Vector Quantization Fast Search Algorithm using Hyperplane Based k-dimensional Multi-node Search Tree

Kam-Fai Chan Alton, Kam-Tim Woo and Chi-Wah Kok
Depart of EEE, Hong Kong University of Science and Technology, Clear Water Bay, Kowloon, HONG KONG

Abstract

A vector quantization fast search algorithm using hyperplane based k-dimensional multi-node search tree is presented. Misclassification problem associated with hyperplane decision is eliminated by a multi-level back-tracing algorithm. The vector quantization complexity is further lowered by a novel relative distance quantization rule. Triangular inequality is applied to lower bound the search distance, thus eliminated all the sub-tree in the k-dimensional search tree during back-tracing. Vector quantization image coding results are presented which showed the proposed algorithm outperform other algorithms in literature both in PSNR and computation time.

1 Introduction

Vector Quantization (VQ) is a useful data compression tool for speech and image coding. A vector quantizer, $Q(\cdot)$, maps the k-dimensional (k-d) Euclidean space, $\mathbb{R}^k$, into the codebook C:

$$Q : x \in \mathbb{R}^k \longrightarrow C_j \in C \qquad \text{with } j \in [1, 2, \cdots, N]$$

where x is the input vector, the codebook is a collection of N codevectors, $C_i : i = 1, 2, \cdots, N$, in $\mathbb{R}^k$, and j is the codeword index. N is also known as the size of the codebook C. The codeword index can be transmitted or stored, and thus achieve compression. The reconstructed signal at the output of the VQ decoder is given by C_j which is one of the codevectors in C indexed by j, the output codeword index from the VQ encoder. The VQ encoder is optimal when the chosen codevector C_j for input signal x minimize a given distance measure $d(x, C_j)$. Different distance function has been investigated for various applications. In this paper, we will concentrate on the Euclidean distance which founds applications in image coding [1] etc.

Although VQ is an efficient compression tool, it has very high computational complexity which hindered it's applications in everyday life. The high computational complexity of VQ is the result of the quantizer which computes the distance between the input vector x and every codevectors in the codebook C. Various researches have been dedicated to search for an efficient vector quantizer, which can be roughly classified into two categories

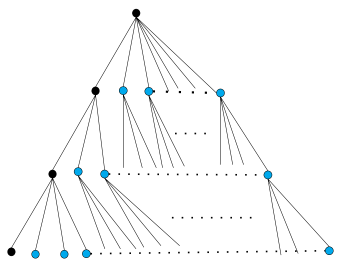

Figure 1: Misclassification Problem

1. An efficient codebook structure [6]
2. An efficient distance computation method [8]

Tree structured VQ proposed in [6] belongs to the first category. The tree structured VQ is further developed in [3, 5], where k-d tree structured VQ is proposed. The k-d tree is a multi-node tree in which each non-terminal node has multiple descendants [10] as shown in Fig.1. The non-terminal node will therefore partition the voronoi cell into groups according to the cell position with respect to the partition hyperplane, H_i, $i \in [1, 2, \cdots, n-1]$, which are parallel to each other, and results in n branches spanning the partition node. The hyperplane H_i in k-d space is defined by the normal vector h and position vector c_i as

$$H_i = \{x \in \mathbb{R}^k : h^T x + c_i = 0\} \tag{1}$$

The generation of the partition hyperplanes will be discussed in Section 2. The nearest neighbor search algorithm descends along the non-terminal nodes of the k-d tree to one of the child node S_i according to the hyperplane decision rule

$$S_i \quad : \quad \left\{c_{i-1} \le h^T x < c_i\right\}, \tag{2}$$

with $c_0 = -\infty$, and $c_M = \infty$ as shown in Fig.2.

When the search reaches a terminal node, the codevectors in the terminal node are examined exhaustively to find the one closest to the query vector Without loss of generality, the following will assume the terminal node contains one codevector. Comparison between

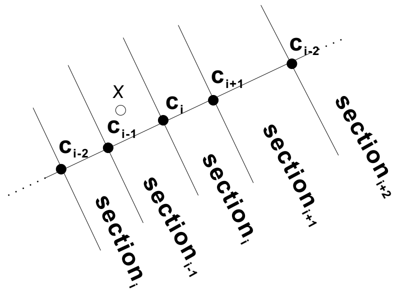

Figure 2: multi-levels formed by hyperplanes

binary tree and multi-node tree with the same number of terminal node shows that the computation complexity can be reduced dramatically. For example, the quadtree hyperplane search algorithm for codebook size 256 has an average depth of 4 compared to 8 for that of the binary tree hyperplane search algorithm. Since the average number of distance computation, eq.(2), is linearly proportional to the average tree depth. Therefore it can be concluded that there will be a foreseeable reduction in the computational complexity between binary tree search to quadtree search. Unfortunately, hyperplane based clustering suffers from misclassification error with respect to the nearest neighbor rule [3]. A simple back-tracing algorithm has been proposed in [3] to eliminate hyperplane based misclassification. The back-tracing algorithm exhaustively compare codewords along the back-tracing path which inevitably increased the computational complexity of the VQ process. Such increases in computational complexity will become a bigger problem with the increase in the number of branches for each node in the search tree. Because along the back-tracing path, there are more vectors required to be compared than that of the binary tree.

