JOURNAL OF INFORMATION SCIENCE AND ENGINEERING 16, 661-668 (2000)

Short Paper

Optimal Feature-based Vector Quantization of Image Coding Using Integral Projections

YIH-CHUAN LIN
Department of Computer Science and Information Engineering
Shu-Te Institute of Technology
Kaohsiung County, Taiwan 824, R.O.C.
E-mail: yclin@mail.stit.edu.tw

This paper presents a fast algorithm of vector quantization for image coding that relies on three conditions of early termination to confine the search space, resulting in acceleration of the encoding process. These termination conditions are derived based on the gray-level features that are extracted from the individual vectors of pixels. Each incoming vector is compared to the codebook entries first using these fast tests. The codebook entries that fail one or more of the fast tests can be rejected without further consideration. Thus, time-consuming computations of the squared Euclidean distance in the proposed system are performed on only a few codebook entries that first pass all three fast tests. Verified results for the system show over 97% reduction of execution time compared to the full search algorithm and 50% compared to the mean-ordered partial codebook search method in image VQ encoding.

Keywords: vector quantization, codebook search, image coding, integral projections, fast algorithms, feature extraction

1. INTRODUCTION

Signal compression using vector quantization (VQ) has received much attention because of its promising compression ratio and relatively simple structure. Unlike scalar quantization, VQ requires segmenting of the source signal sequence into a group of individual vectors. For each input vector, the VQ encoder searches through a predesigned codebook, in which several representative vectors are included, for the entry that is closest to the input vector according to a distance measure. The address of this selected codevector in the codebook is then transmitted to the VQ decoder over a communication channel and is employed by the decoder to retrieve the same codevector from the codebook in order to reconstruct an approximation of the corresponding input vector. The compression ability of VQ is obtained because the code bits required for transmitting the codevector index are significantly fewer in number than those of the input vector.

Formally, a vector quantizer can be regarded as a mapping Q from the k-dimensional space R^k to a finite subset Y of R^k. That is,

Received February 16, 1998; revised September 1 & October 20, 1998; accepted March 18, 1999.
Communicated by Zen Chen.

$Q: R^k \rightarrow Y,$

where $Y = \{y_i \mid i = 1...N; y_i \in R^k\}$ is called the codebook and N stands for the codebook size. For each input vector $x \in R^k$, the mapping Q assigns to x a codevector y_i from the codebook such that for a given distance measure $d(\cdot,\cdot)$, y_i is the codevector closest to x among all the codevectors in Y, i.e. $d(x, y_i) < d(x, y_j)$ $\forall j \neq i$. Most VQ systems define their distance measures as the squared Euclidean distance (SED):

$$d(x, y_i) = \sum_{j=1}^{k} (x_j - y_{ij})^2, \tag{1}$$

where x_j and y_{ij} denote the jth components of x and y_i, respectively.

For a fixed bit rate, better VQ performance can be obtained by increasing the vector size. However, this will lead to exponential growth in computational complexity grows exponentially. Searching for the closest codevector becomes an intensive computation process. To circumvent this problem, many fast algorithms have been developed, such as Bei and Gray's partial distortion search (PDS) [1], Huang and Chen's triangle inequality eliminating rule (TIE) [2], Ra and Kim's mean-ordered partial codebook search (MPS) [3], and others [4-13]. These algorithms provide both fast searching speed and reconstructed images which are identical in quality to that produced using the standard exhaustive algorithm. In MPS, Ra and Kim defined a very simple distance measure, the squared mean distance (SMD):

$$d_m(x, y_i) = \left(\sum_{j=1}^{k} x_j - \sum_{j=1}^{k} y_{ij} \right)^2 \tag{2}$$

and used it to reject codevectors that have much enough SMD to the input vector x without measuring the time-consuming SED. The success of MPS relies on the observation that the minimum SED codevector is probably nearby the closest SMD codevector. In some situations, however, there are a considerable number of codevectors that have small SMD but are very much different in the SED sense from the input vector e.g., those vectors showing complicated components like edges. Especially, when the source image is rather complicated, the effectiveness of SMD will decrease considerably. To alleviate this phenomenon, besides the block mean value, more features in a block of pixels should be extracted and utilized in a simple manner to speed up VQ.

