Hadamard Transform Based Equal-average Equal-variance Equal-norm Nearest Neighbor Codeword Search Algorithm

Shu-Chuan Chu and John F. Roddick
School of Informatics and Engineering
Flinders University
Adelaide, Australia
Email: {shuchuan.chu,roddick} @infoeng.flinders.edu.au

Zhe-Ming Lu
Department of Automatic Test and Control
Harbin Institute of Technology
Harbin, China
Email: zhemingl@yahoo.com

Jeng-Shyang Pan
Department of Electronic Engineering, Kaohsiung University of Applied Sciences
Kaohsiung, Taiwan
Email: jspan@cc.kuas.edu.tw

Abstract

This paper presents a novel efficient nearest neighbor codeword search algorithm based on three elimination criteria in Hadamard transform (HT) domain. Before the search process, all codewords in the codebook are Hadamard-transformed and sorted in the ascending order of their first elements. During the search process, we firstly perform the HT on the input vector and calculate its variance and norm, and secondly exploit three efficient elimination criteria to find the nearest codeword to the input vector using the up down search mechanism near the initial best-match codeword. Experimental results demonstrate the performance of the proposed algorithm is much better than that of most existing nearest neighbor codeword search algorithms, especially in the case of high dimension.

1. Introduction

Vector quantization (VQ) has been widely used in image compression and speech coding [1]. The basic idea of VQ is exploiting the statistical dependency among vector components to obtain a high compression ratio. The task of codeword search is to search the best match codeword $c_j = (c_{j1}, c_{j2}, \cdots, c_{jk})$ from the given codebook $C = \{c_1, c_2, \cdots, c_N\}$ for the input vector $x = (x_1, x_2, \cdots, x_k)$ such that the distortion between this codeword and the input vector is the smallest among all codewords, where N is the codebook size and k is the vector dimension. The most common measure of distortion between x and c_i is the squared Euclidean distance, i.e.,

$$d(x, c_i) = \sum_{l=1}^{k} (x_l - c_{il})^2 \tag{1}$$

From the above equation, we can see that the *full search* (FS) algorithm requires kN multiplications, $(2k-1)N$ additions and $N-1$ comparisons to encode each input vector. If the VQ system possesses large codebook size and high dimension, the computation load will be very high during the encoding process. To reduce the search complexity of the FS algorithm, many fast nearest neighbor codeword search algorithms have been presented. These algorithms can be grouped into three categories: spatial (or temporal) inequality based [2]–[7], pyramid structure based [8] and transform domain based [9]–[11]. The spatial (or temporal) inequality based algorithms eliminate unlikely codewords by utilizing the inequalities based on the characteristic values such as sum, mean, variance and L_2 norm of the spatial or temporal vector. The pyramid structure-based algorithms reject impossible codewords by using the inequalities layer by layer. The transform domain-based algorithms efficiently perform the elimination criteria in wavelet or Hadamard transform. In this paper, we present a novel fast codeword search algorithm based on Hadamard transform with three elimination criteria, which are very efficient in the case of high dimension.

2. Basic definitions and properties

Before describing the proposed algorithm, we give some basic definitions and properties in advance. Let H_n be the $2^n \times 2^n$ Hadamard square matrix with elements in the set $\{1, -1\}$. By assuming all of the following vectors are k-dimensional vectors and $k = 2^n$ ($n > 0$), the following basic definitions can be introduced:

`Definition 1:` $H_1 = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$ and $H_{n+1} = \begin{bmatrix} H_n & H_n \\ H_n & -H_n \end{bmatrix}$.

`Definition 2:` The Hadamard-transformed vector X of the vector x is defined as:

$$X = H_n x \tag{2}$$

And the Hadamard-transformed codeword c_i of the codeword c_i is defined as:

$$C_i = H_n c_i \quad (3)$$

Definition 3: The Hadamard-transformed standard deviation of vector X can be defined as:

$$V_X = \sqrt{\sum_{l=2}^{k} X_l^2} \quad (4)$$

And the Hadamard-transformed standard deviation of the codeword C_i can be defined as:

$$V_i = \sqrt{\sum_{l=2}^{k} C_{il}^2} \quad (5)$$

Definition 4: The Hadamard-transformed norm of vector X can be defined as:

$$\|X\| = \sqrt{\sum_{l=1}^{k} X_l^2} \quad (6)$$

And the Hadamard-transformed norm of the codeword C_i can be defined as:

$$\|C_i\| = \sqrt{\sum_{l=1}^{k} C_{il}^2} \quad (7)$$

Note that compared with Equations 4, Equation 6 takes the first element of the vector X into account. Based on above definitions, we can get the following lemmas.

