Binary Tree Based Fast Search Algorithm for Closest Codevector in the Codebook for Vector Quantization

H. B. Kekre and Tanuja K. Sarode

Abstract: **Vector Quantization is a powerful tool for data compression. One of the most serious problems for vector quantization is the high computational complexity of searching for the closest codevector in the codebook during the encoding phase. This paper proposes an algorithm to search for the closest codevector in which codevectors are arranged in the binary tree. The search algorithms and the issues related to search algorithms are studied and implemented. It is observed that proposed algorithm is faster than other fast algorithms, although it gives slightly more mean squared error (mse) than the sequential search algorithm but it is in the acceptable range.**

***Index terms*:- Vector Quantization, Codevector, Codebook, Fast Search.**

I. Introduction

Vector quantization(VQ) technique is very efficient for image coding not only because it improves compression ratio but also simplifies computational complexity. VQ has received great attention over last decade because of it's promising compression ratio and relatively simpler structure[7]-[24].

VQ can be defined as a mapping of k-dimensional Euclidean space R^k to into a finite subset C of R^k. The finite set C is known as the codebook and C = {c_i / i = 1, 2, ….. , N } where c_i is a codeword and N is the size of codebook.

Codebook can be generated using clustering algorithms or using transform domain techniques [25]-[27].

Another aspect of vector quantization is an encoding phase that requires an exhaustive search of codebook. The search algorithms and the issues related to search algorithms are studied and implemented on 10 facial images and the results are compared. All the search algorithms are implemented on the global codebook prepared from 10 facial images using clustering algorithm.

Dr. H. B. Kekre, Professor, Department of Computer Engineering, Thadomal Shahani Engineering College, Bandra (west) , Mumbai 50, India,
Email : hbkekre@yahoo.com,

Tanuja K. Sarode, Assistant Professor, Department of Computer Engineering, Thadomal Shahani Engineering College, Bandra (west) , Mumbai 50, India,
Email: tanuja_0123@yahoo.com

II. EXHAUSTIVE SEARCH ALGORITHM[1]

Exhaustive search algorithm is also called as sequential search algorithm.

Let codebook contain N codevectors with dimension k, each codevector is associated with an index i.

The given input vector is encoded by searching codevector with minimal distortion. Compute the squared Euclidean distance between the given input vector and the first codevector of the codebook and assign this square difference to d^2_{min} and the index of first codevector as minimal distortion index.

Then for each codevector the distortion is computed and if the current distortion is smaller than the minimal distortion i.e. d^2_{min}, then assign the current distortion to the d^2_{min} and the index of codevector as minimal distortion index. Thus after N calculations the index for minimum distortion is found. Decoding is then to look up in the codebook for given index i the associated codevector.

If the size of the codebook is **N**, then the full codebook search requires Nk multipications, **N(2k-1)** additions and **N-1** comparisons for each k-dimensional training vector. The time required for quantization of an image is excessively large and hence there is a need for fast search algorithm.

III. Some Existing Fast Closest Codeword Search Algorithms

A. *Partial Distortion Elimination Algorithm (PDE)[1,3]*

The PDE algorithm allows the early termination of the distortion calculation between a training vector and a codeword by introducing a premature exit condition in the search process.

For each training vector x, the algorithm first calculates the distortion between x and an arbitrary codeword and takes this distortion as the current minimum distortion d^2_{min}. Then, for any other codeword $y_i = \{y_{i1}, y_{i2},\dots\dots\dots\dots,y_{ik}\}$, if there exists q<k with the accumulated distortion for the first q samples as in equation below,

$$d^2(x, y_j) = \sum_{n=1}^{k}(x_n - y_{jn})^2 \quad (1)$$

larger than the current minimum distortion d^2_{min}, i.e. $\sum_{n=1}^{q}(x_n - y_{jn})^2 \geq d^2_{min}$ this algorithm stops computing

the distortion for codeword y_i and begins trying the next codeword. This will reduce **(k-q)** multiplications and **2(k-q)** additions, per vector but will have more mse.