2 Multi-Node Search Tree Generation

The VQ k-d binary search tree is constructed by hyperplane that splits the input data set into equally populated non overlapping data set according to the hyperplane decision rule in eq.(2) and the cutting point set c_i [13]. If c_i is the centroid of S_i,the hyperplanes can be considered as the bisector of centroid which result in equally populated partition. The orientation of the hyperplane is chosen to be perpendicular to the principle component of the input covariance matrix R

$$R = E(x - \bar{x})(x - \bar{x})^T \tag{3}$$

where E is the expectation operator, $\bar{x}$ is mean of input training vector x, and T denotes transposition. The principle component of R is the eigenvector associates with the largest absolute eigenvalues. The hyperplane is constrained to pass through the centroid of the input data set such that equally populated partition is obtained. To growth the tree, cluster in one of terminal node will be chosen for further partition. Follow the results from [11] the terminal node partition that results with the largest reduction in total sum of square errors (SSE) will be chosen for partitioning. Such that the generated tree will minimize the total SSE.

Let's consider the quad tree as an example to simplify our discussion on multi-node search tree generation. Based on the bipartition idea, the training set will be divided into 4 equally populated intervals in partition nodes. To growth the tree, the variance of the data set associated with each terminal nodes are computed. The one with the largest variance are partitioned, hopping that the partitioning will lower the variance of the partitioned data sets, and thus reduce the quantization error as mentions before. The above partitioning will be repeated until the number terminal nodes reaches the desired codebook size, e.g. 256. Noted that the above quadtree growth will method will result in a minimum total variance search tree with each node being partitioned with equally populated branches. As a result, the generated quadtree may not minimize the total SSE, and thus may not be optimal as will be discussed in Section 3.

2.1 Eliminating Misclassification

The VQ encoding process begins by tracing the k-d tree according to the hyperplane decision rule, eq.(2). Unfortunately, this simple decision rule may misclassify the input query vector in the nearest neighbor sense as discussed in [3] which is also illustrated in Fig.3. The query vectors Q_1 and Q_2 are both lie on the left hand side of the hyperplane H, as does the codevector A. However, it is the codevector B that is closer to the query record Q_1 than A as observed by comparing the radius d_A and d_B of the two circle that enclose the query vector with center at A and B, respectively.

2.1.1 Back-tracing

Back-tracing has been proposed in [3, 4] to overcome hyperplane associated misclassification. A more general back-tracing process is considered in this paper for multi-node hyperplane search tree. Let's consider the back-tracing process for quadtree as an example which is illustrated in Fig.3 for various multi-level backtracing search paths. The simplest back-tracing process is shown in Fig.3a. During VQ encoding, when the k-d search tree has reached a terminal node, back-tracing will search the neighboring branch of the k-d tree until it reach another terminal node. The Euclidean distance among the codevectors in the four terminal nodes and the query vector will be compared. We called this kind of back-tracing as zero level back-tracing. This is because, it is obvious that the hyperplane decision

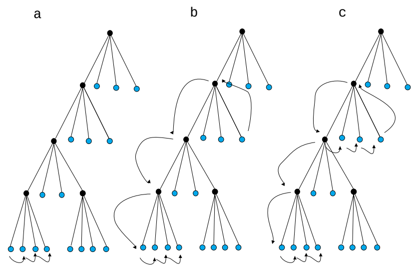

Figure 3: 3 different level back-tracing Code Trees

during back-tracing will also suffer from misclassification. Multi-level back-tracing can overcome such misclassification problem. Shown in Fig.3b is the one level back-tracing which will found another terminal node in the neighboring branch associated with the terminal node of the k-d tree search result according to the zero level back-tracing criteria. The distance between all the searched terminal nodes with the query vector will be compared and the smallest one will be chosen as the quantization codevector. As a result, one level back-tracing can reduce the probability of misclassification better than zero level back-tracing. In general, a n level back-tracing will search all the terminal nodes in $n+1$ tree branches which is neighbors to the terminal node of the multinode k-d tree search result. Showing in Fig.3c is an example of the 2-level back-tracing. Noted that the n level back-tracing will stop whenever, the branches share the same parent. In that case misclassification will not be possible. We adopted triangular inequality technique [10] in the back-tracing procedure, such that it eliminates sibling nodes during back-tracing by lower bounding the distance from the query points to the sibling nodes. When the distance is greater than the k-th nearest neighbor found so far, further search on the sibling sub-tree will not be necessary [9].