Lemma 1: The distortion between two spatial vectors and the distortion between the corresponding transformed vectors have the following relationship:

$$d(X, C_i) = kd(x, c_i) \quad (8)$$

Lemma 2: The first element of X is equal to the sum of all components of x, i.e.,

$$X_1 = S_x \quad (9)$$

Where S_x denotes the sum of vector x. This equation can be derived from the fact that each element in the first row of H_n has the same value '1'.

Lemma 3: The standard deviation of the transformed vector X and the norm of the transformed vector X have the following relationship:

$$V_X = \sqrt{\|X\|^2 - X_1^2} \quad (10)$$

Based on Equations 4 and 6, we can easily obtain Equation 10. According to above definitions and lemmas, we can obtain the following three theorems:

Theorem 1:

$$|X_1 - C_{i1}| \leq \sqrt{d(X, C_i)} = \sqrt{kd(x, c_i)} \quad (11)$$

Because $(X_1 - C_{i1})^2$ is one of the summation items in $\sum_{l=1}^{k}(X_l - C_{il})^2$, so above inequality is obviously tenable.

Theorem 2:

$$|V_X - V_i| \leq \sqrt{d(X, C_i)} = \sqrt{kd(x, c_i)} \quad (12)$$

Proof: This inequality is equivalent to the following inequalities:

$$\begin{aligned}
&\Leftrightarrow \quad V_X^2 + V_i^2 - 2V_X V_i \leq \textstyle\sum_{l=1}^{k}(X_l - C_{il})^2 \\
&\Leftrightarrow \quad \textstyle\sum_{l=2}^{k} X_l^2 + \sum_{l=2}^{k} C_{il}^2 - 2V_X V_i \\
&\qquad \leq \textstyle\sum_{l=1}^{k} X_l^2 + \sum_{l=1}^{k} C_{il}^2 - \sum_{l=1}^{k} 2X_l C_{il} \\
&\Leftrightarrow \quad -2V_X V_i \leq X_1^2 + C_{i1}^2 - \textstyle\sum_{l=1}^{k} 2X_l C_{il} \\
&\Leftrightarrow \quad -2V_X V_i \leq (X_1 - C_{i1})^2 - \textstyle\sum_{l=2}^{k} 2X_l C_{il}
\end{aligned}$$

Based on the Cauchy-Schwarz inequality

$$\sqrt{\sum_{l=2}^{k} X_l^2}\sqrt{\sum_{l=2}^{k} C_{il}^2} \geq \sum_{l=2}^{k} X_l C_{il},$$

we can get

$$\begin{aligned}
2V_X V_i \geq \sum_{l=2}^{k} 2X_l C_{il} &\Rightarrow -2V_X V_i \leq -\sum_{l=2}^{k} 2X_l C_{il} \\
&\Rightarrow -2V_X V_i \leq (X_1 - C_{i1})^2 - \sum_{l=2}^{k} 2X_l C_{il}
\end{aligned}$$

This completes the proof.

Theorem 3:

$$|\|X\| - \|C_i\|| \leq \sqrt{d(X, C_i)} = \sqrt{kd(x, c_i)} \quad (13)$$

Proof: This inequality is equivalent to the following inequalities:

$$\begin{aligned}
&\Leftrightarrow \quad \|X\|^2 + \|C_i\|^2 - 2\|X\| \cdot \|C_i\| \leq \textstyle\sum_{l=1}^{k}(X_l - C_{il})^2 \\
&\Leftrightarrow \quad \textstyle\sum_{l=1}^{k} X_l^2 + \sum_{l=1}^{k} C_{il}^2 - 2\|X\| \cdot \|C_i\| \\
&\qquad \leq \textstyle\sum_{l=1}^{k} X_l^2 + \sum_{l=1}^{k} C_{il}^2 - \sum_{l=1}^{k} 2X_l C_{il} \\
&\Leftrightarrow \quad 2\|X\| \cdot \|C_i\| \geq \textstyle\sum_{l=1}^{k} 2X_l C_{il} \\
&\Leftrightarrow \quad \sqrt{\textstyle\sum_{l=1}^{k} X_l^2} \cdot \sqrt{\textstyle\sum_{l=1}^{k} C_{il}^2} \geq \textstyle\sum_{l=1}^{k} X_l C_{il}
\end{aligned}$$

The last inequality is the Cauchy-Schwarz inequality. This completes the proof.

