HIGH DIMENSION ADAPTIVE VECTOR QUANTIZATION

Tzung Kwang Wang, John Foster, Shahryar S. Ardalan

Department of Electrical Engineering
North Carolina A&T State University
Greensboro, North Carolina 27411

ABSTRACT

It is well known that block coding systems have displayed better performance at a given bit rate, or require lower bit rate at a given performance criteria than scalar coding schemes [1-5]. For block coding systems Vector Quantization (VQ) has achieved performance near rate-distortion function [6]. The major disadvantage of VQ with stationary codebook was substantial degradation in quality for out-of-training-sequence inputs. Also the training requirements of regular VQ consumes tremendous amounts of CPU cycles, thereby making it unsuitable for real-time adaptation for digital communication applications.

INTRODUCTION

One major criteria for digital waveform coders is to provide the maximum signal fidelity - signal-to-noise ratio (SNR) - for a given transmission rate. Many coding algorithms have evolved over the past several decades which try to achieve this goal. Pulse Code Modulation (PCM), Delta Modulation (DM), Adaptive Delta Modulation (ADM), and Adaptive Delta Pulse Code Modulation (ADPCM) have received wide acceptance with digital voice transmission systems and are considered the benchmark standards for digital waveform coding [1-4].

Recent advances in the computational capabilities of communication electronics have enabled research towards more complex coding algorithms which either need large amounts of memory or high computational capability [5-7]. Due to the evolution of Application Specific Integrated Circuit (ASIC) chips, many of these new generation algorithms are within the realm of real-time capability and are easily implemented on a single chip.

Vector Quantization (VQ) is a block coding technique which has been implemented for both source coding and waveform coding systems [9-11]. Due to the vector coding nature of the coding algorithm, the VQ SNR performance is superior to scalar quantizers such as PCM, DM, ADM, and ADPCM [12]. The advantage of the VQ coding algorithm can be used in either of two ways: i) for a fixed bit-rate yield a superior SNR versus other systems, or ii) for a fixed SNR yield a lower required bit-rate versus other systems.

Although VQ has demonstrated superior performance over scalar coding system two major disadvantages have inhibited its acceptance. Inherent to VQ systems are the codebook of optimum signal shapes called codewords. These codewords are found by an iterative design procedure which finds locally optimum shapes based on a training signal. Although the VQ codebook works well within the trained data, outside-of-training-sequence tests can yield sub-par performance. Secondly, because of the fixed codebook quality of VQ systems, the hope for a universal codebook means using a training sequence of tremendous length containing all types of speakers. Even with a very long training sequence of data, there still could be non-typical sequences which will make the VQ SNR fall below an acceptable level.

A method which could adapt the codebook to the changing statistics of the waveform would eliminate the primary disadvantage of the VQ coding scheme. This attribute is precisely what Adaptive Vector Quantization (AVQ) attempts to achieve. With the addition of a dynamically changing codebook algorithm AVQ can be used as a universal coder.
ADAPTIVE VECTOR QUANTIZATION SYSTEM DESCRIPTION

The AVQ system description is shown in Figure 1. The AVQ coding process can be described as follows:

Encoder
i) Given an initial codebook $C = \{c_1, c_2, c_3, \ldots c_M\}$ of $M - K$ dimension codewords, and an $L$-bit quantizer for error quantization.
ii) Encode $M$ input samples by choosing the nearest codeword from the AVQ codebook using Euclidian distance as the distortion measure.
iii) Quantize the error between the input vector and the optimum codeword to produce an error vector.
v) Transmit the Log($M$) bits corresponding to the optimum codeword and the quantized error bits to the decoder.
v) Update the codebook by adding the quantized error to the current optimum codeword to achieve a new codeword. Go back to step ii).

Decoder
i) Given an initial codebook $C = \{c_1, c_2, c_3, \ldots c_M\}$ of $M - K$ dimension codewords, and an $L$-bit quantizer for error quantization.
ii) From the received Log($M$) codebook bits and the $L*K$ error bits, create an estimation of the original input vector by adding the current codeword to the quantized error vector.
iii) Update the codebook by adding the quantized error to the currentcodeword to achieve a new codeword. Go back to step ii).

