Pattern Recognition Letters 21 (2000) 399–405

Pattern Recognition Letters

www.elsevier.nl/locate/patrec

Multiplication-free fast codeword search algorithm using Haar transform with squared-distance measure

Wen-Jyi Hwang [a,*], Ray-Shine Lin [a], Wen-Liang Hwang [b], Chung-Kun Wu [c]

[a] *Department of Electrical Engineering, Chung Yuan Christian University, Chung-li 32023, Taiwan, ROC*
[b] *Institute of Information Science, Academia Sinica, Taiwan, ROC*
[c] *Department of Electrical Engineering, Chung Yuan Christian University, Chung-li 32023, Taiwan, ROC*

Received 29 June 1999; received in revised form 30 December 1999; accepted 18 January 2000

Abstract

This letter presents novel multiplication-free fast codeword search algorithms for encoding of vector quantizers (VQs) based on squared-distance measure. The algorithms accomplish fast codeword search by performing the partial distance search (PDS) in the wavelet domain. To eliminate the requirement for multiplication, simple Haar wavelet is used so that the wavelet coefficients of codewords are finite precision numbers. The computation of squared distance for PDS can therefore be effectively realized using additions. To further enhance the computational efficiency of the algorithms, the addition-based squared-distance computation is decomposed into a number of stages. The PDS process is then extended to these stages to reduce the addition complexity of the algorithm. In addition, by performing PDS over smaller number of stages, lower computational complexity can be obtained at the expense of slightly higher average distortion for encoding. Simulation results show that our algorithms are very effective for the encoding of VQs, where both low computational complexity and average distortion are desired. © 2000 Elsevier Science B.V. All rights reserved.

Keywords: Vector quantization; Fast codeword search; Image compression

1. Introduction

Vector quantizers (VQs) (Abut, 1990; Gersho and Gray, 1992) have been shown to be very effective for signal compression and video coding. In the design of a VQ, in addition to its rate-distortion performance, the computational complexity for encoding is usually the important concern. For the full-search VQs with high vector dimension and/or large number of codewords, although a good rate-distortion performance can be achieved, the high computational complexity for encoding might cause the VQs impractical for the implementation of realtime processing systems. To eliminate the drawback, a number of fast codeword search algorithms (Bei and Gray, 1985; Hwang et al., 1997; Paliwal and Ramasubramanian, 1989) have been proposed for the encoding of full-search VQs. Although reasonable reduction in computational complexity is achieved,

* Corresponding author. Tel.: +886-3-456-3171-4808; fax: +886-3-456-3160.
E-mail address: whwang@dec.ee.cycu.edu.tw (W.-J. Hwang).

0167-8655/00/$ - see front matter
PII: S 0 1 6 7 - 8 6 5 5 (0 0) 0 0 0 0 7 - 6

multiplications are still required for these algorithms.

The objective of this letter therefore is to present multiplication-free fast codeword search algorithms for the encoding of full-search VQs based on squared-distance measure. Similar to the technique presented in (Hwang et al., 1997), our algorithms achieve fast encoding by performing the partial distance search (PDS) (Bei and Gray, 1985) in the wavelet domain (Vetterli and Kovacevic, 1995). The wavelet used for the fast search is the simple Haar wavelet so that the wavelet coefficients of codewords are finite-precision numbers. Consequently, squared distance calculation can be realized using additions. In our algorithms, the addition-based squared computation is decomposed into a number of stages. The PDS is also extended to these stages to further reduce the addition complexity of the algorithm. When all the stages for squared computation are considered in PDS, the actual closest codeword to each sourceword can always be found and the rate-distortion performance is not degraded. Moreover, by performing the PDS over smaller number of stages, lower computational complexity can be obtained at the expense of possible slight degradation in average distortion. Therefore, in our algorithms, in addition to be multiplication-free, the addition complexity is allowed to be controlled to provide best trade off between computational complexity and rate-distortion performance in accordance with the application needs.