3. Proposed algorithm

From Lemma 1, we know that the codeword that is closest to the input vector in the spatial domain is also closest to the input vector in the HT domain. Therefore we can find the corresponding best codeword in the spatial domain by searching the best codeword in the HT domain. From Definition 1, we know that the Hadamard transform based algorithms require the vector dimension to be the power of 2, i.e., $k = 2^n$. From Definition 2, we can also see that no multiplication is required for the HT.

Before describing the proposed algorithm, we first introduce the HTPDS (Hadamard Transform based on Partial Distance Search) presented in [10]. It is well known that the energy of codewords can be compacted into few elements by HT, so PDS can be efficiently used to reject unlikely codewords. Suppose each codeword c_i is with dimension $k = 2^n$. Assume the "so far" smallest transform domain distortion is D_{min}, if the first element C_{i1} of the uninspected transformed codeword C_i is larger than $MAXSUM = X_1 + \sqrt{D_{min}}$ or less than $MINSUM = X_1 - \sqrt{D_{min}}$, then C_i will not be the nearest codeword of X according to Theorem 1. Therefore, the distance calculation is necessary only for those transformed codewords whose first elements ranging from $MINSUM$ to $MAXSUM$. To perform the HTPDS algorithm, N Hadamard transformed codewords for all spatial codewords should be computed off-line and stored.

From above, we can easily see that the HTPDS algorithm only use one characteristic value, i.e., the sum of the spatial vector or the first element of the transformed vector, so HTPDS can be viewed as the equal-average (or equal-sum) nearest neighbor search algorithm in Hadamard transform domain (HTENNS). To further improve the search efficiency of HTPDS algorithm, we also consider another two characteristic values, i.e., Hadamard transformed norm and variance, in the proposed HTEEENNS (Hadamard Transform based on Equal-average Equal-variance Equal-norm) algorithm.

Based on Theorems 1, 2 and 3, assume the "so far" smallest transform domain distortion is D_{min}, three elimination criteria based on transformed vector X and codeword C_i can be stated as follows: *If*

$$C_{i1} \geq X_1 + \sqrt{D_{min}} \text{ or } C_{i1} \leq X_1 - \sqrt{D_{min}} \quad (14)$$

Then $d(X, C_i) \geq D_{min}$, and thus the codeword c_i can be eliminated. *If*

$$V_i \geq V_X + \sqrt{D_{min}} \text{ or } V_i \leq V_X - \sqrt{D_{min}} \quad (15)$$

Then $d(X, C_i) \geq D_{min}$, and thus the codeword c_i can be eliminated. *If*

$$\|C_i\| \geq \|X\| + \sqrt{D_{min}} \text{ or } \|C_i\| \leq \|X\| - \sqrt{D_{min}} \quad (16)$$

Then $d(X, C_i) \geq D_{min}$, and thus the codeword c_i can be eliminated.

With the above elimination criteria in hand, let $d^m(X, C_i) = \sum_{l=1}^{m}(X_l - C_{il})^2$ denote the partial distance between X and C_i, where $1 \leq m \leq k$, the proposed algorithm can be illustrated as follows:

Off-line steps:

1) HT is performed on all codewords c_i to obtain transformed codewords C_i.
2) The transformed codewords C_i are sorted in the ascending order of their first elements.
3) The standard deviation V_i and norm $\|C_i\|$ of each transformed codeword C_i are also computed and stored in the ordered transformed codebook.

On-line steps for each input vector x:

1) Perform HT on the input vector x to obtain X, and then compute its standard deviation V_X and norm $\|X\|$.
2) Obtain the tentative matching codeword C_p whose index is calcaluated by $p = arg\ min_i|X_1 - C_{i1}|$.
3) Compute the squared Euclidean distortion $D_{min} = d(X, C_p)$ for the initial matching codeword C_p, and then calculate the square root $SD_{min} = \sqrt{D_{min}}$. Set $S_{min} = X_1 - SD_{min}$, $S_{max} = X_1 + SD_{min}$, $V_{min} = V_X - SD_{min}$, $V_{max} = V_X + SD_{min}$, $N_{min} = \|X\| - SD_{min}$, $N_{max} = \|X\| + SD_{min}$. Set $u = 1$.
4) If $p + u > N$ (is the codebook size) or codewords from C_{p+u} to C_N have been deleted, go to step 5. Otherwise check codeword C_{p+u}. This step includes four sub-steps as follows:

Step4.1: If $C_{(p+u)1} \geq S_{max}$ or $C_{(p+u)1} \leq S_{min}$ is satisfied, then codewords from C_{p+u} to C_N can be deleted, go to step 5. Otherwise, go to step 4.2.

