Multi Switched Split Vector Quantizer

M. Satya Sai Ram, P. Siddaiah, and M. Madhavi Latha

Abstract—Vector quantization is a powerful tool for speech coding applications. This paper deals with LPC Coding of speech signals which uses a new technique called Multi Switched Split Vector Quantization, This is a hybrid of two product code vector quantization techniques namely the Multi stage vector quantization technique, and Switched split vector quantization technique. Multi Switched Split Vector Quantization technique quantizes the linear predictive coefficients in terms of line spectral frequencies. From results it is proved that Multi Switched Split Vector Quantization provides better trade off between bitrate and spectral distortion performance, computational complexity and memory requirements when compared to Switched Split Vector Quantization, Multi stage vector quantization, and Split Vector Quantization techniques. By employing the switching technique at each stage of the vector quantizer the spectral distortion, computational complexity and memory requirements were greatly reduced. Spectral distortion was measured in dB, Computational complexity was measured in floating point operations (flops), and memory requirements was measured in (floats).

Keywords—Unconstrained vector quantization, Linear predictive Coding, Split vector quantization, Multi stage vector quantization, Switched Split vector quantization, Line Spectral Frequencies.

I. INTRODUCTION

Most forms of speech coding techniques are based on lossy algorithms. Lossy algorithms are considered acceptable when encoding speech because the loss in quality is undetectable by the human ear. Uncompressed speech is usually transmitted at 64 kb/s, using 8 bits/sample and at a rate of 8 KHZ for sampling. Any bit rate below 64 kb/s is considered as compression. This paper deals with a lossy compression technique called Linear predictive coding [1]-[3] which uses a vector quantization technique[4]-[6] called Multi switched split vector quantization. Multi switched split vector quantization technique is a hybrid of Multi stage vector quantization technique (MSVQ) [7]-[9], and Switched split vector quantization technique (SSVQ) [7]. As quality, complexity and memory requirements have a direct impact on marketability and cost of the under laying products or services, the performance of MSSVQ is evaluated by using the spectral distortion, computational complexity and memory requirements. In MSSVQ a number of Multi stage vector quantizers are connected in cascade where the difference between the input vector and quantized vector of one stage is fed as an input to the next stage. Each stage of MSVQ employs SSVQ using soft decision in which a number of codebooks are connected in parallel. In this paper two codebooks connected in parallel are taken so as to maintain a tradeoff between the switch bits and the number of codebooks to be searched at each stage of Multi stage vector quantizer i.e., when two codebooks are connected in parallel with soft decision scheme the input vector is quantized by using the two codebooks connected in parallel, where as with hard decision scheme the input vector is quantized in only one of the two codebooks. As only one bit is required for the two switches with both the soft and hard decision schemes, there can be an improvement in spectral distortion performance with soft decision scheme for the same computational complexity and memory requirements when compared to hard decision scheme, so this paper deals with MSSVQ using a soft decision scheme.

MSSVQ algorithm mainly consists of the following steps:

a) Select a switch
b) Generate the codebook from the training sequence
c) Obtain the quantized vectors from from the codebook generated
d) Extract the new trained sequence from the old and quantized training sequence.
e) Repeat steps a to d for all the switches of a stage.
f) Obtain the approximate vector at each stage.
g) Repeat steps a to f for the required number of stages
h) Finally obtain the approximate of the input vector by summing the approximate vectors at each stage.

The aim of this article is to provide a general review of MSSVQ, and to compare its performance with other existing product code vector quantization techniques. The practical limitations, regarding computational complexity and memory requirements as a function of bit rate are discussed. The spectral distortion performance of MSSVQ is evaluated in LSF parameter quantization [10]-[12] for narrow band speech coding.

II. MULTI SWITCHED SPLIT VECTOR QUANTIZATION

The basic idea of MSSVQ is to use p stages, m switches and s splits, and its goal is to reduce the spectral distortion, computational complexity, and memory requirements this is achieved by the use of two product code vector quantization techniques MSVQ and SSVQ. SSVQ is a hybrid of switch vector quantization and split vector quantization [13]-[14] techniques. The use of split vector quantizer makes the less availability of bits at each split of the vector quantizer as a result the complexity and memory requirements were greatly
reduced but the dependencies that exists across the dimensions (splits) of a vector will be lost as a result the spectral distortion will be slightly increased. With the use of Multi stage vector quantizer the number of bits used for quantization will be divided at each stage of the vector quantizer as a result the complexity and memory requirements can be reduced greatly, likewise the use of a switch vector quantizer exploits the correlation that exists across all dimensions of a vector quantizer.

