UNIVERSITÀ DEGLI STUDI DI PADOVA

FACOLTÀ DI INGEGNERIA

Department of Information Engineering - DEI

ICT (Internet and Multimedia)

Source Coding Course
Final Project

Implementation of DPCM Algorithm

Supervised By
Professor Giancarlo Calvagno

Prepared By
Mohammad kameli

Summer 2018

<u>Outline</u>

1- <u>Introduction</u>

    1-1 DPCM Algorithm

    1-2 Golomb Coding

2- <u>MATLAB® Implementation</u>

    2-1 DPCM Algorithm

    2-2 Quantization

    2-3 Golomb Coding

        2-3.1 Geometric distribution Parameter

        2-3.2 Mapping data

3- <u>Result</u>

    3-1 SNR

    3-2 Rate

1. Introduction

Presented project provides implementation of DPCM (Differential Pulse-Code Modulation) in MATLAB® which is followed by Golomb Coding to compress the CD-quality audio signals (16 bit/sample) in a lossy coding technique.

The goal of the reports is the following:

1- Developing an algorithm for encoding and decoding CD-quality audio signals (16 bit/sample) by means of the DPCM technique.
2- Using a linear predictor of order N = 1; 2; 4.
3- Using a Golomb code for coding the prediction error.
4- Computing the rate and SNR for different combinations of predictors and number of quantization levels using different kind of audio signals (voice, music of different genre).

1.1 *DPCM (Differential Pulse-Code Modulation)*

Differential Pulse-Code Modulation (DPCM) is a coding technique based on transmitting the weighted difference of previous transmitted samples instead of sending each sample independently, regarding the correlation of samples of data. Although it takes fewer bits to encode differences than it takes to encode the original samples, we have not said whether it is possible to recover an acceptable reproduction of the original sequence from the quantized difference value. When we were looking at lossless compression schemes, we found that if we encoded and transmitted the first value of a sequence, followed by the encoding of the differences between samples, we could losslessly recover the original sequence. [Figure.1] representing the encoder and decoder of DPCM algorithm respectively.

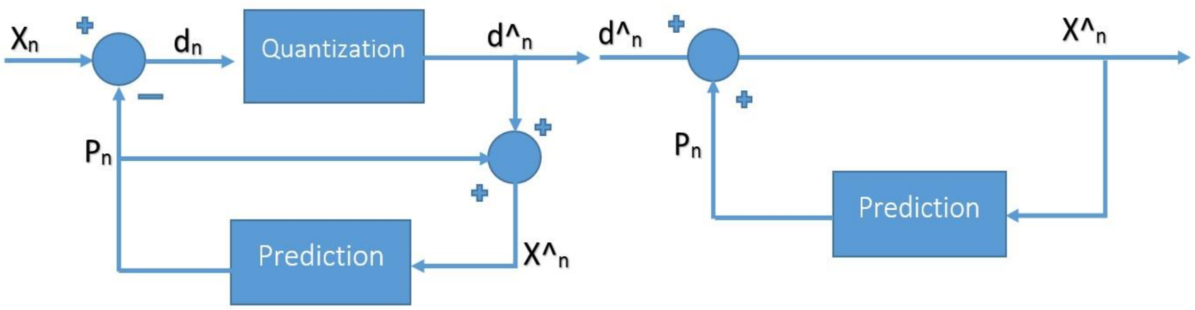


Figure. [1]: DPCM Encoder-Decoder

Moreover, the closed loop prediction is provided to avoid quantization error accumulation. Theoretically, if the quantization error process is zero mean, the errors will cancel each other out in the long run. In practice, often long before that can happen, the finite precision of the machines causes the reconstructed value to overflow.

Notice that the encoder and decoder are operating with different pieces of information. The encoder generates the difference sequence based on the original sample values, while the decoder adds back the quantized difference onto a distorted version of the original signal.

We can solve this problem by forcing both encoder and decoder to use the same information during the differencing and reconstruction operations. The only information available to the receiver about the sequence $\{x_n\}$ is the reconstructed sequence $\{ \hat{x}_n\}$. As this information is also available to the transmitter, we can modify the differencing operation to use the reconstructed value of the previous sample, instead of the previous sample itself.