Step4.2: If $V_{p+u} \geq V_{max}$ or $V_{p+u} \leq V_{min}$ is satisfied, then codeword C_{p+u} can be deleted, go to step 5. Otherwise, go to step 4.3.

Step4.3: If $\|C_{p+u}\| \geq N_{max}$ or $\|C_{p+u}\| \leq N_{min}$ is satisfied, then codeword C_{p+u} can be deleted, go to step 5. Otherwise, go to step 4.4.

Step4.4: Using PDS to compute $d^m(X, C_{p+u}) = \sum_{l=1}^{m}(X_l - C_{(p+u)l})^2$ from $m = 1$ to $m = k$, if $d^m(X, C_{p+u}) \geq D_{min}$, then codeword C_{p+u} can be deleted, go to step 5. Otherwise, if $d(X, C_{p+u}) < D_{min}$, then update $D_{min} = d(X, C_{p+u})$, $SD_{min} = \sqrt{D_{min}}$, $S_{min} = X_1 - SD_{min}$, $S_{max} = X_1 + SD_{min}$, $V_{min} = V_X - SD_{min}$, $V_{max} = V_X + SD_{min}$, $N_{min} = \|X\| - SD_{min}$ and $N_{max} = \|X\| + SD_{min}$, go to step 5.

5) If $p - u < 1$ or codeword from C_1 to C_{p-u} have been deleted, go to step 6. Otherwise check codeword C_{p-u}. This step includes four sub-steps as follows:

Step5.1: If $C_{(p-u)1} \geq S_{max}$ or $C_{(p-u)1} \leq S_{min}$ is satisfied, then codewords from C_1 to C_{p-u} can be deleted, go to step 6. Otherwise, go to step 5.2.

Step5.2: If $V_{p-u} \geq V_{max}$ or $V_{p-u} \leq V_{min}$ is satisfied, then codeword C_{p-u} can be deleted, go to step 6. Otherwise, go to step 5.3.

Step5.3: If $\|C_{p-u}\| \geq N_{max}$ or $\|C_{p-u}\| \leq N_{min}$ is satisfied, then codeword C_{p-u} can be deleted, go to step 6. Otherwise, go to step 5.4.

Step5.4: Using PDS [2] to compute $d^m(X, C_{p-u}) = \sum_{l=1}^{m}(X_l - C_{(p-u)l})^2$ from $m = 1$ to $m = k$, if $d^m(X, C_{p-u}) \geq D_{min}$, then codeword C_{p-u} can be deleted, go to step 6. Otherwise, if $d(X, C_{p-u}) < D_{min}$, then update $D_{min} = d(X, C_{p-u})$, $SD_{min} = \sqrt{D_{min}}$, $S_{min} = X_1 -$

SD_{min}, $S_{max} = X_1 + SD_{min}$, $V_{min} = V_X - SD_{min}$, $V_{max} = V_X + SD_{min}$, $N_{min} = \|X\| - SD_{min}$ and $N_{max} = \|X\| + SD_{min}$, go to step 6.

6) Set $u = u + 1$. If $p + u > N$ and $p - u < 1$ or all codewords have been deleted, terminate the algorithm. Otherwise go to step 4.

4. Experimental results

We performed experiments on a Pentium-4 (2GHz) IBM-PC using two 512×512 monochrome images Lena and Baboon with 256 grey scales. Four codebooks of size 1024 and dimensions ($8 \times 8 = 64$ or $16 \times 16 = 256$) were designed using LBG algorithm [1] with the Lena image as the training set. The two images were used to test the effectiveness of the algorithms. The proposed algorithm ($HTEEENNS$) was compared to the FS (Full search), PDS (*Partial Distortion Search*) [2], ENNS(*Equal-average Nearest Neighbor Search*) [3], EENNS(*Equal-average Equal-variance Nearest Neighbor Search*) [4], EEENNS (*Equal-average Equal-variance Equal-norm Nearest Neighbor Search*) [6], SVEENNS(*Sub-vector based Equal-average Equal-variance Nearest Neighbor Search*) [7], NOS(*Norm Ordered Search*) [5], TNOS(*Transform-domain-based Norm Ordered Search*) [11] and HTPDS (*Hadamard Transform domain Partial Distortion Search*) [10] algorithms in terms of the CPU time and the arithmetic complexity (the average number of distance calculations per input vector) for different codebook sizes and vector dimensions as shown in Table I for Lena image and Table II for Baboon image. Because the Lena image is in the training set, while the Baboon image is a high-detail image outside the training set, the encoding time of Baboon image is much longer than that of the Lena image.