At each stage of SVQ the 10-dimmensional LSF vector is split into 3 parts of 3, 3, 4 divisions respectively. During codebook generation bits are allocated depending on the frequency of the LSFs. Preference is given to high frequency LSFs, when the number of bits is not divisible by 3. For a particular switch the generation of codebooks at different stages is shown in Fig. 1.

![Fig. 1 Codebook Generation at different stages](image)

- Initially the codebook at the first stage is generated by using the Linde, Buzo and Gray (LBG) [15] algorithm with the training vectors set as an input.
- Secondly the training difference vectors are extracted from the input training vectors set and the quantized training vectors of the first stage.
- Finally the training difference vectors are used to generate the codebook of the second stage.

This procedure is continued for the required number of stages. Finally the decoder takes the indices, $i_I$, from each stage and adds the quantized vectors at each stage so as to obtain the reconstructed vector $\hat{x}$ given by

$$\hat{x} = Q[x_1] + Q[e_1] + Q[e_2] + \ldots.$$  

Where $Q[x_1]$ is the quantized input vector at the first stage, $Q[e_1]$ is the quantized error vector at the second stage and $Q[e_2]$ is the quantized error vector at the third stage and so on. As this process involves the quantization of the error vectors and summing of the error vectors with the approximate vector at the first stage the spectral distortion performance can be greatly improved when compared to SSVQ and SVQ.

### III. COMPLEXITY AND MEMORY REQUIREMENTS

The computational complexity of a Switch vector quantizer is given by:

$$\text{Complexity}_{\text{switch}} = 4n^2 b_m^m - 1$$  \hspace{1cm} (1)
The computational complexity of a Split vector quantizer, is given by:

\[ \text{Complexity}_{SVQ} = \sum_{i=1}^{s} (4n^2 2^{b_i} - 1) \]  

(2)

The computational complexity of a Multistage vector quantizer is given by:

\[ \text{Complexity}_{MSVQ} = \sum_{i=1}^{n} (4n^2 2^{b_i} - 1) \]  

(3)

The computational complexity of a Switched split vector quantizer is given by:

\[ \text{Complexity}_{SSVQ} = (4n2^{b_m} - 1) \sum_{i=1}^{s} (4n^2 2^{b_i} - 1) \]  

(4)

The computational complexity of a Multi switched split vector quantizer is given by:

\[ \text{Complexity}_{MSSVQ} = p(4n2^{b_m} - 1) + \sum_{i=1}^{p} \left( \sum_{j=1}^{s} (4n^2 2^{b_j} - 1) \right) \]  

(5)

The memory requirements of a Split vector quantizer is given by:

\[ \text{Memory}_{SVQ} = n2^{bm} \]  

(6)

The memory requirements of a Multistage vector quantizer is given by:

\[ \text{Memory}_{MSVQ} = \sum_{i=1}^{m} n2^{b_i} \]  

(7)

The memory requirements of a Switched split vector quantizer is given by:

\[ \text{Memory}_{SSVQ} = \sum_{i=1}^{m} n2^{b_i} + 4n2^{b_m} \sum_{i=1}^{s} n_i 2^{b_i} \]  

(8)

The memory requirements of a Multi switched split vector quantizer is given by:

\[ \text{Memory}_{MSSVQ} = \sum_{i=1}^{m} n2^{b_i} + 4n2^{b_m} \sum_{i=1}^{p} \left( \sum_{j=1}^{s} n_i 2^{b_j} \right) \]  

(9)

where \( n \) is the dimension of the vector \( b_m \) is the number of bits allocated to the switch vector quantizer \( m = 2^{b_m} \) is the number of switching directions, \( p \) is the number of stages \( s \) is the number of splits.