2.2 Relative Distance Quantization

A relative distance quantization method is proposed in this section to compensate for the increased computational complexity in the back-tracing process. Consider Fig.4, where A and B are the codevectors, and Q_1 is the input query vector. We can decide if Q_1 is closer to A or B by considering P, the projection point of Q_1 on the line AB. Let K be the ratio of the one dimensional distance AP to AB along the line AB as

$$AP = K(AB) \tag{4}$$

$$\Rightarrow \quad P = K(AB) + A \tag{5}$$

Substite P from eq.(5) into the projection equation

$$AB \cdot (Q_1 - P) = 0 \tag{6}$$

$$\Rightarrow \quad AB \cdot Q_1 - AB \cdot (K(AB) + A) = 0 \tag{7}$$

$$K = \frac{AB \cdot (Q_1 - A)}{\mid AB \mid^2} \tag{8}$$

When $K \leq 0.5$, AP is longer than BP. By the property of right angle triangle, $K \leq 0.5$ implies AQ_1 is longer than BQ_1. Thus the nearest neighbor of Q_1 is given by the decision rule

$$A \quad : \quad \{K \leq 0.5\} \tag{9}$$

$$B \quad : \quad \{K > 0.5\} \tag{10}$$

The computation complexity of the relative distance based vector quantization method has $O(k)$, where only $2k$ additions and $N+1$ multiplication is required, compared to that of direct Euclidean distance computation which requires $2k$ multiplications and $2k-2$ additions and has $O(2k)$. Note that the line AB, and distance AB are assumed to have been pre-computed and stored in the VQ encoder. With dimension 16 in the image coding simulations in Section 3, substantial reduction in the computation times have been achieved. Note that the relative distance based quantization method is applicable in the discussed back-tracing problem, and all other nearest neighbor search problem.

2.3 Combining the Two Methods

The above two VQ techniques can be combined to reduce the computation complexity and overcome the misclassification problem in multi-node k-d tree based VQ. The relative distance quantization is applied to the terminal nodes in multi-level back-tracing which compensate for the increased complexity of the back-tracing algorithm for the elimination of misclassification associated with hyperplane decision rules.

3 Image Coding Results

Image coding results are presented to evaluate the performance of the proposed vector quantizer by measuring the number of floating point operations used in the quantization process and the peak signal to noise ratio (PSNR) of the quantized images. Various vector quantizers are implemented by Matlab ver. 6 on a Pentium 4 1.5GHz PC with 128MB memory. The PSNR is defined as

$$PSNR = 10 \log_{10} \frac{255^2}{\frac{1}{256 \times 256} \sum_{i=0}^{255} \sum_{j=0}^{255} (x_{i,j} - \hat{x}_{i,j})^2} (dB)$$

where $x_{i,j}$ and $\hat{x}_{i,j}$ are the $(i,j)^{th}$ pixel of the original and the decoded image respectively. The vector size is chosen to be 4×4, such that with a codebook size 256, the mean residual VQ image codec has a compression ratio of 8, i.e. 1 bit/pixel. The training set has 4×4096 vectors extracted from 4 $256 \times 256 \times 8$-bit grey-level images of human portrait ("Lenna", "Tiffany", "Barbara", "Zelda"). The training set is used to generate a quadtree with arbitrary hyperplane as discussed in Section 2.

The PSNR of various VQ encoded images using the proposed VQ algorithm and algorithms in [2, 3] are listed in Table 3. The results showed that the back-tracing algorithm can achieve better quantization results in both binary and multi-node search trees. The

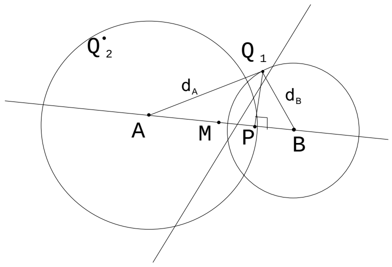

Figure 4: Misclassification Problem

	Lenna		Boat	
Algorithm	Flops	PSNR	Flops	PSNR
Binary	1830156	29.938	1821959	26.776
Binary(0 level)	2618376	29.958	2612788	27.004
Quad	1524051	27.909	1515446	27.157
Quad(0 level)	2110782	27.961	2099864	27.251
	Airplane		Pepper	
Binary	1807485	26.522	1820760	27.410
Binary(0 level)	2591350	26.698	2791977	28.169
Quad	1502189	26.628	1515934	27.582
Quad(0 level)	2085579	26.722	2075937	27.654

Table 1: The PSNR (dB) of various image coding results using different vector quantization methods.