B. Partial search partial distortion algorithm (PSPD)[1,4]

PSPD algorithm builds up a partial codebook based on the mean value m_x of a k-dimensional training vector $x = \{x_1, x_2,\dots\dots\dots\dots,x_k\}$ in which m_x is defined as m_x = integer part of $[\,1/k \sum x_j + 0.5]$ The algorithm then uses the PDE method to search the partial codebook for the closest codeword.

The PSPD algorithm first calculates the mean values of all codewords and sorts the codebook according to increasing order of the codeword means. For each training vector, it then finds the codeword y_p with minimum mean difference to the training vector. The PDE method is then employed to find the closest codeword in this partial codebook form the codevector y_{p+1} to the last codevector in the codebook.

Sometimes the closest codeword may not be located in the partial codebook, this will introduce more distortion than sequential search.

C. Fast Nearest Neighbour Search Algorithm (FNNS)[1,5]

FNNS algorithm uses triangular inequality to reject a great many unlikely codewords. For a vector x, it first finds a probably nearby codeword y_i with distortion $d^2(x, y_i)$. This algorithm then eliminates those codewords, which are impossible to be closest codewords, based on the triangle inequality and a pre-computed table, which contains the distances of all pairs of codewords.

That is, for each codeword y_j, if

$d(y_j, y_i) > 2d(x, y_i)$

Through the triangular inequality, we have

$d(x, y_j) + d(x, y_i) \geq d(y_j, y_i) > 2d(x, y_i)$

The above inequality can be reduced to

$d(x, y_j) > d(x, y_i)$

Therefore those codevector with distances to y_i larger than $2d(x, y_i)$ will be eliminated from consideration to be a candidate of the closest codeword.

Let N be the size of codebook.

Let map(p, j) be the Euclidean distance between p^{th} codevector and j^{th} codevector

Let d_{min} be the Euclidean distance between 1^{st} codevector and the training vector x.

Let p=1

For j=2 to N

begin

If $2d_{min}$ <= map(p,j)

Then compute d=Euclidean distance between j^{th} codevector and the training vector x.

If d<=d_{min} then

Assign d_{min}=d and p=j

End.

Repeat the above for loop for all training vectors.

This algorithm requires a table of size $N^2/2$ to store the distances of all pairs of codevectors. When N is large, the memory requirement is a serious problem.

IV. PROPOSED ALGORITHM

Codevectors are arranged in a binary tree structure, at each iteration during the formation of codebook. Clustering algorithm is used to generate the codebook.

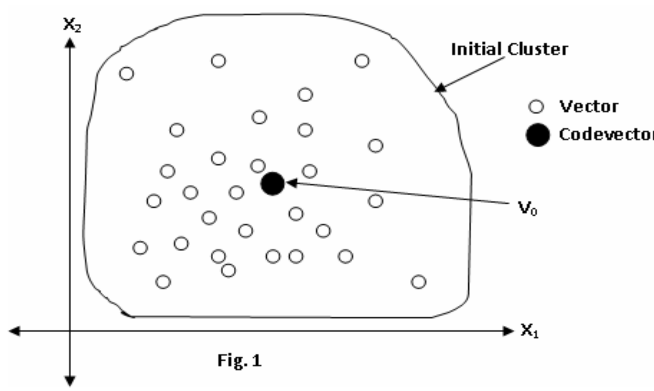

Fig. 1

Clustering algorithm[2] is an iterative algorithm; in the 1^{st} iteration we get one cluster and the corresponding codevector V_0 as shown in the fig. 1. This codevector V_0 forms the root of the binary tree at level 0.

In the 2^{nd} iteration two clusters are generated and hence two codevectors V_{00} and V_{01} from the codevector V_0, as shown the fig. 2. Codevectors V_{00} and V_{01} are the children of the codevector V_0, as shown in the fig. 3 at level 1 of the tree.

