values and the numbers of iterations for the Lena image with different codebook sizes are shown, where 'Best' and 'Ave' denote the best and the average results over ten runs respectively. In Table 2, the PSNR values and the numbers of iterations for the Peppers image with different codebook sizes are shown. From Table 1 and Table 2, we can see that, if the random selection technique is used, our scheme requires the least average number of iterations than other algorithms and can also get better codebooks than other algorithms on average.

Table 1. Performance comparison for Lena image with random selection initialization (CBsize: codebook Size; itr: number of iterations)

CBSize	256		512		1024	
Performance	PSNR(dB)	itr	PSNR(dB)	itr	PSNR(dB)	itr
KMeans: Best	30.447	25	31.293	29	32.138	27
KMeans: Ave	30.379	36	31.237	38	32.083	34
MKM_F: Best	30.505	22	31.453	23	32.439	20
MKM_F: Ave	30.465	26	31.420	27	32.387	26
MKM_V: Best	30.470	16	31.420	14	32.452	17
MKM_V: Ave	30.436	23	31.393	24	32.383	22
Our: Best	30.514	15	31.466	17	32.455	15
Our: Ave	**30.472**	**22**	**31.434**	**23**	**32.401**	**19**

5. **Conclusions.** This paper presents an improved K-means algorithm for image vector quantization. The main idea is to classify the training vectors into 8 categories based on the edge orientation of each vector, and then randomly select initial codewords from each category with the number of codewords proportional to the number of vectors in each category. The experimental results based on three test images show that our algorithm can converge to a better locally optimal codebook with a faster convergence speed.

Table 2. Performance comparison for Peppers image with random selection initialization (CBsize: codebook Size; itr: number of iterations)

CBSize	256		512		1024	
Performance	PSNR(dB)	itr	PSNR(dB)	itr	PSNR(dB)	itr
KMeans: Best	29.863	32	30.591	24	31.368	19
KMeans: Ave	29.799	43	30.540	37	31.313	29
MKM_F: Best	29.956	25	30.789	25	31.712	18
MKM_F: Ave	29.917	37	30.714	33	31.625	24
MKM_V: Best	29.912	20	30.758	19	31.732	16
MKM_V: Ave	29.861	30	30.710	24	31.593	20
Our: Best	29.968	16	30.796	17	31.725	15
Our: Ave	**29.923**	**24**	**30.720**	**23**	**31.636**	**19**

Acknowledgment. This work was partly supported by the Zhejiang Provincial Natural Science Foundation of China (Grant No. LY15F010003 and No. LY14F020024), the Public Good Research Project of Science and Technology Program of Zhejiang Province, China. (NO.2016C31097), the Science and Technology Project of QuZhou, Zhejiang, China.(NO.2015Y005), the Project for Visiting Scholars' Professional Development in Universities of Zhejiang Province, China (No.FX2014102),the Project for Young Academic Leaders' Academic Promotion in Colleges and Universities of Zhejiang Province, China(No. pd2013450), and the Natural Science Foundation of Zhejiang Province, China (No. LY15F020041).

REFERENCES

[1] A. Gersho, R.M. Gray, Vector quantization and signal compression, Springer Science and Business Media, 2012.

[2] F. D. V. R. Oliveira, H. L. Haas, J. G. R. C. Gomes, A. Petraglia, CMOS imager with focal-plane analog image compression combining DPCM and VQ, *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 60, no.5, pp. 1331–1344, 2013.

[3] F. X. Yu, H. Luo, Z. M. Lu, Colour image retrieval using pattern co-occurrence matrices based on BTC and VQ, *Electronics Letters*, vol. 47, no.2, pp. 100–101, 2011.

[4] Y. Linde, A. Buzo, R. M. Gray, An algorithm for vector quantizer design, *IEEE Transactions on Communications*, vol. 28, no.1, pp. 84–95, 1980.

[5] A. Vasuki, P. Vanathi, A review of vector quantization technique, *IEEE Potentials*, vol. 25, no.4, pp. 39–47, 2006.

[6] M. H. Horng, T. W. Jiang, The artificial bee colony algorithm for vector quantization in image compression,*The 4th IEEE International Conference on Broadband Network and Multimedia Technology (IC-BNMT)*, pp. 319–323, 2011.

[7] H. A. S. Leitao, W. T. A. Lopes, F. Madeiro, PSO algorithm applied to codebook design for channel-optimized vector Quantization, *IEEE Latin America Transactions*, vol. 13, no. 4, pp. 961–967, 2015.

[8] S. Alkhalaf, O. Alfarraj, A. M. Hemeida, Fuzzy-VQ image compression based hybrid PSOGSA optimization algorithm, em IEEE International Conference on Fuzzy Systems (FUZZ-IEEE), pp. 1–6, 2015.

[9] C. C. Chang, Y. C. Li, and J. B. Yeh, Fast codebook search algorithms based on tree-structured vector quantization, Pattern Recognition Letters, vol. 27, no. 10, pp. 1077–1086, 2006.

[10] Z. Pan, K. Kotani, and T. Ohmi, An improved full-search-equivalent vector quantization method using the law of cosines, *IEEE Signal Processing Letters*, vol. 11, no. 2, pp. 247–250, 2004.

[11] J. Z. C. Lai, Y. C. Liaw, and J. Liu, A fast VQ codebook generation algorithm using codeword displacement, *Pattern Recognition*, vol. 41, no. 1, pp. 315-319, 2008.

[12] D. Lee, S. Baek, K. Sung, Modified K-means algorithm for vector quantizer design, IEEE Signal Processing Letters, vol. 4, no.1, pp. 2-4, 1997.

[13] K. K. Paliwal, V. Ramasubramanian, Comments on "modified K-means algorithm for vector quantizer design", *IEEE Transactions on Image Processing*, vol. 9, no.11, pp. 1964–1967, 2000.
[14] I. Katsavounidis, C. C. J. Kuo, Z. Zhang, A new initialization technique for generalized Lloyd iteration, *IEEE Signal Processing Letters*, vol. 1, no.10, pp. 144–146, 1994.
[15] G. Y. Kang, S. Z. Guo, D. C. Wang, L. H. Ma, Z. M. Lu, Image retrieval based on structured local binary kirsch pattern, *IEICE Transactions on Information and Systems*, vol. 96, no.5, pp. 1230–1232, 2013.