An Optimized Fast Codeword Search Algorithm for Vector Quantization

Mobin Jamali
Department of Electrical and Computer Engineering
Isfahan university of Technology
Isfahan, 84156-83111, IRAN
mobin.jamali@ec.iut.ac.ir

Vahid Ghafarinia
Department of Electrical and Computer Engineering
Isfahan university of Technology
Isfahan, 84156-83111, IRAN
ghafarinia@cc.iut.ac.ir

Mohammad Ali Montazeri
Department of Electrical and Computer Engineering
Isfahan university of Technology
Isfahan, 84156-83111, IRAN
montazeri@cc.iut.ac.ir

***Abstract* — Vector quantization is a popular data compression method with a widespread use in signal processing and pattern recognition applications. However its computational demand is increased tremendously as the size of date set and the dimension of samples grow larger. Here we have proposed a new fast-search algorithm to enhance the computational cost and the execution time of the conventional VQ method. This algorithm relies on the Chebyshev distance for distortion measurement and Triangular Inequality Elimination (TIE) for codeword elimination. The proposed algorithm was employed for the recognition of isolated spoken Persian digits. Results show that this algorithm can remarkably reduce the number of calculations while its encoding error is closed to the full search algorithm**.

Keywords— Vector Quantization, Codeword Search, Triangular Inequality Elimination

I. Introduction

Vector Quantization (VQ) is an efficient technique for data compression. It has been widely used in speech coding, pattern classification, image compression and watermarking. The large compression rate and fast coding by a simple algorithm are the main advantages of VQ have made it a popular compression method [1, 2].

Mapping from a k-dimension space, R^k, into a smaller finite subset with size N, called codebook, is the definition of quantizing process. By considering $C=\{y_1, y_2, ..., y_N\}$ as a given codebook with N members called codeword and $X=\{x_1, x_2, ..., x_k\}$ as an input vector with dimension k, the process of finding the best matched codeword $y_b=\{y_{b1}, y_{b2}, .., y_{bk}\}$ with input vector X so that the distortion between X and y_b yields to the minimum value is called codeword search [3]. The squared Euclidean distance given by Equation (1) is widely used as the matching distortion measure.

$$D(x, y_b) = \sum_{t=1}^{k} (x_t - y_{bt})^2 \quad (1)$$

It is clear that the complexity of distortion calculation will increase exponentially by vector dimension and the size of codebook. This drawback will be more critical in the real-time encoding applications. By assuming N *as* the codebook size and k as the vector dimension, there are $N\times k$ multiplications, $N\times(2k-1)$ additions and $N-1$ comparisons in the whole encoding process for one input vector. So the complexity of encoding process by utilizing Full Search (FS) algorithm will be considerably high and the encoding procedure will be time-consuming [4].

So far, several fast search algorithms have been proposed to enhance the basic VQ method. Reducing the searching time by the elimination of irrelevant codewords is the basic idea of these proposed algorithms. The elimination rule is set around the statistical characteristics of the codewords. Using the mean value of input vectors and codewords in Equal mean Nearest Neighbor Search (ENNS) algorithm [5] and using the variance of input vectors and codewords as well as the mean value in Equal mean Equal Variance Nearest Neighbor Search (EENNS) algorithms [6] are examples of fast search methods. Also, several improved and extended versions of ENNS techniques have been introduced in the literature [7-10]. All of these algorithms try to reject more unlikely codewords during the search process. The method proposed in [11] uses the Triangular Inequality Elimination (TIE) as the criterion to eliminate the unlikely codewords and alleviate the computation volume. Partial Distortion Search (PDS) is a simple and efficient algorithm which doesn't uses specific elimination condition and is often used in the last step of distortion calculation. PDS first calculates the partial distortion between codeword and input vector and uses it as a premature termination condition for the remaining calculations [12].

Neighbor To Neighbor (NTN) is another fast codeword search algorithm which reduces the computation time and eliminates the codewords using the similarity of contiguous input vectors as a rejection criterion [13]. The HTCP algorithm [4] uses Chebyshev distance measure with less computation complexity in Hadamard domain instead of Euclidean distance to find the best matched codewords. NTN and HTCP are approximation-based algorithms in which the dependence of results on a tuning parameter is the main drawback. Incorrect selection of user-dependent parameters will affect the resulting quantization error and increase it more than that of the conventional full search algorithm.