IV. SPECTRAL DISTORTION

In order to objectively measure the distortion between a coded and uncoded LPC parameter vector, the spectral distortion is often used in narrow band speech coding. For the \( i^{th} \) frame the spectral distortion (in dB), \( SD_i \) [16] is defined as:

\[ SD_i = \frac{1}{(f_s - f_l)} \int_{f_l}^{f_s} \left[ \frac{1}{10 \log_{10} \left| x_i(f) \right|} - \frac{1}{10 \log_{10} \left| \hat{x}_i(f) \right|} \right]^2 \, df \, (dB) \]  

(10)

Where \( f_s \) is the sampling frequency and \( x_i(f) \) and \( \hat{x}_i(f) \) are the LPC power spectra of the uncoded and coded \( i^{th} \) frame, respectively. \( f \) is the frequency in Hz, and the frequency range is given by \( f_l \) and \( f_s \). The frequency range used in practice is 0-4000Hz. The average spectral distortion \( SD \) [10] is given by

\[ SD = \frac{1}{N} \sum_{i=1}^{N} SD_i \]  

(11)

The conditions for transparent speech from narrowband LPC parameter quantization are.

- The average spectral distortion (SD) must be less than or equal to 1dB.
- There must be no outlier frames having a spectral distortion greater than 4dB.
- The no of outlier frames between 2 to 4dB must be less than 2%.

V. RESULTS

Tables I, II, III, IV and V shows the spectral distortion (dB), computational complexity (k flops/frame), and memory requirements (ROM) at various bit rates for a 3- part split vector quantizer, 3-stage multistage vector quantizer, 2-switch, 3- part split vector quantizer, 2-stage, 2-switch, 3- part multi switched split vector quantizer and 3-stage, 2- switch, 3-part multi switched split vector quantizer. From tables 1 to 5 and from Fig’s 3 & 4 it is observed that 3-stage, 2-switch, 3-part MSSVQ has better spectral distortion performance, less computational complexity, and memory requirements when compared to 3-part SVQ, 3-stage MSVQ, and 2-switch, 3-part SSVQ. From Fig’s 5 & 6 it is observed that for 2-stage, 2-switch, 3-part MSSVQ, and 3-stage, 2-switch, 3-part MSSVQ as the number of stages increases the computational complexity, and memory requirements decreases as the number of bits/stage decreases from 24 to 16 frame of quantization and from 15 to 9 bits/frame the case is reversed. For 3-part SVQ transparency in quantization is achieved at 24 bits/frame, for 3-stage MSVQ transparency is achieved at 22 bits/frame, for 2-switch, 3-part SSVQ transparency is achieved at 22 bits/frame, and for 2-stage, 2-switch, 3-part MSSVQ transparency is achieved at 21 bits/frame. From the results it is proved that MSSVQ has better spectral distortion performance, less computational complexity and memory requirements when compared to all the above mentioned product code vector quantization techniques. From tables 6 to 8 gives the number of unstable frames for a given bit rate the instability is due to the independent quantization of the sub vectors in SVQ, SSVQ and MSSVQ and it can be observed that the number of unstable frames increases with the decrease in bitrate.
Fig. 3 Complexity for 3-part SVQ, 3-stage MSVQ, 2-switch 3-part SSVQ, and 3-stage 2-switch 3-part MSSVQ at various bit rates

Fig. 4 Memory requirements for 3-part SVQ, 3-stage MSVQ, 2-switch 3-part SSVQ, and 3-stage 2-switch 3-part MSSVQ at various bit rates

Fig. 5 Complexity for 2–stage 2-switch 3-part and 3–stage 2-switch 3-part MSSVQ at various bit rates

Fig. 6 Memory requirements for 2–stage 2-switch 3-part and 3–stage 2-switch 3-part MSSVQ at various bit rates

Fig. 7 Spectral Distortion Performance for 3-part SVQ, 3-stage MSVQ, 2-switch 3-part SSVQ, and 3-stage, 2-switch, 3-part MSSVQ at various bit rates
### TABLE I

<table>
<thead>
<tr>
<th>Bits / frame</th>
<th>SD(dB)</th>
<th>2-4 dB</th>
<th>&gt;4dB</th>
<th>ROM (klops/frame)</th>
<th>ROM (floats)</th>
</tr>
</thead>
<tbody>
<tr>
<td>24(8+8+8)</td>
<td>1.45</td>
<td>0.43</td>
<td>0</td>
<td>10.237</td>
<td>2560</td>
</tr>
<tr>
<td>23(7+8+8)</td>
<td>1.67</td>
<td>0.94</td>
<td>0</td>
<td>8.701</td>
<td>2176</td>
</tr>
<tr>
<td>22(7+7+8)</td>
<td>1.701</td>
<td>0.78</td>
<td>0.1</td>
<td>7.165</td>
<td>1792</td>
</tr>
<tr>
<td>21(7+7+7)</td>
<td>1.831</td>
<td>2.46</td>
<td>0.2</td>
<td>5.117</td>
<td>1280</td>
</tr>
</tbody>
</table>