2. Preliminaries

In this section we review some basic facts of wavelet transform (Vetterli and Kovacevic, 1995) and the fast codeword search algorithm performing PDS in the wavelet domain (Hwang et al., 1997). Let $\boldsymbol{X}$ be the n-stage discrete wavelet transform (DWT) of a $2^n \times 2^n$ vector $\boldsymbol{x}$. Then, as shown in Fig. 1, $\boldsymbol{X}$ is also a $2^n \times 2^n$ vector containing sub-vectors $\boldsymbol{x}_{L(-n)}$ and $\boldsymbol{x}_{Vk}, \boldsymbol{x}_{Hk}, \boldsymbol{x}_{Dk}$, $k = -n, \ldots, -1$, each with dimension $2^{(n+k)} \times 2^{(k+n)}$. Note that, in the DWT, the sub-vectors $\boldsymbol{x}_{Lk}$ (lowpass sub-vectors) and $\boldsymbol{x}_{Vk}, \boldsymbol{x}_{Hk}, \boldsymbol{x}_{Dk}$ (V, H and D orientation selective highpass sub-vectors), $k = -n, \ldots, -1$, are obtained recursively from $\boldsymbol{x}_{L(k+1)}$ with $\boldsymbol{x}_{L0} = \boldsymbol{x}$. The decomposition of $\boldsymbol{x}_{L(k+1)}$ into four sub-vectors $\boldsymbol{x}_{Lk}, \boldsymbol{x}_{Vk}, \boldsymbol{x}_{Hk}, \boldsymbol{x}_{Dk}$ can be carried out using a simple quadrature mirror filter (QMF) scheme as shown in (Vetterli and Kovacevic, 1995).

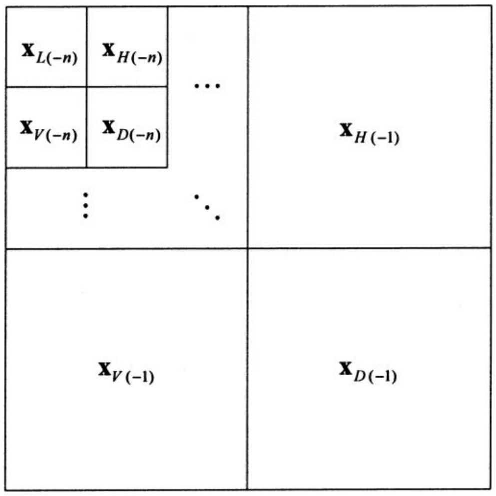

Fig. 1. The DWT of a vector $\boldsymbol{x}$.

Now, suppose there are K codewords in the codebook of a full-search VQ: $\boldsymbol{y}^1, \ldots, \boldsymbol{y}^K$, each one with dimension $2^n \times 2^n$. Let $\boldsymbol{x}$ be the sourceword with the same dimension as these codewords. The objective of the fast search algorithm for the full-search VQ is to reduce the computational time for finding a codeword whose squared distance is closest to the sourceword. That is, the algorithm reduces the computational complexities for finding $\boldsymbol{y}^{j^*}$, where $j^* = \arg\min_{1 \leqslant j \leqslant K} D(\boldsymbol{x}, \boldsymbol{y}^j)$, and $D(\boldsymbol{u}, \boldsymbol{v}) = \sum_{i=1}^{N} (u_i - v_i)^2$ is the squared distance between $\boldsymbol{u}$ and $\boldsymbol{v}$ and N is the dimension of $\boldsymbol{u}$ and $\boldsymbol{v}$.

Let $\boldsymbol{X}$ and $\boldsymbol{Y}^j$ be the n-stage DWT of $\boldsymbol{x}$ and $\boldsymbol{y}^j$, respectively. It can be shown that $D(\boldsymbol{x}, \boldsymbol{y}^j) = D(\boldsymbol{X}, \boldsymbol{Y}^j)$. Starting from the upper-left corner of the DWT coefficients, we index the elements of vectors $\boldsymbol{X}$ and $\boldsymbol{Y}^j$ in the zig-zag order as shown in Fig. 2. Let X_i and Y_i^j be the ith element of $\boldsymbol{X}$ and $\boldsymbol{Y}^j$, respectively. Moreover, let $D^m(\boldsymbol{X}, \boldsymbol{Y}^j) = \sum_{i=1}^{m} (X_i - Y_i^j)^2$, $m = 1, \ldots, 2^n \times 2^n$, be the partial distance between $\boldsymbol{X}$ and $\boldsymbol{Y}^j$. Since $D(\boldsymbol{X}, \boldsymbol{Y}^j) > D^m(\boldsymbol{X}, \boldsymbol{Y}^j)$, it follows that:

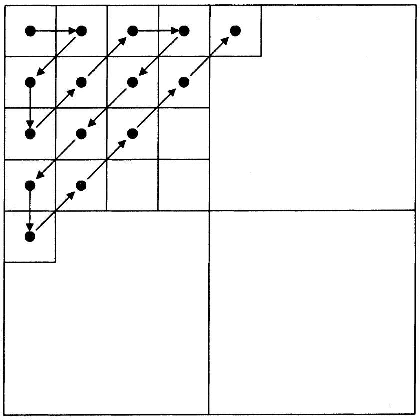

Fig. 2. Zig-zag ordering of DWT coefficients.

$$D(\boldsymbol{x}, \boldsymbol{y}^j) > D^m(\boldsymbol{X}, \boldsymbol{Y}^j). \quad (1)$$

In particular, for $m = 1$, we have $D(\boldsymbol{x}, \boldsymbol{y}^j) > (X_1 - Y_1^j)^2$. It then follows that:

$$\sqrt{D(\boldsymbol{x}, \boldsymbol{y}^j)} > |X_1 - Y_1^j|. \quad (2)$$

In the algorithm, before the search process, we obtain the DWT of the codewords. To perform the fast codeword search, first we initialize the *current closest codeword* to be $\boldsymbol{y}^p$, where $p = \arg\min_j D^1(\boldsymbol{X}, \boldsymbol{Y}^j)$ and *the current minimum distortion* $D_{\min}$ to be $D(\boldsymbol{x}, \boldsymbol{y}^p)$. For each codeword $\boldsymbol{y}^j$ to be searched and we compute $|Y_1^j - X_1|$. Suppose $|Y_1^j - X_1| > \sqrt{D_{\min}}$, it follows from Eq. (2) that $\sqrt{D(\boldsymbol{x}, \boldsymbol{y}^j)} > \sqrt{D_{\min}}$. Hence, $\boldsymbol{y}^j$ is not the closest codeword to $\boldsymbol{x}$ and can be rejected. Suppose $|Y_1^j - X_1| < \sqrt{D_{\min}}$, then we perform the following PDS process. Starting from $m = 2$, for each value of m, $m = 2, \ldots, 2^n \times 2^n$, we first evaluate $D^m(\boldsymbol{X}, \boldsymbol{Y}^j)$. Suppose $D^m(\boldsymbol{X}, \boldsymbol{Y}^j) > D_{\min}$, then from Eq. (1), it follows that $D(\boldsymbol{x}, \boldsymbol{y}^j) > D_{\min}$ and $\boldsymbol{y}^j$ can be rejected. Otherwise, we go to the next value of m and repeat the same process. This PDS process is continued until $\boldsymbol{y}^j$ is rejected or m reaches $2^n \times 2^n$. If $m = 2^n \times 2^n$, then we compare $D(\boldsymbol{X}, \boldsymbol{Y}^j)$ with $D_{\min}$. If $D(\boldsymbol{X}, \boldsymbol{Y}^j) < D_{\min}$, then the *current minimum distortion* $D_{\min}$ is replaced by $D(\boldsymbol{X}, \boldsymbol{Y}^j)$ and the *current closest codeword* to $\boldsymbol{x}$ is set to $\boldsymbol{y}^j$. After all the codewords are searched, the final *current closest codeword* is the actual closest codeword to $\boldsymbol{x}$ and the $D_{\min}$ is the corresponding distance.