A new algorithm is presented in this paper that is based on the SMD measure and incorporates the use of integral projections extracted as vector features from the blocks (vectors) of pixels in real images. While there are other excellent feature extraction techniques in the field of signal processing, a technique which uses integral projection will be simpler than others, especially in terms of the complexity of its VLSI circuit implementation.

Using integral projections, the present algorithm can reduce the VQ search complexity more than SMD can. In fact, integral projections have already been used in a variety of image coding applications, including the feature-based block-matching algorithm (FBBMA) [14], the feature-based VQ (FBVQ) [15] and the projection-VQ (PVQ) [16-18]. All these previous schemes merely work on the calculated projections instead of the original image pixels for the purpose of reducing the computation complexity. However, these feature-based algorithms cause some degree of degradation in terms of quality. Thus, this paper improves the optimality-preserving ability by using integral projections in VQ codebook search and illustrates the usefulness of integral projections in speeding up the codebook search process. In the next section, the definition of integral projections is stated, and the optimality-preserving ability of integral projections is derived. Section 3 gives experimental performance results for several images to examine the effectiveness of the proposed scheme and to compare it with other alternatives. The experimental results show that the present algorithm can reduce VQ search complexity more than can SMD, alone, when comparing with the full-search algorithm.

2. THE PROPOSED ALGORITHM

The objective of this section is to define integral projections and establishing search early-terminated conditions for integral projections so as to reduce the heavy computational load during VQ encoding.

A. Integral projections

Roughly speaking, the integral projections used in this paper can be defined as sums of gray-levels along any fixed direction in a block of pixels. For any specific block of pixels $x = [x_{ij}]$, where $i, j = 1, 2, ..., m$, two kinds of integral projections can be defined as follows:

(1) vertical projections (vp_{xj}): $vp_{xj} = \sum_{i=1}^{m} x_{ij}, \quad 1 \le j \le m$

(2) horizontal projections (hp_{xi}): $hp_{xi} = \sum_{j=1}^{m} x_{ij}, \quad 1 \le i \le m$

These projections possess several distinguishing features, such as edge location and orientation, and are likely to be different for different vectors. In [15], integral projections were interpreted as being capable of carrying information about the pixel value along the horizontal and vertical directions, respectively.

B. Fast closest codevector search using integral projections.

The basic idea behind the proposed fast search algorithm depends on construction of several searching constraints (test conditions). During the searching period, these test conditions are employed before the squared Euclidean distance is used in order to discard the codevectors which are not closest to the input vector. In addition to the SED and SMD measures in MPS, two additional distance measures based on the vertical and horizontal projections, respectively, are defined in the proposed algorithm. They are

$$d_v(x, y_i) = \sum_{l=1}^{m} (vp_{xl} - vp_{yil})^2, \tag{3}$$

$$d_h(x, y_i) = \sum_{l=1}^{m} (hp_{xl} - hp_{yil})^2. \tag{4}$$

Since integral projections are simple and relevant features of a block of pixels, the distance measures defined in (3) and (4) can help SMD provide a higher discarding rate for codevectors. In the following, we present a theoretical foundation for the usefulness of applying integral projections to the closest codevector search problem. Note that although the proof regarding SMD is well known, it is included in this paper because the proofs developed here are new and more straightforward compared with those in [3].

Lemma 1: If $a_1, a_2, ..., a_n$ are n arbitrary real numbers, then

$$(a_1 + a_2, \cdots, +a_n)^2 \le n(a_1^2 + a_2^2, \cdots, +a_n^2).$$

Proofs: For any pair of the real numbers (a_i, a_j), we have $(a_i - a_j)^2 \ge 0$, or equivalently,

$$a_i a_j \le \frac{a_i^2 + a_j^2}{2}. \tag{5}$$

Summing all the pairs of n real numbers on both sides of (5) yields

$$\sum_{i=1}^{n}\sum_{j=1}^{n} a_i a_j \le \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}(a_i^2 + a_j^2),$$

$$i.e., \sum_{i=1}^{n} a_i \sum_{j=1}^{n} a_j \le \frac{1}{2}\sum_{i=1}^{n}\left(na_i^2 + \sum_{j=1}^{n} a_j^2\right) = \frac{1}{2}\left(n\sum_{i=1}^{n} a_i^2 + n\sum_{j=1}^{n} a_j^2\right).$$

