A Simple Improved Full Search for Vector Quantization Based on Winograd's Identity

Kuo-Liang Chung, Wen-Ming Yan, and Jung-Gen Wu

***Abstract*—Vector quantization (VQ) technique is a well known method in image compression. Employing Winograd's identity, this letter presents a simple improved method in order to cut the computation time in the full search method for VQ nearly 50%. Some experiments are carried out to confirm the theoretical analysis.**

***Index Terms*—Full search, image compression, vector quantization, Winograd's identity.**

I. Introduction

Vector quantization (VQ) [2] has a long history in lossy image compression. In VQ, the sender encodes an image using an encoder, and the receiver decodes the compressed image using a decoder. For both the sender and receiver, they have the same codebook, which can be constructed using many algorithms, for example, the Linde, Buzo, and Gray (LBG) algorithm [4]. The image is first partitioned into many blocks. Suppose each block is a 4 × 4 subimage viewed as a sixteen-dimensional (16-D) vector, then a 512 × 512 image can be decomposed into 128 × 128 blocks. Given a block, i.e., input vector, it is an important research topic to design a search algorithm for finding the closest codeword in the codebook.

Let the input vector be k-dimensional and be denoted by $\mathbf{x} = (x_1, x_2, \cdots, x_k)$. Let the size of the codebook be $N \times k$, i.e., there are N codewords, $\mathbf{y}_1, \mathbf{y}_2, \cdots, \mathbf{y}_N$, each codeword being k-dimensional. The ith codeword $\mathbf{y}_i$ for $1 \leq i \leq N$ is denoted by $(y_{i1}, y_{i2}, \cdots, y_{ik})^t$. Given an input vector $\mathbf{x}$, finding the closest codeword in the codebook is equal to finding $\mathbf{y}_p$ such that the squared Euclidean distance is minimal, i.e., $d^2(\mathbf{x}, \mathbf{y}_p) = \min_i d^2(\mathbf{x}, \mathbf{y}_i) = \min_i \sum_{j=1}^{k} (x_j - y_{ij})^2$. In the full search method (FS for short), we examine all the codewords in the codebook. Commonly, this distance is also called the distortion between the input block and the resulting reproduction of the block and is used to measure the quality of the compression. Although VQ is an efficient method for low bit rate image compression, it is time consuming for high-dimensional vectors in the search phase. Many researchers [1], [2], [5], [6], [3] have developed different kinds of efficient search algorithms. However, without any preprocessing overhead, FS has advantages such as simplicity and global optimal solution.

Employing Winograd's identity [7], this letter presents a simple improved FS (IFS for short) in order to cut the computation time in FS for VQ nearly 50%. Some experiments are carried out to confirm the theoretical analysis. In fact, plugging other data structures such as a k-dimensional tree [5] and the other coordinate formulation such as spherical distance coordinate formulation [6] into the proposed IFS can improve the search time more, but the detailed discussion is beyond the scope of this paper.

Manuscript received July 17, 2000. This work was supported in part by Grants NSC89-2213-E011-061, NSC87-2119-M002-006, and NSC89-2614-H-003-001-F20, National Science Council, Taiwan, R.O.C. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. R. L. de Queiroz.

K.-L. Chung is with the Department of Information Management, Institute of Computer Science and Information Engineering, National Taiwan University of Science and Technology, Taipei, Taiwan 10672, R.O.C. (e-mail: klchung@cs.ntust.edu.tw).

W.-M. Yan is with the Department of Computer Science and Information Engineering, National Taiwan University, Taipei, Taiwan, R.O.C. (e-mail: ganboon@csie.ntu.edu.tw).

J.-G. Wu is with the Department of Information and Computer Education, National Taiwan Normal University, Taipei, Taiwan 10764, R.O.C. (e-mail: jgwu@ice.ntnu.edu.tw).

Publisher Item Identifier S 1070-9908(00)10181-6.