3. Multiplication-free partial distance search in the transform domain

Although performing PDS in the wavelet domain can significantly reduce the computational complexity without degrading rate-distortion performance of VQs, multiplications are still required for the algorithm. The objective of this letter therefore is to present multiplication-free fast search techniques which performs PDS in the transform domain. In our algorithm, the Haar wavelet is used to obtain wavelet coefficients for PDS. This is because the coefficients in Haar domain are finite-precision numbers so that the squared distance calculation for PDS can be realized without multiplications. In addition, because of the simplicity of the Haar wavelet, even the fixed-point DSP processors can be effectively used for implementing the transform without large computational overhead.

Assume the elements in a vector $\boldsymbol{x}$ are q-bit integers. That is, if x is an element of $\boldsymbol{x}$, then x can be expressed as

$$x = \mathscr{X}_{q-1}2^{q-1} + \mathscr{X}_{q-2}2^{q-2} + \cdots + \mathscr{X}_0 2^0, \quad (3)$$

where $\mathscr{X}_i$, $q-1 \leqslant i \leqslant 0$, are binary numbers taking only values of 0 or 1. Let X be an element of $\boldsymbol{X}$ in the Haar domain. Based on Eq. (3), if X is located in the subbands at intermediate resolution level k, (that is, $X \in \{\boldsymbol{x}_{Hk}, \boldsymbol{x}_{Vk}, \boldsymbol{x}_{Dk}\}$, where $-(n-1) \leqslant k \leqslant -1$) then X can be expressed as

$$X = \mathscr{X}_{q-k-1}2^{q-k-1} + \mathscr{X}_{q-k-2}2^{q-k-2} + \cdots + \mathscr{X}_k 2^k, \quad (4)$$

where $\mathscr{X}_i$, $k \leqslant i \leqslant q-k-1$, are also binary numbers taking only values of 0 or 1. Otherwise, if X is located in the lowest resolution level, (that is, $X \in \{\boldsymbol{x}_{L(-n)}, \boldsymbol{x}_{H(-n)}, \boldsymbol{x}_{V(-n)}, \boldsymbol{x}_{D(-n)}\}$) then the number of bits required for storing X becomes largest and X can be expressed as:

$$X = \mathscr{X}_{q+n-1}2^{q+n-1} + \mathscr{X}_{q+n-2}2^{q+n-2} + \cdots + \mathscr{X}_{-n}2^{-n}. \quad (5)$$

Given two vectors $\boldsymbol{x}$ and $\boldsymbol{y}$, define $A_m = |X_m - Y_m|$, where the indexing of DWT coefficients are given in Fig. 2. Suppose X_m and Y_m are inside subbands at resolution level k, then A_m can be written as

$$A_m = \mathscr{A}_{q-k-1}2^{q-k-1} + \mathscr{A}_{q-k-2}2^{q-k-2} + \cdots + \mathscr{A}_k 2^k, \tag{6}$$

where $\mathscr{A}_i$, $k \leqslant i \leqslant q-k-1$ are binary numbers taking only values of 0 or 1. When performing the PDS, the computation of A_m^2 is necessary. To eliminate the requirement for multiplication when computing A_m^2, in this letter we propose two techniques: the usual partial product accumulation (PPA) (Katz, 1995) and modified PPA.

The PPA technique is based on the fact that

$$A_m^2 = A_m(\mathscr{A}_{q-k-1}2^{q-k-1} + \cdots + \mathscr{A}_k 2^k), \tag{7}$$

$$A_m^2 = \mathscr{B} + \sum_{r=k}^{q-k-2} \mathscr{B}_r, \tag{8}$$

where

$$\mathscr{B} = A_m \mathscr{A}_{q-k-1} 2^{q-k-1} \tag{9}$$

and

$$\mathscr{B}_r = A_m \mathscr{A}_r 2^r. \tag{10}$$

Define $\mathscr{S}_1 = \{\mathscr{B}, \mathscr{B}_r, k \leqslant r \leqslant q-k-2\}$. Since the computation of elements in $\mathscr{S}_1$ require only shift operations, multiplication is not necessary when PPA is used for computing A_m^2. In addition, from Eq. (8), it follows that A_m^2 can be decomposed into $q-2k$ terms in which each term involves only the computation of distinct single element in $\mathscr{S}_1$. Therefore, the PDS can be extended over these terms to further reduce the computational complexity. From Eqs. (9) and (10), it is observed that $\mathscr{B}$ and $\mathscr{B}_r$ having larger r value in $\mathscr{S}_1$ have higher energy than the other elements. Consequently, these elements are scanned first for PDS. It is known that higher discrepancy in energy among elements for PDS can result in higher efficiency for fast codeword search. Nevertheless, all the elements in $\mathscr{S}_1$ are simply the shifted version of A_m, and therefore the difference in energy among them might not be large. To enhance the discrepancy in energy, the modified PPA is proposed for PDS.

