Republic of Iraq
AL-Nahrain University
College of Science

Effect of Resampling and Requantization on the Compression of Digital Audio Data

A Thesis
Submitted to the College of Science Al-Nahrain University In Partial Fulfillment of the Requirements for the Degree of Master of Science in Physics

BY
SHEREEN ABDUAL QADIR MAHDI AL-Samara' i
(B.Sc. 2002)

Supervisors

Dr. Laith A. Al –Ani | **Dr. Loay E. George**

April 2005 | **Rabee Alawal 1426**

Abstract

The study of the resampling and requantization methods of digital audio data is one of the major assets project. Which these methods used to compression the audio data.

In this search the application of some resampling methods on the audio signal was investigated by reducing the number of samples while the audio quality is maintained. The considered resampling methods are the

Linear, Quadratic, Cubic spline, Lagrange and Bezier

and for each method the level of sampling reduction was investigated by applying the down sampling rate using and then up sampling using the above mentioned interpolation method. The efficiency of each method under consideration will be determined with the aid of quality criteria like peak signal to noise ratio (PSNR). The Lagrange, Cubic spline, and Beizer interpolation methods provided have the same results and good quality.

Also in this search the results of applying the **uniform** and **non–uniform** quantization methods are presented the effect of the quantization steps on the audio quality investigated. The results proved the uniform quantization method is better than non–uniform quantization method.

A listening test was used to prove the efficiency of each method, the test sample has different backgrounds and they prove when the decimation rate and the step of quantization increase the audio quality will be decrease

Contents

Chapter Four: Experimental Results

Chapter Five: Conclusions and Future Work

List of Abbreviations

ADC	Analog to Digital Converter
CD	Compact Disk
dB	Decibel
DAC	Digital to Analog Converter
Hz	Hertz
MSE	Mean Square Error
PAM	Pulse Amplitude Modulation
PCM	Pulse Code Modulation
PSNR	Peak Signal to Noise Ratio
RIFF	Resource Interchange File Format

<u>List of Figures</u>

List of Tables

Acknowledgement

First, I thank my God for given me health to complete this work, also I would like to express my deep appreciation and sincere of gratitude to my supervisors Dr. Laith A. Al-Ani and Dr. Loay E. George for suggesting this project, valuable guidance, and encouragement during the development of this project.

I wish to thank the head of Physics department Dr. Ahmed K. Ahmed and the staff for their kind attention.

My deep thanks to my father, mother, sisters ′′Sheamaa, Beadaa, Sarah′′, and my brother ′′Khalid′′.

My deep appreciation to(Mr. Mohammed Sahib, Mr. Husam, Ammar Abdual Sattar, Abduallah, Israa, Etab, and Wesam) for their assistance during to preparing of this thesis.

Shereen

Appendix A

The Wave File Format

The header structure can be represented by the following record structure:

Field name	Field size (bytes)
Sign	4
WavSiz	٤
Format	٤
Block type	٤
Sound card	٤
File format type	٢
No. of channels	٢
Sampling rate	٤
Byte rate	٤
Byte per sample	٢
Bits per sample	٢
Chunk name	٤
Chunk size	٤

The WAV format stars with the RIFF header:

Name	Size	Description
Sign	4	Contains the letters’’RIFF’’ in ASCII form
WavSiz	4	This is the size of the entire file in bytes minus 8 bytes for the two fields not included in this count: sign and wavsiz
Format	4	Contains the letters ’’WAVE’’

The WAV format consists from ’’fmt’’ and ’’data’’
The ’’fmt’’ describes the sound data format:

Name	Size	Description
Block type	4	Contain the letters ’’fmt’’
Sound card	4	This is the type of the used sound card during the recording stage
File format type	2	PCM = 1
No. of channels	2	Mono = 1, Stereo = 2
Sampling rate	4	8000, 11024, 22048, 44069
Byte rate	4	No. of bits per second
Byte per sample	2	No. of bytes per samples
Bits per sample	2	8 bits = 8, 16 bits = 16
Chunk name	4	Contains the letters ’’data’’
Chunk size	4	This is the number of bytes in data

Supervisor's Certification

We certify that this thesis was prepared under our supervision at the 'AL-Nahrain University' as a partial requirements for the degree of Master of Science in Physics.