II. The Proposed IFS

Let S and T be two given subsets of R^k, where S and T can be viewed as the set of input vectors and the codebook, respectively. In FS, for each input vector $\mathbf{x} = (x_1, x_2, \cdots, x_k)^t \in S$, we want to find a $\hat{\mathbf{y}} \in T$ such that

$$d^2(\mathbf{x}, \hat{\mathbf{y}}) = \min\{d^2(\mathbf{x}, \mathbf{y}): \mathbf{y} \in T\}.$$

Given a codeword

$$\mathbf{y} = (y_1, y_2, \cdots, y_k)^t \in T$$

the squared Euclidean distance between the input vector $\mathbf{x}$ and the codeword $\mathbf{y}$ is written as

$$\begin{aligned} d^2(\mathbf{x}, \mathbf{y}) &= \sum_{j=1}^{k} (x_j - y_j)^2 \\ &= \sum_{j=1}^{k} (x_j^2 - 2x_j y_j + y_j^2) \\ &= \sum_{j=1}^{k} x_j^2 - 2\sum_{j=1}^{k} x_j y_j + \sum_{j=1}^{k} y_j^2. \end{aligned} \tag{1}$$

Employing Winograd's identity, the second summation term at the right side of (1) can be written as

$$\begin{aligned} \sum_{j=1}^{k} x_j y_j &= \sum_{j=1}^{k/2} (x_{2j-1} y_{2j-1} + x_{2j} y_{2j}) \\ &= \sum_{j=1}^{k/2} [(x_{2j-1} + y_{2j})(x_{2j} + y_{2j-1}) \end{aligned}$$

$$- x_{2j-1}x_{2j} - y_{2j-1}y_{2j}]$$
$$= \sum_{j=1}^{k/2} (x_{2j-1} + y_{2j})(x_{2j} + y_{2j-1}) - \sum_{j=1}^{k/2} x_{2j-1}x_{2j} - \sum_{j=1}^{k/2} y_{2j-1}y_{2j}.$$

Hence, the squared Euclidean distance in (1) becomes

$$\begin{aligned} d^2(\mathbf{x}, \mathbf{y}) &= \sum_{j=1}^{k} x_j^2 - 2 \cdot \left[\sum_{j=1}^{k/2} (x_{2j-1} + y_{2j})(x_{2j} + y_{2j-1}) - \sum_{j=1}^{k/2} x_{2j-1}x_{2j} - \sum_{j=1}^{k/2} y_{2j-1}y_{2j}\right] + \sum_{j=1}^{k} y_j^2 \\ &= \left[\sum_{j=1}^{k} x_j^2 + 2\sum_{j=1}^{k/2} x_{2j-1}x_{2j}\right] + \left[\sum_{j=1}^{k} y_j^2 + 2\sum_{j=1}^{k/2} y_{2j-1}y_{2j}\right] \\ &\quad - 2\left[\sum_{j=1}^{k/2} (x_{2j-1} + y_{2j})(x_{2j} + y_{2j-1})\right]. \end{aligned}$$

For any $\mathbf{z} = (z_1, z_2, \cdots, z_k)^t \in R^k$ and $\mathbf{w} = (w_1, w_2, \cdots, w_k)^t \in R^k$, we define

$$f(\mathbf{z}) = 2\sum_{j=1}^{k/2} z_{2j-1}z_{2j} + \mathbf{z}^t\mathbf{z} = \sum_{j=1}^{k/2} (z_{2j-1} + z_{2j})^2$$

and

$$g(\mathbf{z}, \mathbf{w}) = \sum_{j=1}^{k/2} (z_{2j-1} + w_{2j})(z_{2j} + w_{2j-1})$$

and we then have

$$d^2(\mathbf{x}, \mathbf{y}) = f(\mathbf{x}) + f(\mathbf{y}) - 2g(\mathbf{x}, \mathbf{y}).$$

If $\hat{\mathbf{y}} \in T$ satisfies

$$f(\hat{\mathbf{y}}) - 2g(\mathbf{x}, \hat{\mathbf{y}}) = \min\{f(\mathbf{y}) - 2g(\mathbf{x}, \mathbf{y}) \colon \mathbf{y} \in T\}$$

then we have

$$\begin{aligned} d^2(\mathbf{x}, \hat{\mathbf{y}}) &= f(\mathbf{x}) + f(\hat{\mathbf{y}}) - 2g(\mathbf{x}, \hat{\mathbf{y}}) \\ &= f(\mathbf{x}) + \min\{f(\mathbf{y}) - 2g(\mathbf{x}, \mathbf{y}) \colon \mathbf{y} \in T\} \\ &= \min\{f(\mathbf{x}) + f(\mathbf{y}) - 2g(\mathbf{x}, \mathbf{y}) \colon \mathbf{y} \in T\} \\ &= \min\{d^2(\mathbf{x}, \mathbf{y}) \colon \mathbf{y} \in T\}. \end{aligned} \qquad (2)$$

TABLE I
ENCODING TIME FOR THE 256 × 16 CODEBOOK

	FS	IFS	%
Lena	18.62	10.00	53.7
Baboon	18.57	10.05	54.1
F16	18.24	10.00	54.8
Pepper	18.51	10.00	54.0