### TABLE II

<table>
<thead>
<tr>
<th>Bits / frame</th>
<th>SD(dB)</th>
<th>2-4 dB</th>
<th>&gt;4dB</th>
<th>ROM (klops/frame)</th>
<th>ROM (floats)</th>
</tr>
</thead>
<tbody>
<tr>
<td>24(8+8+8)</td>
<td>0.984</td>
<td>1.38</td>
<td>0</td>
<td>30.717</td>
<td>7680</td>
</tr>
<tr>
<td>23(7+8+8)</td>
<td>1.238</td>
<td>1.2</td>
<td>0.1</td>
<td>25.597</td>
<td>6400</td>
</tr>
<tr>
<td>22(7+7+8)</td>
<td>1.345</td>
<td>0.85</td>
<td>0.13</td>
<td>20.477</td>
<td>5120</td>
</tr>
<tr>
<td>21(7+7+7)</td>
<td>1.4</td>
<td>1.08</td>
<td>0.3</td>
<td>15.357</td>
<td>3840</td>
</tr>
</tbody>
</table>

### TABLE III

<table>
<thead>
<tr>
<th>Bits / frame</th>
<th>SD(dB)</th>
<th>2-4 dB</th>
<th>&gt;4dB</th>
<th>ROM (klops/frame)</th>
<th>ROM (floats)</th>
</tr>
</thead>
<tbody>
<tr>
<td>24(12+12)</td>
<td>0.957</td>
<td>1.06</td>
<td>0</td>
<td>8.78</td>
<td>4372</td>
</tr>
<tr>
<td>23(11+12)</td>
<td>1.113</td>
<td>1.29</td>
<td>0.14</td>
<td>7.244</td>
<td>3604</td>
</tr>
<tr>
<td>22(11+11)</td>
<td>1.119</td>
<td>0.52</td>
<td>1.3</td>
<td>5.196</td>
<td>2580</td>
</tr>
<tr>
<td>21(10+11)</td>
<td>1.127</td>
<td>1.3</td>
<td>0.56</td>
<td>4.428</td>
<td>2196</td>
</tr>
</tbody>
</table>

### TABLE IV

<table>
<thead>
<tr>
<th>Bits / frame</th>
<th>SD(dB)</th>
<th>2-4 dB</th>
<th>&gt;4dB</th>
<th>ROM (klops/frame)</th>
<th>ROM (floats)</th>
</tr>
</thead>
<tbody>
<tr>
<td>24(12+12)</td>
<td>0.071</td>
<td>0</td>
<td>0</td>
<td>0.9</td>
<td>396</td>
</tr>
<tr>
<td>23(11+12)</td>
<td>0.083</td>
<td>0</td>
<td>0</td>
<td>0.836</td>
<td>364</td>
</tr>
<tr>
<td>22(7+7+8)</td>
<td>0.92</td>
<td>0</td>
<td>0</td>
<td>0.772</td>
<td>332</td>
</tr>
<tr>
<td>21(7+7+7)</td>
<td>0.13</td>
<td>0</td>
<td>0</td>
<td>0.708</td>
<td>300</td>
</tr>
</tbody>
</table>

### VI. CONCLUSION

MSSVQ provides better trade-off between bit rate and spectral distortion performance, computational complexity, and memory requirements, when compared to other product code vector quantization schemes like SVQ, MSVQ, and SSVQ. So MSSVQ is proved to be better. The decrease in the computational complexity is due to the less availability of bits at each stage of quantization as the number of stages increases. From Fig. 4 it can be observed that for SSVQ the memory required is high when compared to SVQ. This has been overcome by MSSVQ where the memory required is less when compared to SVQ, MSVQ, and SSVQ. So MSSVQ is
proved to be better when compared to all the above product code vector quantization techniques.

ACKNOWLEDGMENT

The authors place on record their grateful thanks to the authorities of R. V. R. & J. C. College of Engineering, A.P., India, K. L. College of Engineering, A. P., India, and Jawaharlal Nehru Technological University, College of Engineering, Hyderabad, India for providing the facilities.

REFERENCES