Therefore,

$$(a_1 + a_2, \cdots, +a_n)^2 = \sum_{i=1}^{n} a_i \sum_{j=1}^{n} a_j \le \frac{n}{2}\left(\sum_{i=1}^{n} a_i^2 + \sum_{j=1}^{n} a_j^2\right) = n(a_1^2 + a_2^2, \cdots, +a_n^2).$$

This completes the proof.

We are now ready to establish the additional search constraints based on the integral projections.

Theorem 1: Let x, y be two blocks m by m pixels in size, which are represented by $[x_{ij}]_{m\times m}$ and $[y_{ij}]_{m\times m}$, respectively, and the integral projections of which are $\{vp_{xj}, hp_{xi} \mid i, j = 1, 2 ... m\}$ and $\{vp_{yj}, hp_{yi} \mid i, j = 1, 2 ... m\}$. Then the following three inequalities hold:

(a) $d_m(x, y) \le m^2 d(x, y)$, (6)
(b) $d_v(x, y) \le md(x, y)$, (7)
(c) $d_h(x, y) \le md(x, y)$, (8)

where $d(x, y)$ stands for the SED between x and y, as defined in (1).

Proofs: Let $a_{ij} = x_{ij} - y_{ij}$. Then,

$$\left(\sum_{i=1}^{m}\sum_{j=1}^{m} a_{ij}\right)^2 = \left(\sum_{i=1}^{m}\sum_{j=1}^{m} x_{ij} - \sum_{i=1}^{m}\sum_{j=1}^{m} y_{ij}\right)^2 = d_m(x, y).$$

Applying Lemma 1, we have,

$$d_m(x, y) = \left(\sum_{i=1}^{m}\sum_{j=1}^{m} a_{ij}\right)^2 \le m^2 \sum_{i=1}^{m}\sum_{j=1}^{m} a_{ij}^2 = m^2 \sum_{i=1}^{m}\sum_{j=1}^{m} (x_{ij} - y_{ij})^2 = m^2 d(x, y).$$

This proves inequality (6). Next, to prove inequality (7), we first calculate the vertical projections of matrix $[a_{ij}]_{m\times m}$, obtaining $vp_{aj} = \sum_{i=1}^{m} a_{ij}$ and $(vp_{aj})^2 = (vp_{xj} - vp_y)^2 = (a_{1j} + a_{2j}, ..., + a_{mj})^2$.

Applying Lemma 1 again, we then have

$$d_v(x,y) \le m\sum_{j=1}^{m}(a_{1j}^2 + a_{2j}^2, \cdots, + a_{mj}^2) = md(x,y).$$

This completes the proof of (7). With the proof for (7), inequality (8) becomes self-explanatory.

Returning to the closest codevector search problem, let us assume that the integral projections of each codevector have been calculated and stored in advance, and that the currently known minimum squared Euclidean distance between an input vector x and a certain codevector is stored as $dis_{\min}$. The initial value of $dis_{\min}$ is determined as the SED between the input vector x and the codevector that has a minimum SMD with the input vector x. For an arbitrary codevector y_k, if any of the following test conditions holds:

(i) $m^2 dis_{\min} < d_m(x, y)$,
(ii) $m dis_{\min} < d_v(x, y)$, and
(iii) $m dis_{\min} < d_h(x, y)$,

then, after applying Theorem 1, codevector y_k can be discarded without calculating the SED because $d(x, y_k)$ can be induced to be greater than $dis_{\min}$ in this situation. For each codevector, test conditions (i) to (iii) are checked in a sequential manner. If either of the first two test conditions fails, then the codevector being considered should be further checked by the successive test conditions; otherwise it is rejected, and the next codevector in the codebook is checked again. In the situation where test conditions (i) to (iii) are all unsatisfied, the SED has to be measured to determine whether the codevector should be rejected or not. That is, to find the closest codevector, SED measurements are performed only on those codevectors whose integral projections lead to failure of all three test conditions (i)-(iii). Note that if each check made by the SED reveals that the codevector being examined is the newly closest one, the $dis_{\min}$ must be updated to become the minimum one found so far. Since the integral projections convey a significant information in the corresponding block and the arithmetic operations required for calculating the squared differences of the integral projections are much simpler than those needed for SED, a remarkable amount of computation can be saved.