From (2), since $f(\mathbf{x})$ will not affect the search result, the concerning computation will be dominated by $\min\{f(\mathbf{y}) - 2g(\mathbf{x}, \mathbf{y}) \colon \mathbf{y} \in T\}$, which constitutes the main body of the proposed IFS. First, the values of all the $f(\mathbf{y})$'s can be calculated in the preprocessing step easily. Once all the $f(\mathbf{y})$ values have been calculated, they can be used repeatedly. The formal algorithm for implementing the proposed IFS is listed as follows.

```
for $i \leftarrow 1$ to $n$
  $a_i \leftarrow f(\mathbf{y}_i)$
end for
for $j \leftarrow 1$ to $m$
  min $\leftarrow \infty$
  for $i \leftarrow 1$ to $n$
    $t \leftarrow a_i - 2 * g(\mathbf{x}_j, \mathbf{y}_i)$
    if min $> t$ then
      min $\leftarrow t$
      $pj \leftarrow i$
    end if
  end for
  /*** Now $\mathbf{y}_{pj}$ satisfy $d^2(\mathbf{x}_j, \mathbf{y}_{pj}) = \min\{d^2(\mathbf{x}_j, \mathbf{y}) \colon \mathbf{y} \in T\}$ ***/
  encode $\mathbf{x}_j$ by the index $pj$
end for
```

Using straightforward computation for calculating (1), it takes k subtractions, $k-$ 1 additions, and k multiplications for the input vector and one codeword $\mathbf{y}$. Since there are N codewords to be examined, in total, Nk subtractions, $N(k-1)$ additions, and Nk multiplications are needed for computing (1). In the proposed IFS, we first calculate the values of $f(\mathbf{y})$'s for all $\mathbf{y}$'s. Since the size of the codebook is small and can be used repeatedly, this preprocessing time is negligible. Afterward, for each $\mathbf{x} \in S$ and $\mathbf{y} \in T$, calculating the value of $f(\mathbf{y}) - 2 * g(\mathbf{x}, \mathbf{y})$ needs only one subtraction, $k + (k/2)$ additions, and $k/2$ multiplications. In total, N subtractions, $N(k + (k/2))$ additions, and $Nk/2$ multiplications are needed for each input vector $\mathbf{x} \in S$. For any modern computer, the time required for a multiplication is much larger than that for an addition (or a subtraction). Therefore, the proposed improved method cuts the computation time to nearly half.

III. EXPERIMENTAL RESULTS

Four popular 512 × 512 images are used as the benchmark to evaluate the performance of the proposed method. The four adopted images are Lena, baboon, F16, and pepper. The machine used is the IBM compatible personal computer with 350 MHz Pentium III microprocessor.

The execution time (in seconds) needed in our IFS and the FS is listed in Table I. Here, the codebook has 256 codewords, and the block size is 4×4, i.e., the size of the codebook is 256×16. The codebook is generated from the Lena image by using the LBG algorithm [4]. The experimental results show that the proposed method cuts computation time nearly 50%, as expected.

REFERENCES

[1] C. D. Bei and R. M. Gray, "An improvement of the minimum distortion encoding algorithm for vector quantization," *IEEE Trans. Commun.*, vol. COM-33, pp. 1132–1133, Oct. 1985.

[2] A. Gersho and R. M. Gray, *Vector Quantization and Signal Compression*. Boston, MA: Kluwer, 1992.

[3] R. M. Gray and D. L. Neuhoff, "Quantization," *IEEE Trans. Inform. Theory*, vol. 44, pp. 2325–2383, Nov. 1998.

[4] Y. Linde, A. Buzo, and R. M. Gray, "An algorithm for vector quantizer design," *IEEE Trans. Commun.*, vol. COM-28, pp. 84–95, Jan. 1980.

[5] V. Ramasubramanian and K. K. Paliwal, "Fast k-dimensional tree algorithms for nearest neighbor search with application to vector quantization encoding," *IEEE Trans. Signal Processing*, vol. 40, pp. 518–531, Mar. 1992.

[6] ——, "An efficient approximation elimination algorithm for fast nearest-neighbor search based on a spherical distance coordinate formulation," *Pattern Recognit. Lett.*, vol. 13, pp. 471–480, 1992.

[7] S. Winograd, "A new algorithm for inner product," *IEEE Trans. Comput.*, vol. 17, pp. 693–694, July 1968.