The AVQ coding algorithm offers several advantages over standard VQ. First, as with many adaptive schemes, the choice for the initial codebook is not critical. After a short transition period the codebook adapts itself to the quantized version of the past $L$-input samples. This attribute eliminates the need for the computationally intensive codebook design algorithm used with standard VQ.

Secondly, the AVQ coding process can adapt to different types of signals. This adaptation process is very similar to ADM or ADPCM scalar systems. In fact AVQ can be interpreted as a vector implementation of ADPCM.

Thirdly, due to the decrease in computational load, AVQ can be more easily implemented via ASIC chips in a real-time system. This feature may make AVQ systems attractive towards standard digital coding networks. The error quantizer is a vector error quantizer which adapts to the energy of the chosen codeword.

CODING COMPLEXITY

The coding complexity can be measured in terms of two items: computational complexity and storage requirements.

Given a $k$-dimensional AVQ system with rate $R_c$ (codebook) and $R_e$ (error), the required comparisons for each vector is given by:

$$\# \text{comparisons} = 2^{R_c k}$$

These comparisons must be done each $k/fs$ seconds where $fs$ if the sampling frequency. Therefore the number of comparisons/second is given by:

$$\frac{\text{comparisons}}{\text{second}} = 2^{R_c k} \frac{fs}{k}$$

14
# comparisons/sec = (fs2**Rck)/k \hspace{1cm} (2)

Since each comparison involves a euclidean distance computation, \( k \) multiplies and \( k \) additions must be done. Since multiplies are an order of magnitude more expensive in actual systems the number of multiplies/second needed for AVQ is given by:

\[ \# \text{ multiplies/second} = fs2^{*}Rck \hspace{1cm} (3) \]

The amount of storage needed is solely for the dynamic codebook. Assuming \( M \) bits per coefficient the storage requirements are given as follows:

\[ \# \text{ storage} = M^{*}RcK \hspace{1cm} (4) \]

For an \( k=16 \), \( Rc = 1 \) bps system, the amount of multiplies per second and storage requirements are 512,000,000 and 128,000 respectively.

**SIMULATION RESULTS**

The programming environment used for computer simulation was a Sun 4/280 with UNIX operating system. C was used as the source code. For higher dimensions the Cray II YMP was used with a vectorized C used as the source code. A speech waveform sequence of 500,000 samples was used as the source. Both quantitative and subjective tests were done to compare the reconstructive speech with the original waveform. The transmission rate chosen was 16 Kbps. A even distribution of 1 bit per sample was used for the codeword index and error quantization. The SNR vs. Dimension comparison is shown in Figure 2.

Five hundred thousand samples were used as the input signal length. The original and reproduced speech signals were sampled at 8 KHz. The results of Figure 2 show an increase in SNR with an increase in dimension - \( k \) for a fixed bit rate.

Subjective tests were conducted to rank listener preference of the signals. The objective and subjective rankings are shown in Table 1.

**CONCLUSIONS**

In this paper Adaptive Vector Quantization was presented as a waveform coding system with application to digital voice transmission.

![SNR vs. Dimension Size, Rc = 1, Re = 1](image)

Figure 2 - AVQ SNR vs. Dimension Size, \( Rc = 1 \), \( Re = 1 \)

<table>
<thead>
<tr>
<th>Dimension</th>
<th>Ranking</th>
</tr>
</thead>
<tbody>
<tr>
<td>22</td>
<td>1</td>
</tr>
<tr>
<td>20</td>
<td>2 (tie)</td>
</tr>
<tr>
<td>18</td>
<td>2 (tie)</td>
</tr>
<tr>
<td>16</td>
<td>3</td>
</tr>
<tr>
<td>14</td>
<td>4</td>
</tr>
<tr>
<td>12</td>
<td>5</td>
</tr>
<tr>
<td>10</td>
<td>5 (tie)</td>
</tr>
</tbody>
</table>

The AVQ coding algorithms were modeled and simulated on a computer. The results show high SNR performance (above 20 dB @ \( R=2 \)) with relatively low coding complexity. Future research will involve several areas. Further work with vector error quantizers and higher dimensions may yield AVQ systems which can be used at lower bit rates (less than 8 Kbps).

**REFERENCES**