PSNR performance of the VQ image coding results obtained from that of binary tree and quadtree are compatible. Although in several cases the performance of the quadtree are not as good as that of the binary tree, this is due to the sub-optimal hyperplane generation algorithm used in the simulations. The optimal hyperplane generation method discussed in [11] should improve the performance of the quadtree based VQ performance to be close to that of binary tree. However, since this paper discusses only the fast search algorithm as compared to [10], optimal hyperplane generation method is out of scope and will be report in a separate paper.

Also observed in Table 3 is the computational complexity of various VQ algorithms by observing the number of floating point operations reported by Matlabs. The number of floating point operations dramatically reduced from the average of 2373332 to 1667248 by converting a binary tree to quadtree. The VQ image encoding time reduces from an average of 213 seconds for full search VQ to 25 seconds of the proposed quadtree algorithm with back-tracing.

The reduction in computational complexity of the proposed algorithm is achieved by the multi-node search tree, relative distance quantization process, and the triangular inequality for lower bounding the search distance. It is observed that, a large number of back-tracing is required to eliminate hyperplane related misclassification. The application of relative distance quantization effectively reduced the computational complexity for each back-tracing process by an order of magnitude. Furthermore, the application of triangular inequality also helps to remove 10% of the total possible back-tracing paths, and thus lowered the computation time of the proposed algorithm.

4 Conclusions

An arbitrary hyerplane based multi-node k-d tree search algorithm is proposed. Multi-level back-tracing is applied to eliminate hyperplane associated k-d tree VQ misclassification problem. Relative distance decision rule is applied to further reduces the computational complexity in the VQ process. Triangular inequality is used to lower-bound the search distance and thus eliminated a substantial amount of sub-tree in the k-d tree during back-tracing. Image coding results are presented with and without multi-level back-tracing, which showed that the proposed VQ algorithm can achieve comparable PSNR with lower computational complexity.

References

[1] M.Nasrabadi and R.A.King, "image coding using vector quantization: a review," *IEEE Trans. on Communication*, pp.957-971, Aug. 1988.

[2] R.Gray, "Vector Quantization," *IEEE ASSP Mag.*, pp.4-29, Apr. 1984.

[3] S.C.Chan, Ted C.W.Kok, and S.W.Chau, "Codebook Generation and Search Algorithm for Vector Quantization Using Arbitrary Hyperplanes" *Proc. IEEE ISCAS*, vol.3, pp.1885-1888, 1993.

[4] R.F.Sproull, "Refinement to nearest-neighbour searching in k-d trees," *Algorithmica,* vol.6, pp.579-589, 1991.

[5] S.J.Wan, S.K.M.Wong and P.Prusinkiewicz, "An algorithm for multidimensional data clustering," *ACM Trans. on Mathematical Software*, Vol.14, No.2, pp.153-162, 1998.

[6] Makhoul and S.Roucos, and H.Gish, "Vector quantization in speech coding," *Proc. IEEE*, vol. 73, pp.1551-1558, 1995.

[7] L.Guanm and M.Kamel, "Equal-Average Hyperplane Partitioning method for Vector Quantization of Image Data", *Pattern Recognition Letters*, pp.693-699, 1992.

[8] C.K.Chan and C.K.Ma, "A Fast Method of Designing Better Codebooks for Image Vector Quantization", *IEEE Trans. on Communication* vol.42, Fe, 1994, pp. 237-242.

[9] D.Y.Cheng, A.Gersho, B.Ramamurthi, and Y Shoham, "Fast Search Algorithm for Vector Quantization and Pattern Matching," *Proc. IEEE Int'l Conf. Acoustics, Speech, and Signal Processing*, vol.1, pp.9.11.1-9.11.4, Mar. 1984.

[10] J.McNames, "A fast nearest-neighbor algorithm based on a principal axis search tree, ",*IEEE Trans. on Pattern Analysis and Machine Intelligence*, pp. 964-976, Sept. 2001.

[11] Kam-Tim Woo, Kam-Fai Chan Alton and Chi-Wah Kok, "Codebook Generation using Adaptive Arbitrary Hyperplane Algorihm," *submitted to ISIT-2002*, Oct. 2001.

[12] Kam-Tim Woo, Kam-Fai Chan Alton and Chi-Wah Kok, "Vector Quantization Fast Search Algorithm," *submitted to ISCAS-2002*, Nov. 2001.

[13] H.Ferhatosmanoglu, E.Tuncel, D.Agrawal and A.E.Abbadi, "Vector Approximation based Indexing for Non-uniform High Dimensional Data Sets," *Proc. Ninth Intern'l. Conf. Inform. Know. Management*, 2000, pp.202-209.