All of the proposed fast search algorithms try to reject more unrelated codewords during search in order to reduce the numbers of calculations. However the computation burden of the codeword rejection process is an important issue which should be noted. Regarding this issue, here, we have introduced an optimized codeword search algorithm based on the low cost HTCP algorithm and the elimination rule proposed in [11]. Our algorithm benefits from the Chebyshev distortion measure with less computations and an elimination criterion based on TIE. In this algorithm each input vector is encoded based on the contiguous vectors similarity in the input vectors. The compare the performance of the proposed algorithm with its counterparts it was employed for the recognition of isolated spoken Persian digits. The results show that the proposed method is able to reject a prominent number of codewords and hence reducing the required calculations while keeping the error rate close to the full search method.

II. Related Works

This section briefly describes the HTCP [4] and TIE-based algorithms [11] which are the basis of the new algorithm.

A. HTCP algorithm

The HTCP is a fast codeword search algorithm that is constructed based on Hadamard transform (HT), Chebyshev distance and PDS. The best matched codeword in HTCP is found in the HT domain. There is a direct relation between the energy of both the original signal and transformed one. According to this theorem, matching the codeword in the spatial domain and the HT domain will give the same result. In addition, because of the elements of Hadamard matrix which are '1' or '-1', transforming the signal can be performed simply by addition and subtractions without the need to any multiplication.

Finding the best matched codewords per each input vector in HTCP is based on two judgments. First, during the search in the HT domain, Chebyshev distances between the transformed input vector and all of the transformed codewords is measured by (2) where X and Y are transformed input vector and transformed codeword respectively.

$$D(X,Y) = \max_{0 \le i \le k-1} | X_i - Y_i | \tag{2}$$

Then L of the codewords producing the minimum distance is founded and the rest are excluded. These L remained codewords are somehow the estimated best ones. Second, PDS algorithm is performed on the L remaining codewords to find the best matched codeword with the minimum MSE. It should be noted that if the value of L is big enough, the possibility of including the best matched codeword with minimum MSE in the remaining L codewords, is quite high.

B. TIE-based algorithm

The algorithm proposed in [11] utilizes the property of correlations between consecutive input vectors. According to this property, the codewords in neighborhood of best matched codeword of previous vector will have the high probability of picking for current vector. The neighborhood of codewords to each other is determined by distortions calculation between them which is stored in an external memory called transition table. All of the elements in the transition table are sorted into increasing order and its elements are replaced with indices of codewords associated to them. TIE-based algorithm also uses the TIE to reject unlikely cases in the neighboring codewords. By assuming y_a and y_b as two arbitrary codewords and x as an input vector, the TIE is given by (3).

$$\begin{aligned} &if \;\; D(y_a, y_b) \ge 2 \times D(x, y_a) \\ &then \; D(x, y_b) \ge D(x, y_a) \end{aligned} \tag{3}$$

The induction of TIE-based algorithm is as follows: after finding the best matched codeword y_a with previous input vector x, the distortion between y_a and current vector x is calculated as *D_min*. Then, the TIE algorithm is performed on the current vector x, y_a and nearest neighbor codeword of y_a called y_b that is selected from transition table. If y_b doesn't obey (3), y_a will be the best matched codeword with current vector x. Else the distortion between y_b and input vector x is calculated as D_min_2 and *D_min* is updated with $min\{D\_min, D\_min_2\}$. This procedure is repeated for y_b and its related neighbors and will continue until TIE doesn't satisfy. Finally, the best matched codeword related to the current input vector is found. The same procedure is repeated with this codeword as the start point for the next input vector.

III. The Proposed Algorithm

The main drawback of HTCP is the encoding of each input vector separately only based on the Chebyshev distance neglecting the similarity of neighbor vectors. Thus, some of the best matched codewords may be wrongly rejected. Another shortcoming of HTCP is the need to calculate all of the Chebyshev distances because no elimination rule is used. Besides, the TIE-based search algorithms have the advantage of using elimination inequality during search. However, this algorithm hasn't specific defect which leads to dysfunction. The appropriate combination of HTCP and TIE will give a simpler and more efficient method.

As discussed in the previous section, to reduce the number of calculation during search per each input vector, the more dissimilar codewords should be detected and excluded. Although, the similarity measure that is required by rejection rule, is another factor that influences the volume of calculations. To take account of these two factors, in the proposed method, we have coupled the advantages of HTCP and TIE algorithms. The result is a simple algorithm which can reduces the number of irrelevant codewords by fewer computations. The proposed algorithm is performed in two main steps which are an off-line preprocessing for constructing the transition table and an On-line processing for codeword elimination.