Signature:		Signature:	
Name:	*Dr. Laith A. Al-Ani*	Name:	*Dr. Loay E. George*
Title:	*Assist Professor*	Title:	*Assist Professor*
Address:	*College of Science AL-Nahrain University.*	Address:	*Ministry of Higher Education*
Date:	*/ / 2005*	Date:	*/ / 2005*

In the view of the recommendation. I forward this thesis for debate by the examination committee.

Signature:
Name: *Dr. Ahmad K. Ahmad.*
Title: *Assist Professor*
Address: *Head of the Department of Physics*
College of Science
AL-Nahrain University.

Date: */ / 2005*

Chapter Two
Digital Audio Processing

2.1 Introduction

A signal can be defined as function that conveys information. Although signals can be represented in many ways, in all cases the information is contained in a pattern of variations of some form, for example the signal may take the form of pattern of time variations or a spatially varying pattern. Signals are represented mathematically as function of one or more independent variables. For example, a speech signal would be represented as a sampling and quantization [**Opp 75**].

When we hear a voice of a friend or other well know one we recognize it instantly. Similarly if we hear music, we can recognize the sound of particular musical instrument. Some people are even able to recognize the identity of an instrument by the sound alone. So it is clear that the sound of these people and instrument must be different. There are plenty of terms to describe the tone of someone's voice: rich, reedy, discordant, syrupy, and seductive. Musicians have their adjective, but these are poetic rather than precise. Fortunately for the engineer, physicists and mathematicians have provided a precise way of characterizing any sound whenever or however it is produced [**Rab 78**].

Continuous–time, continuous–amplitude signals are sometimes called analog signals. Signal processing systems may be classified along the same lines as signals. That is, continuous–time systems are systems for which both the input and output are continuous–time signals and discrete–time. Systems are those for which the input and output are discrete–time signals. Similarly analog systems are systems for which the input and output are analog signals and digital systems are those for which the input and output are digital signals. Digital signals processing, deals with transformations of signals that are discrete in both amplitude and time [**Opp 75**].

The primary element of a wave is its strength or amplitude, the amplitude is determined by the highest point along the curve of the sound wave, the higher the amplitude, the louder the sound will be. The physical unit of loudness is the decibel (dB), a decibel is algorithmic unit of measuring specifying the degree of loudness of the wave. Varying the amplitude of its wave changes the loudness of a sound. The second element of a wave is its frequency. How high or low a given tone sounds depends on the number of pulses per second. This number of pulses is referred to as the tone's frequency [**Emb 91**].

2.2 The Physics of Sound

For most of us sound is a very familiar phenomenon, since we hear it all the time. Nevertheless, when we try to define sound, we find that we can approach this concept from two different points of view, and we end up with two definitions, as follows [**Sal 98**]:

1. An intuitive definition: sound is the sensation detected by our ears and interpreted by our brain in a certain way.

2. A scientific definition: sound is a physical disturbance in a medium propagated as a pressure wave by the movement of atoms or molecules.

When we speak the sound that we make creates a series of compression and expansion in the air around us. However, for a sound to travel from the sound source to ear, another element must be available to transmit the sound. This "sound carrier" is called a medium. Usually this medium is the air that surrounds us. However, sound can also travel through water. Without a medium, sound transmission is not possible, for example, it's impossible to have a conversation on the moon. Since the moon lacks an atmosphere, a medium is not present to carry the sound from one mouth to the listener's ear [**Sto 93**].