In the 3^{rd} iteration four clusters are generated and hence four codevectors V_{000}, V_{001}, V_{010} and V_{011}. The codevectors V_{000} and V_{001} are the children of the codevector V_{00} and codevectors V_{010} and V_{011} are the children of codevector V_{01} as shown the fig. 3. The process is repeated until we get codebook of desired size or desired mse is obtained.

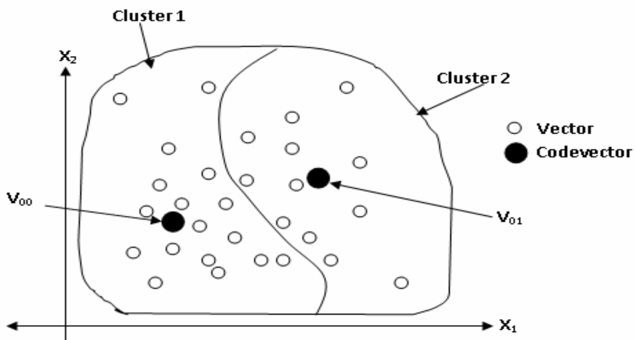

Fig. 2

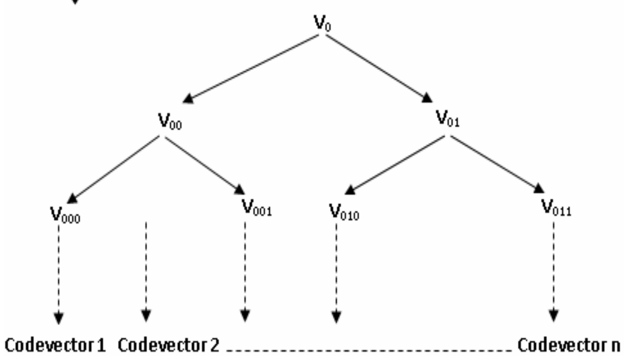

Fig. 3

For the codebook of size 256 final codevectors will be at the level 8.

The codevector at level 0 forms the current parent node. For each input vector squared Euclidean distances between the two codevectors at level 1 (i.e with the children's of the current parent node) are computed and the square distances are compared with each other. The codevector, which gives smaller squared distance, is then marked as current parent node and then again the squared distance between the children's of the current parent node is computed. The process is repeated till final codevector at the level n is reached. At any level if the squared distance happens to be same, then the comparison is made by computing the mean of each codevector. For the tree with height n+1 (nth level) total $2*\log_2 n$ Euclidean distances are computed and $\log_2 n$ comparisons are done.

The only drawback of this algorithm is it requires to store codebook in a binary tree from. This requires (2N)-1 memory locations as compared to $N^2/2$ for FNNS where N is the size of codebook.

V. RESULTS

Among the various applications of VQ we have chosen image compression as the application. All the search algorithms are implemented on 100x100 10 facial images and

are implemented using Matlab 6.0 R12 on Pentium IV 2.4 GHz 256 MB RAM.

Fig 4.a and Fig 4.b shows the average execution in seconds for Gray Scale and color images respectively. It is observed that the proposed algorithm takes lesser time to execute as compared to the other search algorithms. It is observed that proposed algorithm is 89.72% faster than sequential search for gray scale images and 92% faster than sequential search for color images Fig 5 shows the comparison of search algorithms with respect to MSE on 24-bit color bmp images kaiu, tanuja and shardul2 of size 100x100.

Table 1 shows the MSE value for the 10 different 100x100 color images for the search algorithms Sequential Search, PSPD, PDE, FNNS and Proposed. It is observed that proposed algorithms gives less MSE as compared to the FNNS algorithm.