In addition to the storage space needed to store the block means of codevectors, additional storage space is required to store the integral projections of codevectors.

3. EXPERIMENTAL RESULTS

Using computer simulation, the proposed algorithm, which is an extension of MPS, was written in the c language and run on a SunSparc10 workstation for fast VQ image coding. The results of the experiments are summarized in this section. As shown in Fig. 1, four monochromatic images, each 512 by 512 pixels in size with 256 gray-levels for each pixel, were used as the training set for designing codebooks. The well-known LBG algorithm [18] was used to design four codebooks with different sizes, ranging from 128 to 1024, using the same training set. These codebooks were then used to encode two test images, *Lenna* and *Bridge*. Table 1 shows the execution time for encoding *Lenna* with vector sizes 4 by 4, and 8 by 8. As can be seen, almost 97% of the execution time required

(a) Lenna

(b) X-ray

(c) F16

(d) Peppers

Fig. 1. The images used to design codebooks.

Table 1. Execution time (in seconds) required by the considered algorithms to code *Lenna*.

Codebook size	Full search		MPS		DTS		MAV		Proposed	
	4 × 4	8 × 8	4 × 4	8 × 8	4 × 4	8 × 8	4 × 4	8 × 8	4 × 4	8 × 8
128	75	75	5	6	5	4	3	3	3	3
256	150	150	10	11	7	7	5	5	5	5
512	302	302	17	21	13	11	10	9	9	9
1024	607	607	33	38	25	20	20	17	18	152

Table 2. Execution time (in seconds) required by the considered algorithms to code *Bridge*.

Codebook size	Full search		MPS		DTS		MAV		Proposed	
	4 × 4	8 × 8	4 × 4	8 × 8	4 × 4	8 × 8	4 × 4	8 × 8	4 × 4	8 × 8
128	75	75	9	10	7	7	5	6	4	6
256	150	150	16	19	12	12	9	10	8	9
512	302	302	29	38	22	23	17	20	15	18
1024	607	607	57	73	44	46	34	40	32	37

by the full search algorithm and up to 55% of the execution time required by MPS was saved by the proposed algorithm, especially for the larger vector size, 8 by 8. We also implemented two recent-developed alternatives: one named DTS (Double Test Search), uses the block maximum and minimum values to form its two test conditions; the other, called MAV (Mean and Variance), extracts the block mean and variance values and uses them to confine the search space. It is found that MAV exhibits performance comparable to that of the proposed algorithm. DTS is superior to MPS but inferior to MAV and the proposed algorithm. Table 2 lists the results for the image *Bridge*, which shows that more execution time, as compared to the case for *Lenna*, was required by the both algorithms. This is because *Bridge* is more complicated than *Lenna*, and which also is not included in the training set for the design of codebooks. From these tables, it is seen that incorporating horizontal and vertical projections into MPS leads to a drastic reduction in execution time for encoding relatively simple or high contrast images.

4. CONCLUSIONS

In this paper, the projection features of blocks of pixels have been utilized to improve the search speed of MPS. The original MPS achieves speedup of VQ by using only one test condition, which holds the rough feature of the block mean value in order to eliminate the need to compute the squared Euclidean distances for the rejected codevectors. Due to the relevance and compactness of integral projections, the proposed algorithm can refine the roughness of SMD, thus providing a noticeable reduction of execution time in image VQ encoding. The only disadvantage of the proposed algorithm is that in addition to the storage space required for the codebook, it needs additional storage space equal to about $2/m$ times the codebook size, assuming a block size of m by m pixels.