We normally hear sound as it propagates through the air and hits the diaphragm in our ear. However, sound can propagate in many different media. Marine animals produce sounds under water and respond to similar sound. Hitting the end of metal bar with a hammer produces sound waves that propagate through the bar and can be detected at the other end. Good sound insulators are rare, and the best insulator is vacuum, where there are no particles to vibrate and propagate the disturbance. Sound can also be considered as a wave, everthough its frequency may change all the time. It is a longitudinal wave, one where the disturbance is in the direction of the wave itself. In contrast, electromagnetic waves and ocean waves are transverse waves. Their oscillations are perpendicular to the direction of the wave. As any other wave, sound has three important attributes, its speed, amplitude, and period. The frequency of a wave is not an independent attribute; it is the number of periods that occur in one time unit (one second). The unit of frequency is the hertz (Hz). The speed of sound depends mostly on the medium it passes through, and on the temperature [**Sal 98**].

2.3 Digital Wave File

Wave audio files are one of the common formats used to store and play audio data. They support variable sampling frequencies, multiple channels, and a number of compression algorithms [**Wil 03**].

The wave file can be classified according to the number of sampling channels, and the samples resolution. Figure (2.1) presents the four types of PCM wave files.

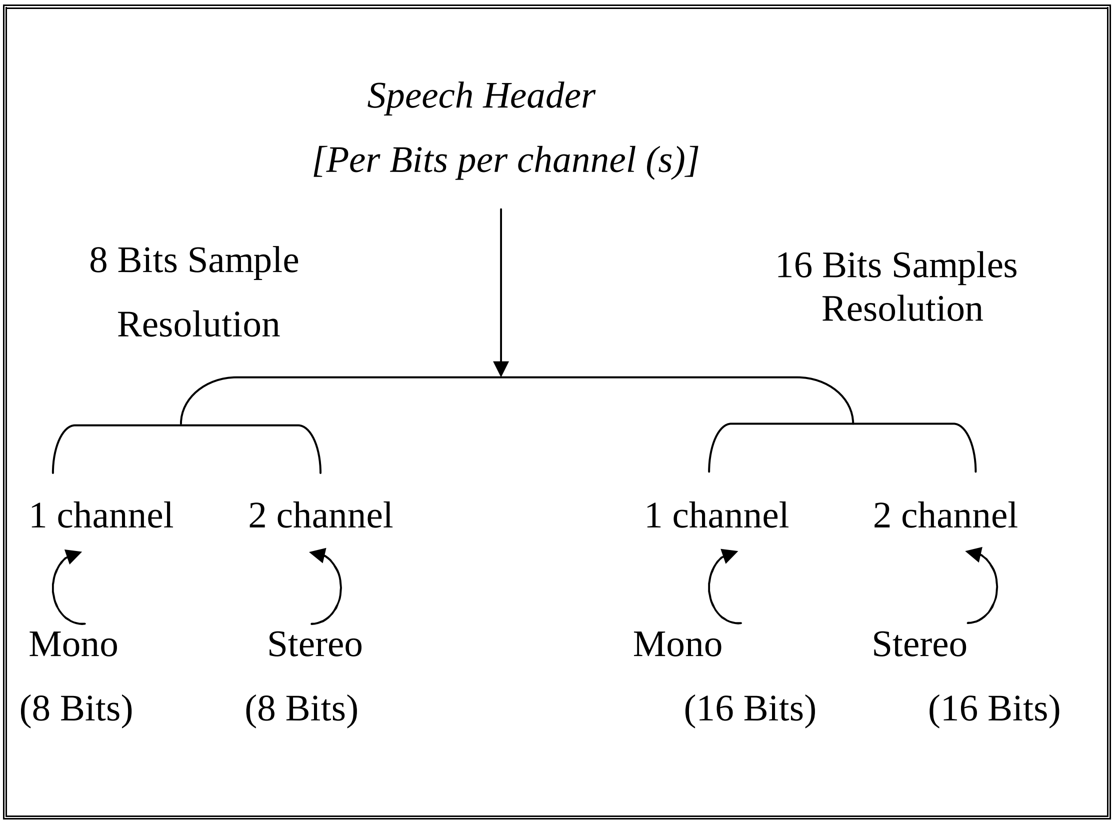


Figure (2.1) Types of PCM wave files

The structure of the wave file can be divided into two parts ***Header*** and ***Data Chunk***.