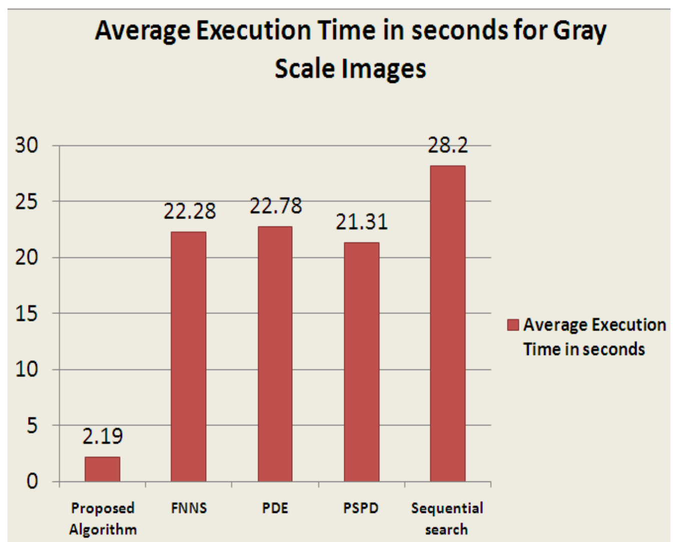

Fig 4.a

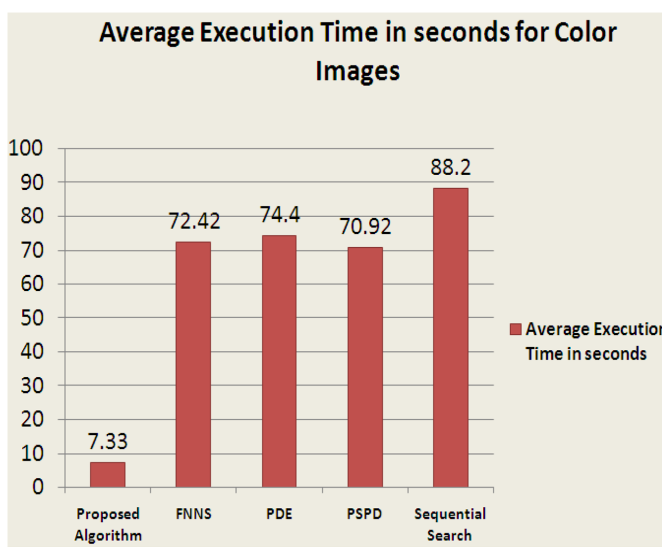

Fig 4.b

TABLE 1 RESULTS OF SEARCH ALGORITHMS ON COLOR IMAGES OF SIZE 100X100

Search Algorithms →	Proposed Algo.	FNNS	PDE	PSPD	Seq. Search
List of bmp Images ↓	**MSE**	**MSE**	**MSE**	**MSE**	**MSE**
Kaiu	**83.99**	**258.3**	**32.82**	**32.82**	**32.82**
Sharmila	**53.17**	**314.1**	**36.16**	**36.16**	**36.16**
Tanuja	**247.8**	**379.7**	**100.4**	**100.5**	**100.4**
Vaijayanti	**117.3**	**244.4**	**68.37**	**68.38**	**68.37**
Shardul1	**78.60**	**112.0**	**34.31**	**35.82**	**34.31**
Shardul2	**81.05**	**152.0**	**55.47**	**55.68**	**55.47**
Ash	**111.5**	**178.8**	**46.41**	**46.42**	**46.41**
Kate	**145.7**	**67.47**	**52.83**	**52.83**	**52.83**
Sonali	**104.9**	**103.9**	**38.48**	**38.48**	**38.48**
Urmila	**80.54**	**217.3**	**27.97**	**27.97**	**27.97**

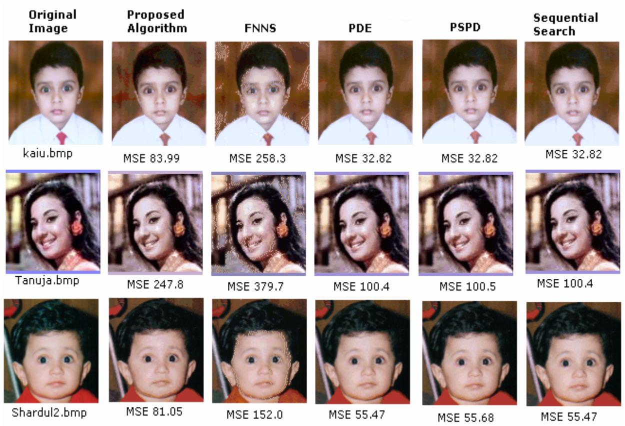

Fig 5.