In the modified PPA, we first note that, from Eq. (6), $A_m^2 = (X_m - Y_m)^2$ can be expressed as

$$A_m^2 = \sum_{s=k}^{q-k-1} \mathscr{A}_s^2 2^{2s} + 2\sum_{r=k+1}^{q-k-1}\sum_{s=k}^{r-1} \mathscr{A}_r \mathscr{A}_s 2^r 2^s. \tag{11}$$

Let

$$\mathscr{C} = \sum_{s=k}^{q-k-1} \mathscr{A}_s^2 2^{2s} \tag{12}$$

and

$$\mathscr{C}_r = \sum_{s=k}^{r} \mathscr{A}_{r+1} \mathscr{A}_s 2^{r+s+2}, \tag{13}$$

where $k \leqslant r \leqslant (q-k-2)$. We therefore can rewrite A_m^2 as

$$A_m^2 = \mathscr{C} + \sum_{r=k}^{q-k-2} \mathscr{C}_r. \tag{14}$$

Let $\mathscr{S}_2 = \{\mathscr{C}, \mathscr{C}_r, k \leqslant r \leqslant q-k-2\}$ be the set of elements for PDS for modified PPA. Observe that, $\mathscr{C}_k = \mathscr{A}_{k+1}\mathscr{A}_k 2^{2k+2}$, which is the element that contain least energy among the elements in $\mathscr{S}_2$, in general has much less energy than $\mathscr{B}_k = \mathscr{A}_k A_m 2^k = \mathscr{A}_k(\mathscr{A}_{q-k-1}2^{q-1} + \cdots + \mathscr{A}_k 2^{2k})$, which is the element that contain least energy among the elements in $\mathscr{S}_1$. In addition, the difference in energy between $\mathscr{C}$ and $\mathscr{B}$, the elements that contain highest energy in $\mathscr{S}_2$ and $\mathscr{S}_1$, respectively, is relatively small. Therefore, $\mathscr{S}_2$ has higher discrepancy in energy among its elements and can be quite effective for PDS.