Header**:** contain thirteen field of information concerned with chunk data, it has length 44 bytes.

Data Chunk: contain a data of speech file, stored in binary format after the conversion from analog to digital form. Its length in byte depends on the recording time.

44 byte	(No. of samples × No. of channel × Sample Resolution / s) byte
Header	*Chunk*

Figure (2.2) Wave file structure

The contents of the wave header structure are:

1. The Signature Resource Interchange File Format (RIFF):

RIFF is a file format for storing many kinds of data, primarily multimedia data like audio and video. It is based on chunks; each chunk has a type, represented by a four-character tag. This chunk type comes first in the file, followed by the size of the chunk, then the contents of the chunk [**Web 03**].

2. The File Size:

It is a long integer number indicates the size of remainder of the file in bytes. It is equal to the length of the entire file -8 byte [**Web 03**].

3. The RIFF Type:

Multimedia applications require the storage and management of a wide variety of data, including bitmaps, audio data, and video data. RIFF provides a way to store all these varied type of data [**Wil 03**].

4. The Block Type:

It is a string type field tell us the kind of the followed chunk (mostly it is a format chunk which implies information about the speech data format.

5. Sound Card:

It is long integer field indicates the type of the used sound card during the recording stage.

6. File Format Type:

It is an integer field indicates the type of coding used to represent the speech wave from data, (the value 1 means Pulse Code Modulation).

7. No. of Channel(s):

It is an integer field indicates the number of recording channels. If it is equal to (1) it means Mono (single) channel otherwise if it is equal to (2) it means stereo (double) channels.

8. Sampling Rate:

It is a long integer field indicates the number of sampling per second, it may be one of the following values [8000, 11024, 22050, 44069] sample per second.

9. Bytes Rate:

It is a long integer field represents the number of bytes needed to store one sample.

10. Chunk Name:

It is a string (4 characters) type field indicates the next chunk type. In most cases it will be a "data" chunk.

11. Chunk Size:

It is a long type field indicates the size of data chunk.

2.4 Digital Audio

Much as an image can be digitized and broken up into pixels, where each pixel is a number, sound can also be digitized and broken up into numbers. When sound is played through a microphone, it is converted into a voltage that varies continuously with time. Such voltage is the analog representation of the sound. Digitizing sound is done by measuring the voltage at many points in time, translating each measurement into a number, and writing the numbers in a file. This process is called sampling. The sound wave is sampled, and the samples become the digitized sound. The device used for sampling is called **Analog–to–Digital Converter** (ADC) [**Sal 98**].

Since the sound samples are numbers, they are easy to edit. However, the main use of a sound file is to play it back. This is done by converting the numeric samples back into voltages that are continuously fed to a speaker. The device that does is called a **Digital–to–Analog Converter** (DAC) [**Sal 98**].

Just as it is possible to convert a sound between pressure wave in air and analog electric signal, it is possible to convert a varying electric signal into a series of digital values, and vice versa. However, because analog and digital sounds are fundamentally different, we always loose information when we make this transformation [**Kie 98**].

There are two factors that determine fidelity of the original analog signal: the sampling rate and the resolution of the sample.

1. Sampling rate is the number of samples that are used to represent one second of sound. By sampling at lower rates we don't lose the sound entirely, just the higher frequencies.

2. The resolution of the sample is the number of bits per sample. It may be 8-bit samples or 16–bit samples. 8–bit samples cannot accurately represent sound. The human brain, by way of its audio sensor (ears) can distinguish very subtle differences in amplitude and frequency. With only 256 recordable levels, many of the subtler elements of a complex sound disappear. On the other hand, 16-bit sampling can differentiate over 65,000 signal levels, which makes it possible to represent a sound with much greater fidelity, while only doubling the storage demand [**Sco 95**]

2.4.1 Pulse Code Modulation (PCM)

When an analog signal is converted to digital form, it is made discrete both in time and in amplitude. Discretization in time is the operation of sampling, while in amplitude it is quantizing. It is worth pointing out that the transmission of analog information by digital means is called (PCM) standing for "**Pulse Code Modulation**".