VI. CONCLUSION

Among the search algorithms
1. Sequential search,
2. Partial distortion elimination algorithm (PDE)
3. Partial search partial distortion algorithm (PSPD
4. Fast nearest neighbor search algorithm (FNNS) and
5. Proposed Algorithm.

It is found that in most of the cases **FNNS** algorithm gives very **high mse** compared to other search algorithms. Also it is observed that **Proposed** algorithm is the **faster** algorithm though it gives slightly more mean squared error (mse) than the sequential search algorithm but it is in the acceptable range. Where as PDE and PSPDE gives less mse than the Proposed algorithm but they take larger execution time as compared to the proposed algorithm as shown in the results.

VII. REFERENCES

[1] C. H. LEE, L. H. CHEN, “Fast Codeword Search Algorithm for Vector Quantization” IEE Proceedings Image Signal Processing vol. 141, No. 3 June 1994.

[2] Y. LINDE, A. BUZO, and R. M. GRAY, "An Algorithm for Vector Quantizer Design," IEEE Transactions on Communications, pp. 702--710, January 1980.

[3] BEI, C.D., and GRAY, " An improvement of the minimum distortion encoding algorithm for vector quantization" , IEEE Trans. 1985, C-33, pp 1132-1133.

[4] HSIEH, C.H., LU, P.C. and CHANG, J.C., "Fast codebook generation algorithm for vector quantization of images", Pattern Recognition Lett. 1991,vol. 12, pp 605-609

[5] ORCHARD, M.D., "A fast nearest neighbour search algorithm", IEEE ICASSP, 1991, pp 2297-2300.

[6] Nam Phamdo, ”Vector Quantisation”, http://www.data-compression.com.

[7] Y.C. Liaw, J.Z.C. Lai, W. ,“Lo, Image restoration of compressed image using classified vector quantization”, Pattern Recogn. vol. 35 No. 2 ,pp 181–192, 2002.

[8] N. M. Nasrabadi, Y. Feng, ,”Image compression using address vector quantization”, IEEE Trans. Commun. vol. 38 No. 12 pp 2166–2173, 1990.

[9] J. Foster, R.M. Gray, M.O. Dunham, “Finite state vector quantization for waveform coding”, IEEE Trans. Inf. Theory vol. 31 No. 3 pp 348–359, 1985.

[10] T. Kim, “Side match and overlap match vector quantizers for images”, IEEE Trans. Image Process. vol. 1 No. 2 pp 170–185, April 1992.

[11] J.Z.C. Lai, Y.C. Liaw, W. ,”Lo, Artifact reduction of JPEG coded images using mean-removed classified vector quantization”, Signal Process. 82 vol. 10 pp 1375–1388, 2002.

[12] K.N. Ngan, H.C. Koh, ,”Predictive classified vector quantization”, IEEE Trans. Image Process. vol. 1 No.3 pp 269–280, 1992.

[13] C.H. Hsieh, J.C. Tsai, ”Lossless compression of VQ index with searchorder coding”, IEEE Trans. Image Process. vol. 5 No. 11 pp 1579–1582, 1996.

[14] J. Z. C. Lai, J.Y. Yen, “Inverse error-diffusion using classified vector quantization”, IEEE Trans. Image Process. vol. 7 No. 12 pp 1753–1758, 1998.