\- *Off-line preprocessing*

1) The Chebyshev distances between all pairs of codewords are calculated and sorted in ascending order in the so called distortions table. The transition table is then constructed by replacing all elements of distortion table with the indices of codewords associated to them. So that i^{th} column of the transition table indicates the codewords in the neighborhood of the i^{th} codeword in the incremental order of the Chebyshev distances.

- *On-line processing*

1) Set $i=1$ and calculate the Chebyshev distances between X_i and all codewords.

2) Store the L number of minimum obtained distances. Find the codeword C_J with minimum MSE in these L codewords and determine it as the best matched one with X_i. Then set $i=i+1$ and go to step 3.

3) Calculate the Chebyshev distance $D_{min} = d(X_i,C_J)$ and go to J^{th} column of transition table. Put the $d(X_i,C_J)$ in the temporal memory called LIST and go to step 4.

4) Set the first element of J^{th} column in the transitions table which indicates the nearest neighbor codeword to the C_J as C_b. Then Go to step 5 and Check the C_b in the TIE.

5) If $d(C_b,C_J)>2\times D_{min}$, go to step 8 otherwise go to step 6.

6) Calculate $D_{min2} = d(X_i,C_b)$. If $D_{min2} < D_{min}$ then set $J=b$, which means that the candidate column is changed. Then put the $d(X_i,C_b)$ in the LIST, update $D_{min} = D_{min2}$ and go to step 4, else go to step 7.

7) Put the $d(X_i,C_b)$ in the LIST and set the next element of J^{th} column as C_b and go to step 5.

8) Find L number of minimum distances in the LIST and calculate the MSE of the codewords associated with these L members. The codeword with minimum MSE will be the best matched one with X_i. Set this codeword as C_J. If i is the last input vector, terminate the algorithm, else set $i=i+1$ and go to step 3.

Notice: if the selected codeword C_b in the transitions table is repetitive, this codeword is ignored and the next element after that is considered.

According to the described steps, unlike the HTCP algorithm, the proposed approach doesn't calculates the Chebyshev distance for all of the codewords. This optimization remarkably affects the calculations volume. Also, in this algorithm the resulting matched codeword with each input vector is the starting point in performing the algorithm on the next vector. This dependency certainly affects the overall encoding MSE of the whole input vectors. Unlike the algorithm proposed in [11], because of using Chebyshev distortion measure during elimination process, the new method will be of low cost and less computations.

IV. SIMULATION RESULTS

In our experiments, we have simulated the encoding algorithms in the field of speech signal coding. The speech data that used in our simulations is the Mell Frequency cepstral coefficients (MFCC) vectors of isolated words associated with Persian digits one to five. The codebook related to each word is generated by 60 training utterances and using classic LBG clustering algorithm [2]. The codebook has been generated in two sizes of 16 and 32 with dimension 14. In the testing step, 60 utterances of a specific word and dissimilar to the training utterances are encoded using the codebook of the same word associated to it. The average value of evaluated parameter between 60 utterances is presented as the results in all tables. The performance of the proposed algorithm have been compared with the conventional FS, NTN [13], the algorithm proposed in [10] as the best version of ENNS-based method, HTCP [4] and TIE-based algorithm proposed in [11]. The average number of eliminated codewords during search is presented in table 1. The parameter L in the proposed algorithm and HTCP and the input parameter of NTN have been set to 4, 4 and 0.6 respectively. These numbers are selected for a fair comparison regarding the computational cost of the algorithms.

TABLE1. AVERAGE NUMBER OF ELIMINATED CODEWORDS.