1.2 *Golomb Coding*

The Golomb codes utilize to encode integers with the assumption that the larger an integer, the lower its probability of occurrence. The simplest code for this situation is the *unary* code. The unary code for a positive integer *n* is simply *n* 1s followed by a 0. Although the unary code is optimal in very restricted conditions, we can see that it is certainly very simple to implement. One step higher in complexity are a number of coding schemes that split the integer into two parts, representing one part with a unary code and the other part with a different code. We can consider the quantized audio output as two geometrics distribution, by finding the probability of the distribution we can introduce the “m” parameter in Golomb coding. Golomb code is optimal for the geometric probability model:

$$P(n) = p^{n-1} q, q = 1 - p \qquad (1)$$

When:

$$m = [\frac{-1}{log2(1-p)}] \qquad (2)$$

We compute the “m” parameter for different genre and different level of quantization, then compute the average number as the parameter of Golomb coding system.

2. MATLAB ® Implementation

Adapting this coding technique to variety of audio input, we tried to use different genre of music (Jazz, Pop, Classic, and Traditional) and voices (Man and Woman) to find the optimum Golomb division parameter.

2.1 DPCM Algorithm

First of all, we need to find the DPCM coefficients for different order of N = 1, 2 and 4. By calculating the autocorrelation function of x_n:

$$Rxx\,(k) = E[x_n\, x_{n+k}\,] \tag{3}$$

The coefficient would be the following:

$$a_N = R^{-1}p \tag{4}$$

Where we have used the fact that the autocorrelation matrix is symmetric. As mentioned before, to avoid accumulation of quantization error we would do the DPCM in a close loop to update and reconstruct the differences regarding the estimation in output.

2.2 *Quantization*

There are two different approaches to quantize the input signal. We can analysis the input signals to find an average quantization width for granular region regarding the density of the differences signals in baseband. Although the overflow error would increase during this idea. In this report we just consider a uniform Quantizer with different level of quantization M = 63, 127, 255 (odd levels to build a midtread Quantizer).

Regarding 16bits input audio signals, the range of input would be between -32768 and +32768 and for three different level of quantization the Δ parameters are:

$\Delta 1 = \frac{65536}{64}$ $\quad\quad$ $\Delta 2 = \frac{65536}{128}$ $\quad\quad$ $\Delta 4 = \frac{65536}{256}$

2.3 *Golomb Coding*

Adjusting division parameter for Golomb coding, needs an assumption which the quantized output have a Geometric distribution. For a narrow Gaussian distribution that would be an acceptable assumption.

2.3.1 *Geometric distribution parameter*

For the geometric distribution, the parameter "*p*" can be estimated by equating the expected value with the sample mean. This is the method of moments, which in this case happens to yield maximum likelihood estimates of "*p*". Let $x_1, ..., x_n$ be a sample where $x_i \geq 0$ for $i = 1, ..., l$ *(l = length of signal)*. Then "*p*" can be estimated as:

$$P\wedge = \frac{l}{\sum_{i=1}^{l} xi} \qquad (5)$$

In addition the “p” parameter of geometric distribution also introduce as the probability of zeroes, ***P(X = 0).*** We can count the number of zero values in the sequence ad divide by the length of sequence. Finally the “m” parameter of Golomb coding would be achieved by equation *(2).*

2.3.2 *Mapping data*

Golomb's scheme was designed to encode sequences of non-negative numbers. However it is easily extended to accept sequences containing negative numbers using an *overlap and interleave* scheme, in which all values are reassigned to some positive number in a unique and reversible way.

The n^{th} negative value (i.e., -n) is mapped to the n^{th} between the two positive intervals to achieve less entropy and could extracted uniquely in the decoder. Finally by normalizing the data by the quantization intervals we would achieve less entropy which the odd number (2n-1) represent the negative integer and the even numbers (2m) would represent the positive integer. This may be expressed mathematically as follows: a positive value *x* is normalized to:

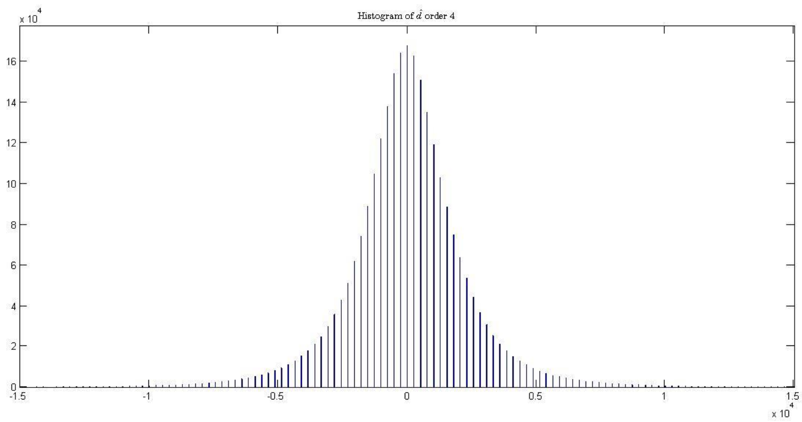
$$(x^{Even} = \frac{x+}{\Delta/2}, x>0) \qquad (6)$$

*A*nd a negative value *x* is shifted and normalized to:

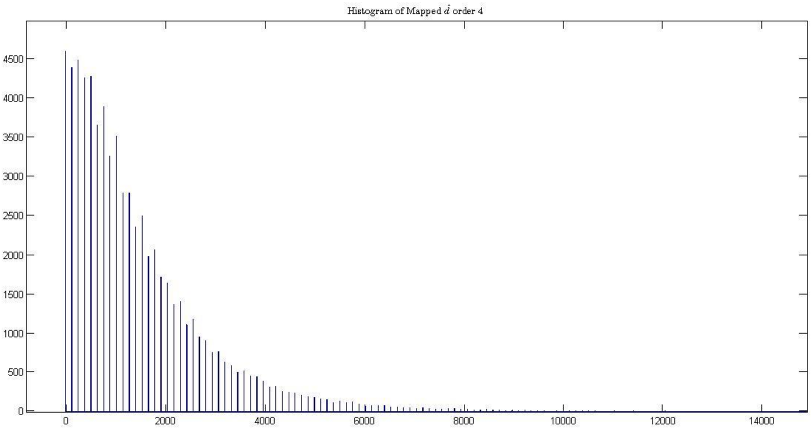
$$(x^{Odd} = \frac{-(x- +\Delta/2)}{\Delta/2}, x<0) \qquad (7)$$

Such a code may be used for simplicity, even if suboptimal. Truly optimal codes for two-sided geometric distributions include multiple variants of the Golomb code, depending on the distribution parameters, including this one.

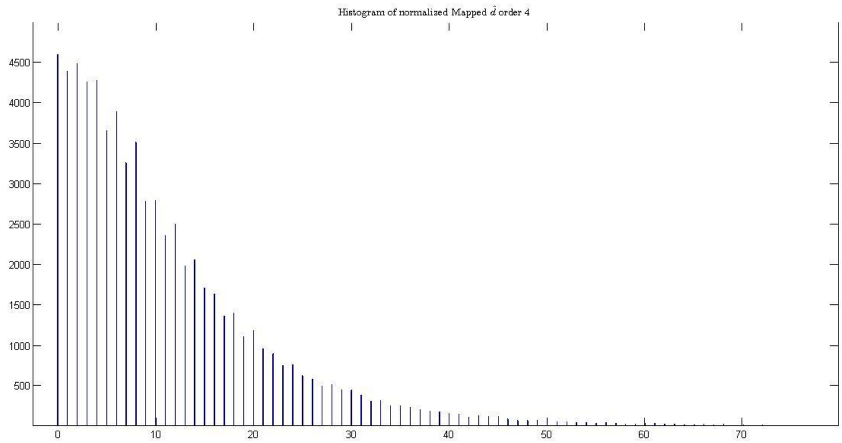
Figure.[2] represent the histogram of the data for both negative and positive integer for the POP music. Figure.[3] represent the mapped data to positive integer and figure.[4] represent the normalized mapped data of POP music.



[Figure.2]: Histogram of POP music samples



[Figure.3]: Histogram of mapped to positive integer of POP music samples



[Figure.4]: Histogram of normalized mapped to positive POP music samples

Regarding mapping the data, the entropy would decrease in a reversible approach which increase the Golomb coding technique property.