[15] P.C. Chang, C.S. Yu, T.H. Lee, “Hybrid LMS-MMSE inverse halftoning technique”, IEEE Trans. Image Process. vol. 10 No. 1 pp 95–103, 2001.

[16] A. Gersho, R.M. Gray, Vector Quantization and Signal Compression, Kluwer Academic Publishers, Boston, MA, 1991.

[17] Y. Linde, A. Buzo, R.M. Gray, “An algorithm for vector quantizer design”, IEEE Trans. Commun. vol. 28 No. 1 pp 84–95, 1980.

[18] W.H. Equitz, “A new vector quantization clustering algorithm”, IEEE Trans. coust. Speech Signal Process. vol. 37 No. 10 pp 1568–1575, 1989.

[19] C.-K. Chan, L.-M. Po, “A complexity reduction technique for image vector quantization”, IEEE Trans. Image Process. vol. 1 No. 3 pp 312–321, 1992.

[20] C.M. Huang, R.W. Harris, “A comparison of several vector quantization codebook generation approaches”, IEEE Trans. Image Process. vol. 2 No. 1 pp 108–112, 1993.

[21] J.Z.C. Lai, C.C. Lue, “Fast search algorithms for VQ codebook generation”, J. Vis. Comm. Image Represent. vol. 7 No. 2 pp 163–168, 1996.

[22] C.M. Huang, Q. Bi, G.S. Stiles, R.W. Harris, “Fast full search equivalent encoding algorithms for image compression using vector quantization”, IEEE Trans. Image Process. vol. 1 No. 3 pp 413–416, 1992.

[23] T. Kaukoranta, P. Fränti, O. Nevalainen, “A fast exact GLA based code vector activity detection”, IEEE Trans. Image Process. vol. 9 No. 8 pp 1337–1342, 2000.

[24] S.W. Ra, J.K. Kim, “Fast mean-distance-ordered partial codebook search algorithm for image vector quantization”, IEEE Trans. Circuits Syst. II 40 vol. 9 pp 576–579, 1993 .

[25] Momotaz Begum, Nurun Nahar, Kaneez Fatimah, M. K. Hasan, and M. A. Rahaman, “An Efficient Algorithm for Codebook Design in Transform Vector Quantization”, *WSCG’2003*, February 3-7, 2003.

[26] Robert Li and Jung Kim,” Image Compression Using Fast Transformed Vector Quantization”,*IEEE Applied Imagery Pattern Recognition Workshop, 2000 Proceedings* 29th Volume , Page(s):141 – 145, 2000.

[27] Zhibin Pan; Kotani, K.; Ohmi, T., ” Enhanced fast encoding method for vector quantization by finding an optimally-ordered Walsh transform kernel”, *ICIP 2005, IEEE International Conference*, Volume 1, Issue, 11-14 Page(s): I – 573, Sept. 2005.

Dr. H. B. Kekre has received B.E. (Hons.) in Telecomm. Engineering from Jabalpur University in 1958, M.Tech (Industrial Electronics) from IIT Bombay in 1960, M.S. Engineering (Electrical Engineering) from University of Ottawa in 1965 and Ph.D.(System Identification) from IIT Bombay in 1970. He has worked Over 35 years as Faculty and HOD Computer science and Engineering at IIT Bombay. From last 11 years working as a Professor in Department of Computer Engineering at Thadomal Shahani Engineering College, Mumbai. His areas of interest are Digital Signal Processing and Image Processing. He has more than 145 papers in National/International Conferences/Journals on his credit.

Tanuja K. Sarode has Received M.E. (Computer Engineering) degree from Mumbai University in 2004, currently Perusing Ph.D. from NMIMS Deemed University. She has more than 8 years of experience in teaching. Currently working as a Assistant Professor in Department of Computer Engineering at Thadomal Shahani Engineering College, Mumbai. Her areas of interest are Image Processing, Signal Processing and Computer Graphics. She has 6 papers in National /International Conferences on her credit.