In the following, we present the PDS technique based on PPA and modified PPA in more detail. Suppose $\boldsymbol{y}^j$ is the *current* codeword to be searched and it has been found that $D^{(m-1)}(\boldsymbol{X}, \boldsymbol{Y}^j)$ is less than the *current* $D_{\min}$. In the original PDS in the transform domain, $D^m(\boldsymbol{X}, \boldsymbol{Y}^j) = D^{(m-1)}(\boldsymbol{X}, \boldsymbol{Y}^j) + (X_m - Y_m^j)^2$ is computed and compared to *current* $D_{\min}$. This operation requires one multiplication. To eliminate the need for multiplications, we first let $A_m^2 = (X_m - Y_m^j)^2$ in the novel PDS algorithm. Then, the PDS is extended over the computation of A_m^2. That is, $D^{(m-1)}(\boldsymbol{X}, \boldsymbol{Y}^j) + \mathscr{D}$ is first compared with $D_{\min}$, where $\mathscr{D} = \mathscr{B}$ when PPA is employed and $\mathscr{D} = \mathscr{C}$ when modified PPA is employed. If $D^{(m-1)}(\boldsymbol{X}, \boldsymbol{Y}^j) + \mathscr{D} > D_{\min}$, it follows that $D(\boldsymbol{x}, \boldsymbol{y}^j) > D_{\min}$ and $\boldsymbol{y}^j$ can be rejected. Otherwise, we start the following fast search process which scans the elements having higher energy first (that is, the $\mathscr{B}_r$ or $\mathscr{C}_r$ having higher index r value). For simplicity, depending on whether PPA or modified PPA is used, we assume $\mathscr{D}_r = \mathscr{B}_r$ (for PPA), or $\mathscr{D}_r = \mathscr{C}_r$ (for modified PPA). Starting from $i = q - k - 2$, for each value of i, $i = q - k - 2, \ldots, k$, we compare $D^{(m-1)}(\boldsymbol{X}, \boldsymbol{Y}^j) + \mathscr{D} + \sum_{r=q-k-2}^{i} D_r$ with $D_{\min}$. If $D^{(m-1)}(\boldsymbol{X}, \boldsymbol{Y}^j) + \mathscr{D} + \sum_{r=q-k-2}^{i} \mathscr{D}_r > D_{\min}$, then we reject $\boldsymbol{y}^j$. Otherwise, we go to next value of i and repeat the same process. This process is continued until $\boldsymbol{y}^j$ is rejected, or i reaches k. If $i = k$ and $D^{(m-1)}(\boldsymbol{X}, \boldsymbol{Y}^j) + \mathscr{D} + \sum_{r=q-k-2}^{k} \mathscr{D}_r < D_{\min}$, then $D^m(\boldsymbol{X}, \boldsymbol{Y}^j) < D_{\min}$. In this case. we increase the value of m by one and the same PDS process for the computation of A_m^2 is executed. This process is continued until $\boldsymbol{y}^j$ is rejected or $D(\boldsymbol{X}, \boldsymbol{Y}^j) = D^{2^n \times 2^n}(\boldsymbol{X}, \boldsymbol{Y}^j) < D_{\min}$ is observed. In the latter case, we replace the current $D_{\min}$ by $D(\boldsymbol{X}, \boldsymbol{Y}^j)$. After all the codewords are searched, the actual closest codeword to $\boldsymbol{x}$ are then identified.

The novel PDS technique can always find the closest codeword to each sourceword and therefore the average distortion for the encoding is not increased. Nevertheless, the multiplication complexity is eliminated at the expense of increase in addition complexity. To further reduce the addition complexity, we can select an integer threshold T, where $k - 1 \leqslant T \leqslant q - k - 2$. During the PDS process, for each m value, all the $\mathscr{D}_r$s having $r \leqslant T$ are ignored (note that $\mathscr{D}$ is always included for PDS). Therefore, for a larger T value, addition complexity can be significantly reduced at the expense of slight increase in average distortion since the actual closest codeword to each sourceword might not be identified by ignoring some $\mathscr{D}_r$s for PDS. For simplicity, we define $A_m^2(T) = \mathscr{D} + \sum_{r=T+1}^{q-k-2} \mathscr{D}_r$ and $D_T^m(\boldsymbol{X}, \boldsymbol{Y}^j) = \sum_{i=1}^{m} A_m^2(T)$. In addition, $D_T(\boldsymbol{X}, \boldsymbol{Y}^p) = D_T^{2^n \times 2^n}(\boldsymbol{X}, \boldsymbol{Y}^p)$.

A detailed outline for our algorithm is listed below:

Initialization
  Given:
    DWT of codewords: $\{\boldsymbol{Y}_1, \boldsymbol{Y}_2, \ldots, \boldsymbol{Y}_K\}$,
    Number of bits for each element in vectors in the original domain: q,
    Threshold for PDS: T.
Step 1.
  For any input source vector $\boldsymbol{x}$, find $\boldsymbol{X}$.
  Find initial *current* closest codeword $\boldsymbol{y}^p$, where
    $p = \arg\min_j D_T^1(\boldsymbol{X}, \boldsymbol{Y}^j)$.
  Set initial *current* $D_{\min} = D_T(\boldsymbol{X}, \boldsymbol{Y}^p)$.
Step 2.
  For each codeword $\boldsymbol{y}^j$ to be searched:
    Set $D_T^0(\boldsymbol{X}, \boldsymbol{Y}^j) = 0$.
    For $m = 1$ to $2^n \times 2^n$ do
      Find the resolution level k of X_m and Y_m^j.
      Compute $A_m = |X_m - Y_m^j|$.
      Obtain $\mathscr{D}$ from A_m.
      Set $\mathscr{S} = D_T^{m-1}(\boldsymbol{X}, \boldsymbol{Y}^j) + \mathscr{D}$
      If $\mathscr{S} > D_{\min}$, then
        Reject $\boldsymbol{y}^j$,
        Goto Step 3.
      Endif
      For $r = q - k - 2$ *downto* $T + 1$
        Compute $\mathscr{D}_r$ from A_m
        $\mathscr{S} \leftarrow \mathscr{S} + \mathscr{D}_r$
        If $\mathscr{S} > D_{\min}$, then
          Reject $\boldsymbol{y}^j$,
          Goto Step 3.
          Endif
      Next r
      $D_T^m(\boldsymbol{X}, \boldsymbol{Y}^j) = \mathscr{S}$.
    Next m
    $D_{\min} \leftarrow D_T(\boldsymbol{X}, \boldsymbol{Y}^j)$
      current closest codeword $\leftarrow \boldsymbol{y}^j$.
Step 3.
  If all the codewords are searched, then stop,
  Otherwise, goto Step 2.

4. Simulation results

In this section, we present some simulation results to demonstrate the effectiveness of the novel PDS technique. The VQ used for encoding is designed using the generalized Lloyd algorithm (Linde et al., 1980). The number of codewords K in the VQ is 1024 and the dimension of vectors is $2^3 \times 2^3$. The training images for the VQ design are three 512 × 512 images "Lena", "Pepper" and "Girl". All the elements of codewords after VQ design are quantized with an 8-bit scalar quantizer (i.e., $q = 8$).

Fig. 3 shows the addition complexity vs. PSNR for the novel PDS technique based on both PPA and modified PPA for various T value. The PSNR is defined as PSNR = 10 log 255^2/(MSE of the reconstructed images). The addition complexity C_a is defined as $C_a = \bar{Z}_a/(2N-1)$, where $\bar{Z}_a$ is the average number of additions per sourceword and $N = 2^n \times 2^n$ is the vector dimension. The addition complexity is measured on the images Lena, Pepper and Girl. From Fig. 3, it is observed that the novel multiplication-free PDS technique based on modified PPA enjoys lower addition complexity than the PDS technique based on PPA. This is because the discrepancy in energy among the elements for PDS based on modified PPA is higher than that for PDS based on PPA. In addition, we find that the degradation of the PSNRs for both PPA and modified PPA is quite small even for highest T value. In particular, when $T = 9$, all the $\mathscr{D}_r$s are ignored for the PDS search and therefore the maximum degradation in PSNR is achieved. In this case, the PSNR for modified PPA is only degraded by 0.17 dB (from 27.30 to 27.13 dB); whereas, the reduction in addition complexity is 67.91% (from 54.60 to 17.52). These results demonstrated the effectiveness of our algorithm.

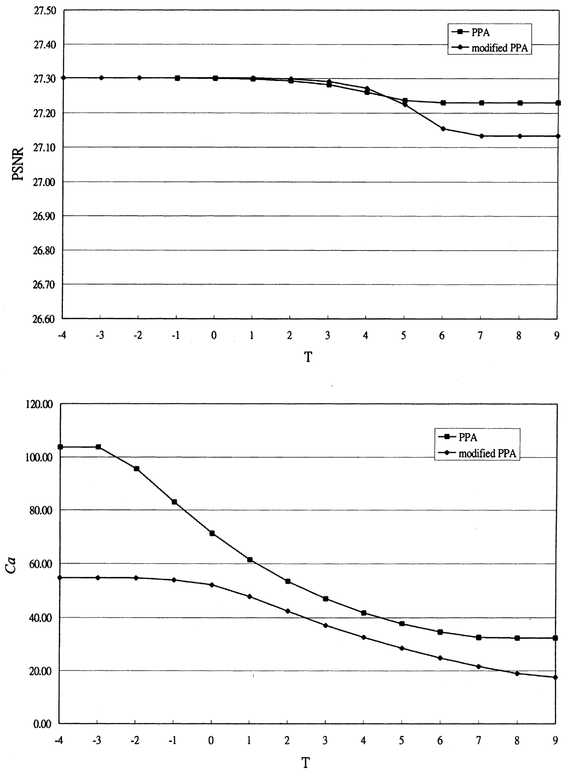

Fig. 3. The PSNR and additional complexity for the novel PDS techniques based on PPA and modified PPA for various T values.