PCM is the first method used in converting analog speech signal to digital forms, and is still widely used in digital speech transmission systems

In PCM, the input speech signal is frequency bounded to exclude any frequency greater than a maximum frequency of the signal f_{max}. This signal is sampled at $f_s \geq 2f_{max}$ sample per second (sampling frequency), to produce the corresponding Pulse Amplitude Modulation (PAM) signal. The produced samples are quantized into the nearest m levels, and the number of bits in the sampling is $P = \log_2(m)$

It is simple to show that a binary codeword of m bits long allows 2^m separate numbers (or single values) to be represented. Thus, if m = 8, we may encode $2^8 = 256$ discrete values, if m = 16 then $2^{16} = 65536$ values may encode [**Wit 82**].

2.4.2 Sampling

The effects of time sampling in both time and frequency domains will first be investigated. We will find that provided the appropriate sampling criterion is satisfied, a continuous-time signal can in principle be exactly reconstructed from its samples without error [**Cav 00**].

The primary objective of our presentation is the understanding of the sampling theorem, which states that when the sampling rate is greater than twice the highest frequency contained in the spectrum of the analog signal, the original signal can be reconstructed exactly from the samples [**Mcc 98**].

The plots shown in figure (2.3) naturally raise the question of how frequently we must sample in order to retain enough information to reconstruct the original continuous–time signal from its samples. The amazingly simple answer is given by ***Shannon sampling theorem*** which states that a continuous–time signal x(t) with frequencies no higher than f_{max} can be reconstructed exactly from its samples $x[n]=x(nT_s)$, if the samples are taken at a rate $f_s=1/T_s$ that is greater than $2f_{max}$. Where, n take only integer values, x[n]: reconstructed signal, T_s: sampling period, and f_s: sampling frequency.

This is a statement of the ***Shannon sampling theorem***, one of the theoretical pillars in modern digital communications, digital control, and digital processing. Notice that the sampling theorem involves two issues. First, it talks about reconstruction of the signal from its samples, although it never specifies the algorithm for reconstruction. Second it gives a minimum sampling rate that depends on the frequency content of x(t), the continuous-time signal. This minimum sampling rate is called the Nyquist rate [**Mcc 98**].

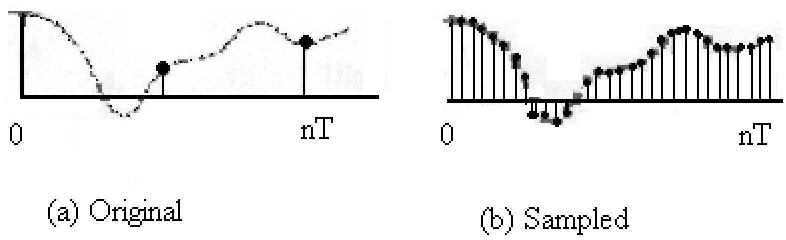


Figure (2.3) Original and its sampled signals

2.4.3 Quantization

It is the step which allows a continuous amplitude signal to be represented in terms of discrete amplitude increments.

The simplest form of quantization is the uniform quantization, where the amplitude range is splitted into equal regions by levels termed quantization level.

Quantization typically effects a distortion which depends on the chosen quantization step size and the number of quantization level [**Con 00**].

The quantization can be arranged in either a uniform fashion, i.e., uniformly distributed from the highest expected value to the lowest expected value, or non–uniformly distributed. Uniform quantizers allow the designer to designate a minimum value for the error of any quantized value. For uniform quantization there are only two parameters: the number of levels and the quantization step size, while non–uniform quantizers can give a significant increase in accuracy, especially when the statistics of the incoming signal are known.