REFERENCES

1. C. Bei and R. M. Gray, "An improvement of the minimum distortion encoding algorithm for vector quantization," *IEEE Transactions on Communication*, Vol. 33, No. 10, 1985, pp. 1132-1133.
2. S. H. Huang and S. H. Chen, "Fast encoding algorithm for VQ-based image coding," *Electronics Letters*, Vol. 26, No. 19, 1990, pp. 1618-1619.

3. S. W. Ra and J. K. Kim, “A fast mean-distance-ordered partial codebook search algorithm for image vector quantization,” *IEEE Transactions on Circuit And System-Part II: Analog and Digital Signal Processing*, Vol. 40, No. 9, 1993, pp. 576-579.
4. M. R. Soleymani and S. D. Morgera, “A fast MMSE encoding algorithm for vector quantization,” *IEEE Transactions on Communication*, Vol. 37, No. 6, 1989, pp. 656-659.
5. M. R. Soleymani and S. D. Morgera, “An efficient nearest neighbor search method,” *IEEE Transactions on Communication*, Vol. 35, No. 6, 1987, pp. 677-679.
6. K. T. Lo and W. K. Cham, “Subcodebook searching algorithm for efficient VQ encoding of images,” *IEE Proceedings-I*, Vol. 140, No. 5, 1993, pp. 327-330.
7. L. Torres and J. Huguet, “An improvement on codebook search for vector quantization,” *IEEE Transactions on Communication*, Vol. 42, No. 2, 1994, pp. 208-210.
8. C. H. Lee and L. H. Chen, “Fast closest codeword search algorithm for vector quantization,” *IEE Proceedings-Visual Image Signal Processing*, Vol. 141, No. 3, 1994, pp. 143-148.
9. S. C. Tai, C. C. Lai, and Y. C. Lin, “Two fast nearest neighbor searching algorithm for image vector quantization,” *IEEE Transactions on Communication*, Vol. 44, No. 12, 1996, pp. 1623-1628.
10. A. Nyeck, H. Mokhari, and A. Tosser-Roussey, “An improved fast adaptive search algorithm for vector quantization by progressive codebook arrangement,” *Pattern Recognition*, Vol. 25, No. 8, 1992, pp. 799-802.
11. C. M. Huang, Q. Bi, G. S. Stiles, and R. W. Harris, “Fast full-search equivalent encoding algorithms for image compression using vector quantization,” *IEEE Transactions on Image Processing*, Vol. 1, No. 3, 1992, pp. 413-416.
12. C. C. Chang, R. F. Chang, W. T. Lee, and C. L. Kuo, “A fast search algorithm for vector quantization,” *Journal of Information Science and Engineering*, Vol. 12, No. 4, 1996, pp. 593-602.
13. W. Li and E. Salari, “A fast vector quantization encoding method for image compression,” *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 5, No. 2, 1995, pp. 119-123.
14. J. S. Kim and R. H. Park, “A fast feature-based block matching algorithm using integral projections,” *IEEE Journal on Selected Areas Communications*, Vol. 10, No. 5, 1992, pp. 968-917.
15. Y. C. Lin and S. C. Tai, “Fast feature-based vector quantization algorithm of image coding,” *Optical Engineering*, Vol. 34, No. 10, 1995, pp. 2918-2926.
16. H. B. Park and C. Wang, “Image compression by vector quantization of projection data,” *IEICE Transaction on Information and System*, Vol. E75-D, No. 1, 1992, pp. 148-155.
17. K. H. Jung and C. Wang, “Projective image representation and its application to image compression,” *IEICE Transactions on Information and System*, Vol. E79-D, No. 2, 1996, pp. 136-142.
18. Y. Linde, A. Buzo, and R. M. Gray, “An algorithm for vector quantization design,” *IEEE Transactions on Communication*, Vol. 28, No. 1,1980, pp. 84-95.

Yih-Chuan Lin (ªL©ö¬u) received the BS degree in applied mathematics from the Chinese Culture University, Taipei, Taiwan, in 1990 and the MS and Ph.D. degrees in electrical engineering from National Cheng Kung University, Tainan, Taiwan, in 1992 and 1997, respectively. He is currently an associate professor of computer science and information engineering at the Shu-Te University of Techndogy.