For comparison purpose we also implement the exhaustive search, the PDS in the original domain (Bei and Gray, 1985) and the PDS in the wavelet domain without eliminating multiplication (Hwang et al., 1997) for VQ encoding. Table 1 shows the addition complexities, multiplication complexities, time complexities and the PSNRs of these algorithms. The multiplication complexity C_m is defined as $C_m = \bar{Z}_m/N$, where $\bar{Z}_m$ is the average number of multiplications per sourceword. The time complexity C_t of each codeword search algorithm is defined as the total CPU time required for arithmetic operations of that algorithm for the images to be encoded. The PC with Pentium III 450 CPU is used for CPU time measurement. All the complexities listed in the table are measured on the images Lena, Pepper and Girl.

Table 1
Addition complexities, multiplication complexities, time complexities and PSNRs of various existing codeword search algorithms

	C_a	C_m	C_t	PSNR
Exhaustive search	1024	1024	50.49	27.30
PDS (Bei and Gray, 1985)	89.87	85.32	4.55	27.30
PDS + wavelet (Hwang et al., 1997)	24.82	16.28	0.91	27.30
PDS + PPA + wavelet ($T = -4$)	103.69	0	1.09	27.30
PDS + PPA + wavelet ($T = 9$)	32.27	0	0.34	27.23
PDS + modified PPA + wavelet ($T = -4$)	54.60	0	0.57	27.30
PDS + modified PPA + wavelet ($T = 6$)	24.81	0	0.27	27.16
PDS + modified PPA + wavelet ($T = 9$)	17.52	0	0.19	27.13

From Fig. 3 and Table 1, it is observed that when $T \geqslant 6$, the addition complexities of the novel PDS technique based on modified PPA are less than those of all the other search techniques listed in Table 1. In addition, the time complexities of the novel PDS algorithm for all T values are lower than those of all the other methods. Based on the simulation results shown above, the novel PDS technique therefore can be an effective alternative for the applications where both real-time processing and high PSNR performance are desired.

5. Concluding remarks

We have shown that the PDS technique based on modified PPA in the wavelet domain is very effective for multiplication-free fast codeword search. While eliminating the requirements for multiplication, the algorithm also significantly reduces the addition complexity without increasing the average distortion for VQ encoding. In addition, the algorithm can accelerate the encoding process further by ignoring some insignificant stages for PDS at the expense of a slight increase in average distortion. Therefore, the algorithm can be efficiently used for the encoding of VQs having high vector dimension and/or large number of codewords.

References

Abut, H., 1990. Vector Quantization. IEEE Press, New York.

Bei, C.D., Gray, R.M., 1985. An improvement of the minimum distortion encoding algorithm for vector quantization. IEEE Trans. Commun. 33, 1132–1133.

Gersho, A., Gray, R.M., 1992. Vector Quantization and Signal Compression. Kluwer Academic Publishers, Dordrecht.

Hwang, W.J., Jeng, S.S., Chen, B.Y., 1997. Fast codeword search algorithm using wavelet transform and partial distance search techniques. Electron. Lett. 33, 365–366.

Katz, R.H., 1995. Contemporary Logic Design. Benjamin/Cummings, Menlo Park, CA.

Linde, Y., Buzo, A., Gray, R., 1980. An algorithm for vector quantizer design. IEEE Trans. Commun. 28, 84–95.

Paliwal, K.K., Ramasubramanian, V., 1989. Effect of ordering the codebook on the efficiency of the partial distance search algorithm for vector quantization. IEEE Trans. Commun. 37, 538–540.

Vetterli, M., Kovacevic, J., 1995. Wavelets and Subband Coding. Prentice-Hall, Eaglewood Cliffs, NJ.