In the present work 8–bits ADC conversion was used, to give 256 quantization levels, and half the levels correspond to negative input voltage, while the other half to positive one [**Dou 87**].

Chapter Five
Conclusions and Future Work

This chapter is dedicated to present a list of conclusions, which derived from the analysis results discussed in chapter four; also some suggestions for future work will be given.

5.1 Conclusions

From the analysis of the test results the following remarks were derived:

1. In the decimation method the sound is rapid because the number of samples per second is reduced.
2. Lagrange, Cubic spline, and Bezier interpolation methods have smaller error than Linear, and Quadratic.
3. The increase of the decimation rate will decrease the quality of the reconstructed signal.
4. When rate of down sample is ten the Peak Signal to Noise Ratio of the Cubic, Lagrange, and Bezier interpolation methods is between 24 to 29 dB more efficient than the Linear, and Quadratic methods.
5. In the quantization the sound is low because the amplitude is reduced.
6. The increase of the quantization step in the uniform quantization will decrease the quality of the reconstructed signal.
7. When the quantization step is 14 the Peak Signal to Noise Ratio of uniform quantization is 37 dB.
8. When the number of level is 4 the Peak Signal to Noise Ratio of non–uniform quantization is between 15 to 16 dB.

9. In non–uniform quantization when the number of the level is decrease the quality of the reconstructed signal will be also decrease.

5.2 Future Work

There are many directions in which the current research work could be developed. Among these directions are the following:

1. By using filters we may increase the down sampling rate.
2. Using other methods of interpolation like (Legender centered function, and cubic β–spline).
3. Apply other algorithms of non–uniform quantization.

Chapter Four
Experimental Results

4.1 Introduction

This chapter is dedicated to describe the application of some resampling methods on the audio signal by reducing the number of samples while the audio quality is maintained. The considered resampling methods are the Linear, Quadratic, Cubic spline, Lagrange and Bezier, and for each method the level of sampling reduction was investigated by applying the down sampling using and then up sampling using the above mentioned interpolation method. Also in this chapter the results of applying the uniform and non–uniform quantization methods to determine the effect of the quantization steps on the audio quality investigated

4.2 Resampling Processes

The signal can be reconstructed for all time from its samples by resampling process. We do this by using the interpolation methods, ***Linear*** interpolation which is the simplest method and it can be used to calculate any number of new samples between two existing samples. There are many methods for interpolating discrete points, for example, ***Lagrange interpolation*** is a classical technique of finding an order N polynomial which passes through N+1 given points.

Cubic splines fits a third order polynomial passing through two points. This allows for a smooth chain of third order polynomial passing through a set of points.

Also, ***Bezier*** interpolation method could be used to interpolate a set of points using smooth curves which don't necessarily pass through the points.

Since Shannon's sampling theorem says it is possible to restore an audio signal exactly from its samples, it makes sense that the best digital audio interpolators would be based on that theory. The block diagram shown in figure (4.1) illustrate the steps of implementing the interpolation methods (Linear, Quadratic, Cubic spline, Lagrange, and Bezier) as resampling methods.