Codebook Size	Algorithm	Word				
		1 /yek/	2 /dow/	3 /seh/	4 /chahar/	5 /panj/
16	FS	0	0	0	0	0
	[11]	8.17	7.06	6.94	6.24	7.67
	NTN	11.67	11.42	11.42	10.77	11.74
	HTCP	12	12	12	12	12
	[10]	12.4412	12.2972	12.23	12.28	12.29
	Proposed	12.481	12.1795	12.13	12.11	12.37
32	FS	0	0	0	0	0
	[11]	18.79	17.2705	17.5159	17.7403	17.9632
	NTN	23.5207	23.4387	23.47	22.9292	23.7944
	HTCP	28	28	28	28	28
	[10]	27.754	27.6139	27.3967	27.6348	27.54
	Proposed	28.145	28.192	28.147	28.138	28.129

According to the results presented in the table 1, the number of eliminated codewords in the new algorithm is higher than other methods. Although, the algorithm proposed in [10] shows similar performance to the new method and can be considered as the main challenger. To compare the computational cost of the algorithms, the average number of multiplications and additions operations per each word's encoding process is presented in table 2 and table 3 respectively.

TABLE 2. AVERAGE NUMBER OF MULTIPLICATION OPERATIONS.

Codebook Size	Algorithm	Word				
		1 /yek/	2 /dow/	3 /seh/	4 /chahar/	5 /panj/
16	FS	8310	68917	6846	9691	8560
	[11]	4069	3839	3839	5867	4435
	NTN	3779	3236	3217	4952	3859
	HTCP	2077	1723	1711	2423	2140
	[10]	4964	4617	4410	6598	5440
	Proposed	1865	1682	1690	2406	1982
32	FS	16621	13783	13694	19383	17121
	[11]	6824	6278	6086	8602	7468
	NTN	6487	5396	5346	7941	6553
	HTCP	2077	1723	1711	2423	2140
	[10]	7602	6943	6482	9470	8312
	Proposed	2036	1662	1670	2379	2105

TABLE 3. AVERAGE NUMBER OF ADDITIONS OPERATIONS.

Codebook Size	Algorithm	Word				
		1 /yek/	2 /dow/	3 /seh/	4 /chahar/	5 /panj/
16	FS	16027	13291	13205	18691	16510
	[11]	7906	7450	7450	11381	8613
	NTN	6786	5827	5793	8963	6925
	HTCP	22408	18583	18462	26133	23083
	[10]	10356	9580	9164	13679	11307
	Proposed	6740	5804	5753	8678	6553
32	FS	32054	26582	26410	37382	33019
	[11]	13221	12159	11789	16661	14468
	NTN	11471	9550	9460	14098	11657
	HTCP	78800	65348	64924	91898	81172
	[10]	15787	14376	13449	19634	17207
	Proposed	7988	7217	7060	9807	8799

The number of required mathematical calculations indicates that most of the introduced algorithms neglect the increment of calculations burden in the performing of their methods. According to the results in table 2 and table 3, the proposed method has remarkably less computations than the other algorithms. That means the proposed method is able to reject a large number of codewords while keeping the computations too low. This improvement is basically depending on using Chebyshev distortion measure and TIE together. Based on these results, the optimality of the new algorithm than the other ones, especially the method of [10] which has the high computations cost, is obvious. Another issue from table 3 is that the number of additions operations in HTCP surpasses the FS algorithm. This is caused by the Hadamard transform in this method. As described in section II, transforming the signal is performed by addition and subtractions operations.

The mean square error of quantization of the investigated search algorithms is compared with the FS method and the results are presented in table 4. Because the MSE of methods proposed in [10] and [11] is the same as FS algorithm, table 4 includes only NTN and HTCP algorithms as the approximation-based methods. To show the effect of algorithm parameter, the results of table 4 includes three different values of L. Also the three different values are determined for the input parameter of HTCP and NTN methods. According to the results presented in table 4, firstly, by increasing the parameter L in the proposed algorithm, MSE will be decreased which means that the probability of wrongly elimination of the best matched codeword will be decreased. Secondly, according to table 4 the new algorithm is more successful than the others in reducing MSE while keeping the calculations fairly low. This improvement is basically depends on using the contiguous vectors similarity in the input vectors.

TABLE 4. MEAN SQUARE ERROR OF QUANTIZATION.