TABLE I

Comparisons of various fast search algorithms for Lena image in the training set.

Codebook size	1024			
Performance	CPU Time(s)		Complexity	
Dimension	8× 8	16× 16	8× 8	16× 16
FS	17.926	17.194	1024	1024
PDS [2]	2.733	2.454	99.62	112.25
ENNS [3]	0.431	0.59	24.10	34.77
EENNS [4]	0.33	0.511	17.71	29.04
EEENNS [6]	0.31	0.49	17.25	27.94
SVEENNS [7]	0.30	0.381	19.87	23.71
NOS [5]	1.262	1.032	81.10	74.18
TNOS [11]	0.441	0.39	15.92	19.04
HTPDS [10]	0.29	0.321	15.22	18.22
Proposed HTEEENNS	0.231	0.291	12.22	16.72

From Tables I and II, we can see that the proposed algorithm is superior to all other algorithms for both low-detail and high-detail images, especially in the case of high dimensionality. For Lena image encoding with the codebook of size 1024, the encoding time of proposed algorithm is only about 1.5 percent of the full search algorithm on average.

TABLE II

Comparisons of various fast search algorithms for Baboon image outside the training set

Codebook size	1024			
Performance	CPU Time(s)		Complexity	
Dimension	8× 8	16× 16	8× 8	16× 16
FS	17.916	17.685	1024.00	1024.00
PDS [2]	6.900	7.851	270.31	315.28
ENNS [3]	2.544	3.485	147.06	193.14
EENNS [4]	2.174	3.184	122.68	185.05
EEENNS [6]	2.093	3.024	117.27	176.57
SVEENNS [7]	1.462	2.103	98.07	130.33
NOS [5]	3.925	4.537	272.58	331.33
TNOS [11]	2.073	2.704	111.22	136.12
HTPDS [10]	1.692	2.133	111.65	136.17
Proposed HTEEENNS	1.422	1.963	92.63	131.40

5. Conclusions

This paper presents a fast codeword search algorithm based on three inequalities in Hadamard transform domain denoted by Theorems 1 to 3. The algorithm can dramatically reduce the complexity in the case of high-dimensional image vector quantization.

REFERENCES

[1] Y. Linde, A. Buzo and R. M. Gray, "An algorithm for vector quantizer design," IEEE Trans. on Communications, vol.COM-28, no.1, pp.84–95, 1980.

[2] C. D. Bei and R. M. Gray, "An improvement of the minimum distortion encoding algorithm for vector quantization," IEEE Trans. on Communications, vol.33, no.10, pp.1132–1133, 1985.

[3] L. Guan and M. Kamel, "Equal-average hyperplane partitioning method for vector quantization of image data," Pattern Recognition Letters, vol.13, no.10, pp.693–699, 1992.

[4] C. H. Lee and L. H. Chen, "Fast closest codeword search algorithm for vector quantization," IEE Proc.- Vision Image and Signal Processing, vol.141, no.3, pp.143–148, 1994.

[5] K. S. Wu and J. C. Lin, "Fast VQ encoding by an efficient kick-out condition," IEEE Trans. on Circuits Systems for Video Technology, vol.10, no.1, pp.59–62, 2000.

[6] Z. M. Lu and S. H. Sun, "Equal-average equal-variance equal-norm nearest neighbor search algorithm for vector quantization," IEICE Trans. Information and Systems, vol.E86-D, no.3, pp.660–663, 2003.

[7] J. S. Pan, Z. M. Lu and S. H. Sun, "An efficient encoding algorithm for vector quantization based on sub-vector technique," IEEE Trans. on Image Processing, vol.12, no.3, pp.265–270, 2003.

[8] C. H. Lee and L. H. Chen, "A fast search algorithm for vector quantization using mean pyramids of codewords," IEEE Trans. on Communications, vol.43, no.(2/3/4), pp.1697–1702, 1995.

[9] W. J. Hwang, S. S. Jeng and B. Y. Chen, "Fast codeword search algorithm using wavelet transform and partial distance search techniques," Electronics Letters, vol.33, no.5, pp.365–366, 1997.

[10] Z. M. Lu, J. S. Pan and S. H. Sun, "Efficient codeword search algorithm based on Hadamard transform," Electronics Letters, vol.36, no.16, pp.1364–1365, 2000.

[11] S. D. Jiang, Z. M. Lu and Q. Wang, "Fast norm-ordered codeword search algorithms for image vector quantization," Chinese Journal of Electronics, vol.12, no.3, pp.373–376, 2003.