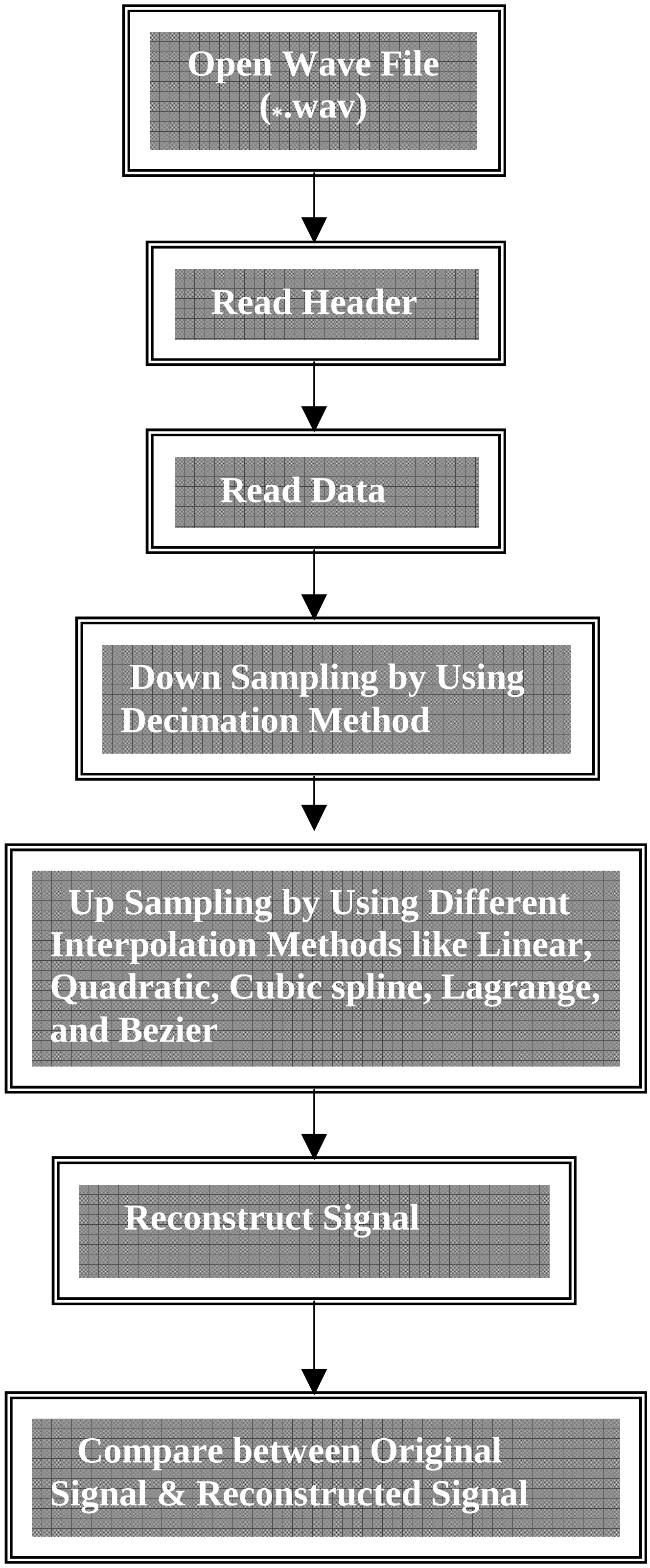


Figure (4.1) Block diagram of resampling process

4.2.1 Linear Interpolation Method

This method is the simplest methods of the interpolation. It is used to interpolate the samples. More details about the mathematical foundation of this model are discussed in chapter three. In this method the interpolate sample depend on the values of the two surrounding samples. Thus, since the samples are averaged. The results are obtained from the equation.

$$P(x) = f(x_i) + \frac{x - x_i}{x_{i+1} - x_i}[f(x_{i+1}) - f(x_i)], \qquad \ldots (4.1)$$

So, from the surrounding samples pair (x_i, x_{i+1}) we can determine the value of $P(x)$ at the point (x) within the interval $[x_i, x_{i+1}]$. Equation (4.1) would be rewritten in the form

$$Yup(x) = (1 - x) \times F_0 + x \times F_1, \qquad \ldots (4.2)$$

Where Yup is interpolated (up sampled) data, x is the normalized relative position of the interpolated samples: $x = (x - x_{i-1}) / (x_i - x_{i-1})$, F_0 ($f(x_i)$), and F_1 ($f(x_{i+1})$) are the nearest known samples.