The “m” parameter calculated for different levels of quantization and different order of DCPM. The average of “m” parameter value is 3.66. We consider the **4** (which is power of 2) as the Golomb coding parameter for our system to adapt for different genre of music and voices.

3. Results

3.1 *SNR*

For the fixed-length quantizer the SNR is:

$$\text{SNR} \ \ (\text{dB}) = 10 \log_{10}\left(\frac{\sigma 2x}{\sigma 2q}\right) \qquad (6)$$

Where σ_x^2 is the variance of input signal and the σ_q^2 is the variance of error which is define as: $e = x - \hat{x} = d - \hat{d}$. The SNR values in dB are listed in tables [4] to [10] regarding different genre of music or different voices:

SNR [dB]	Order N = 1	Order N = 2	Order N = 4
63	57.31	57.31	57.31
127	61	60.98	60.93
255	72.5	72.4	72.4

[Figure.4]: SNR for **Classic Music** and different Level of Quantization

SNR [dB]	Order N = 1	Order N = 2	Order N = 4
63	71.5	71.5	71.5
127	75.26	75.23	75.24
255	86.7	86.7	86.8

[Figure.5]: SNR for **Jazz Music** and different Level of Quantization

SNR [dB]	Order N = 1	Order N = 2	Order N = 4
63	78.68	78.68	78.68
127	82.3	82.3	82.3
255	93.91	93.91	93.89

[Figure.6]: SNR for **POP Music** and different Level of Quantization

SNR [dB]	Order N = 1	Order N = 2	Order N = 4
63	79.57	79.56	79.57
127	83.25	83.25	83.26
255	94.80	94.80	94.80

[Figure.7]: SNR for **Traditional Music** and different Level of Quantization

SNR [dB]	Order N = 1	Order N = 2	Order N = 4
63	55.04	51.50	51.39
127	58.31	58.21	57.02
255	68.68	69.34	68.33

[Figure.8]: SNR for **Man Voice** and different Level of Quantization

SNR [dB]	Order N = 1	Order N = 2	Order N = 4
63	56.48	56.13	55.97
127	59.72	59.36	59.54
255	70.20	69.75	70.21

[Figure.9]: SNR for **Woman Voice** and different Level of Quantization

3.2 *Rate*

The output rate for different DPCM orders are listed below:

Rate [Bits/Sample]	Classic	POP	Jazz	Traditional	Man	Woman
63	3	3.76	3.11	3.35	3	3
127	3.06	4.89	3.46	4.39	3	3
255	3.31	7.22	4.28	5.48	3	3

[Figure.9]: Rate for N =1 (DPCM Order)

Rate [Bits/Sample]	Classic	POP	Jazz	Traditional	Man	Woman
63	3	3.77	3.11	3.35	3.02	3
127	3.02	4.92	3.48	4.03	3.1	3.08
255	3.13	7.28	4.33	5.47	3.38	3.37

[Figure.10]: Rate for N =2 (DPCM Order)

Rate [Bits/Sample]	Classic	POP	Jazz	Traditional	Man	Woman
63	3	3.67	3.11	3.36	3.02	3.01
127	3.02	4.71	3.48	4.05	3.11	3.06
255	3.14	6.86	4.34	5.5	3.37	3.28

[Figure.11]: Rate for N =4 (DPCM Order)

For extracted "m" parameter regarded each music and sound:

Rate [Bits/Sample]	Classic	POP	Jazz	Traditional	Man	Woman
63	2.17	3.74	3.11	3.35	2.19	2.17
127	2.40	4.37	3.46	4.39	2.47	2.43
255	2.98	6.29	4.28	5.48	3.12	3.28
m Parameter	2	6	4	4	2	2

Finally to evaluate the quality of output signals regarding input signals, we calculate the input-output distortion which is shows in table [12].

Distortion [dB]	Classic	POP	Jazz	Traditional	Man	Woman
64	113.5	113.78	113.78	113.78	92.5	109.2
128	99.8	99.92	99.91	99.91	81.5	96.32
256	85	86.05	86.05	86.04	68.05	83.98

[Figure.12]: Input-Output Distortion in dB