Codebook Size	Algorithm		Sound			
			1/yek/	2/dow/	3/seh/	4/chahar/
16	FS		9.9689	8.6620	8.1495	10.7946
	NTN	α =0.5	11.560	9.2854	9.2839	10.9279
		α =0.6	10.319	8.7840	8.4845	10.8217
		α =0.7	10.008	8.6794	8.1945	10.8020
	HTCP	L=3	10.047	8.7794	8.3149	10.9685
		L=4	10.004	8.7084	8.2044	10.8379
		L=5	9.9854	8.677	8.1576	10.8168
	Proposed	L=3	9.9826	8.7356	8.1679	10.8519
		L=4	9.9758	8.6932	8.1576	10.8108
		L=5	9.9747	8.679	8.1560	10.8014
32	FS		8.1024	6.7332	5.9731	6.7286
	NTN	α=0.5	9.9406	7.4824	7.3684	6.9573
		α =0.6	8.5865	6.9169	6.3847	6.7866
		α =0.7	8.1477	6.7686	6.0396	6.7350
	HTCP	L=3	8.2210	6.8466	6.1018	6.8456
		L=4	8.1505	6.8067	6.0478	6.7986
		L=5	8.1248	6.7748	6.0015	6.7598
	Proposed	L=3	8.1592	6.7981	6.0161	6.7972
		L=4	8.1217	6.7630	5.9970	6.7640
		L=5	8.1170	6.7468	5.9841	6.7521

V. CONCLUSION

Fast search algorithms were analyzed based on their computational complexity. It was revealed that the search performance can be improved beyond the available algorithms by coupling the advantages of these algorithms. Based on the results, Chebyshev distance and TIE were used to simultaneously enhance the distortion measurement and codeword exclusion. The proposed fast search algorithm was employed for the recognition of spoken Persian digits. The results show that the number of calculations of the proposed method is at least 2 to 8 times lower than the conventional full search algorithm while the encoding error of the two algorithms is nearly similar. Another important result is that the computational volume of the proposed algorithm is loosely dependent on the size of codebook. While the number of calculations required by other methods largely increases by the size of the codebook, it is slightly increases for the proposed algorithm.

REFERENCES

[1] R. Gray, "Vector quantization", IEEE ASSP Mag., vol. 1, no. 2, pp. 4–29, Apr. 1984.

[2] Y. Linde, a. Buzo, and R. Gray, "An Algorithm for Vector Quantizer Design", IEEE Trans. Commun., vol. 28, no. 1, pp. 84–95, 1980.

[3] J. Deller, J. Proakis, and J. Hansen, Discrete-time processing of speech signals. IEEE PRESS, 2000.

[4] E. Dong and G. Cai, "An Improved Codeword Search Algorithm Based on Hadamard Transform", Signal Processing, 8th International Conference on, vol. 2, 2006.

[5] S.-W. Ra and J.-K. Kim, “A Fast Mean-Distance-Ordered Partial Codebook Search Algorithm for Image Vector Quantization”, IEEE Trans. Circuits Syst. - II Analog Digit. Signal Process., vol. 49, no. 9, pp. 576–579, 1993.

[6] C.-H. Lee, L-H. Chen “Fast closest codeword search algorithms for vector quantisation”, IEE Proc. - Vision, Image, Signal Process., vol. 141, no. 3, p. 143, 1994.

[7] Z. Pan, K. Kotani, and T. Ohmi, “Subvector-Based Fast Encoding Method for Vector Quantization Without Using Two Partial Variances”, Opt. Rev., vol. 13, no. 6, pp. 410–416, 2006.

[8] J.-S. Pan, Z.-M. Lu, and S.-H. Sun, “An efficient encoding algorithm for vector quantization based on subvector technique”, IEEE Trans. Image Process., vol. 12, no. 3, pp. 265–70, Jan. 2003.

[9] J. S. PAN, K. C. HUANG, “A New Vector Quantization Image Coding Algorithm Based on The Extension of The Bound for Minkowski Metric”, Pattern Recognit, vol. 31, no. 11, pp. 1757–1760, 1998.

[10] X. Linbo, Y. Wei, and H. Bang, “An Improved Codeword Search Algorithm Based on Subvector Features”, Fourth International Conference on Intelligent Computation Technology and Automation, vol. 1, pp. 443–446, 2011.

[11] S. Chen and J. Pan, “Fast search algorithm for VQ-based recognition of isolated words”, IEE Proc. I Commun. Speech Vis., vol. 136, no. 6, pp. 391–396, 1989.

[12] B. Chang-Da and R. Gray, “An improvement of the minimum distortion encoding algorithm for vector quantization”, IEEE Trans. Commun., pp. 1132–1133, 1985.

[13] N. Inoue and K. Shinoda, “Neighbor-to-Neighbor Search for Fast Coding of Feature Vectors”, 2013 IEEE Int. Conf. Comput. Vis., pp. 1233–1240, Dec. 2013.