Algorithm (4.1): A program for resampling by using Linear interpolation method.

```
Inputs: (1) Nosamp= No. of input samples
        (2) u = Ratio of up sampling
        (3) yup() = Samples after decimation
        (4) M = Up sampling rate -1
Out put: (1) Y() = Reconstructed samples
    For I = 0, 1, . . . , Nosamp
      F0 = Ydwn (I): F1 = Ydwn (I + 1)
    For j = 1 To M
      Y() = (1 - U) * F0 + U * F1
```

4.2.2 Quadratic Interpolation Method

In the previous section we have discussed the linear interpolation as a method based on evaluating straight line to interpolate the gaps between two points (known samples). Since the result of this simple interpolation method is often less than satisfactory for up sampling audio data, it is important to utilize other kinds of interpolants utilize higher order polynomials which can represent the curves more accurately. The simplest way of doing this is to apply the quadrant interpolant, which requires only three points to reconstruct an arc passing through these three points.

Let us consider the three points (x_0,Y_0), (x_1,Y_1), (x_2,Y_2) then since the quadratic interpolation formula is written as:

$$Yup\,(x) = a_0 + a_1\,x + a_2\,x^2, \qquad \ldots (4.3)$$

Where Yup is interpolated (up sampled) data, x is the normalized relative position of the interpolated samples.

By substituting the relative position values (x_0 = -1; x_1 = 0; x_2 = 1), in equation (4.25), we will get:

$$Y_0 = a_0 - a_1 + a_2, \qquad \ldots (4.4)$$

$$Y_1 = a_0, \qquad \ldots (4.5)$$

$$Y_2 = a_0 + a_1 + a_2, \qquad \ldots (4.6)$$

The solution of above three linear simultaneous equation leads to the following

$$a_1 = \frac{1}{2}(Y_2 - Y_0), \qquad \ldots (4.7)$$

$$a_2 = \frac{1}{2}(Y_2 + Y_0 - 2\,Y_1), \qquad \ldots (4.8)$$

So, substituting the values of Y_0, Y_1, Y_2 in equation (4.5), (4.7), and (4.8) we can get the values of (a_0, a_1, a_3) respectively. Then substituting the determine values of (a_0, a_1, a_3) in equation (4.3) we can get the value of Yup (x) at the relative position (x).

Algorithm (4.2): A program for resampling by using Quadratic interpolation method.

Inputs: (1) Nosamp= No. of input samples

(2) u = Ratio of up sampling

(3) yup() = Samples after decimation

(4) M = Up sampling rate -1

Out put: (1) Y() = Reconstructed samples

For I = 0, 1, . . . , Nosamp

Evaluated the coefficient a_0, a_1 and a_2 from the Quadratic eqaution

For j = 1 to M

$Y() = U * (a_0 * U + a_1) + a_2$

4.2.3 Cubic Polynomial Interpolation Method

Cubic spline is the name of an interpolation method. The weight coefficient for the four surrounding points, two to the left and two to the right of the point intended to be sampled.

Let us consider the four points then the cubic interpolation formula is written as:

$$Yup(x) = a_0 + a_1 x + a_2 x^2 + a_3 x^3, \quad \ldots (4.9)$$

Where Yup is interpolated (up sampled) data, and x is the normalized relative position of the interpolated samples. We assume x takes the values (-1, 0, 1, and 2), substitute these values in equation (4.9), we will get

$$Y_0 = a_0 - a_1 + a_2 - a_3, \quad \ldots (4.10)$$

$Y_1 = a_0,$ . . . (4.11)

$Y_2 = a_0 + a_1 + a_2 + a_3,$ . . . (4.12)

$Y_3 = a_0 + 2\ a_1 + 4\ a_2 + 8\ a_3,$ . . . (4.13)

A straight forward manipulation for the above four linear unknown equations, we get:

$$a_2 = \frac{1}{2}(Y_0 - 2\,Y_1 + Y_2)\,, \quad ...\ (4.14)$$

$$a1 = \frac{1}{3}(4\,b_1 - b_2)\,, \quad ...\ (4.15)$$

$$a3 = \frac{1}{3}(b_2 - b_1)\,, \quad ...\ (4.16)$$

Where

$$b_1 = \frac{1}{2}(Y_2 - Y_0)\,, \quad ...\ (4.17)$$

$$b_2 = \frac{1}{2}(Y_3 - 2\,Y_2 + 3\,Y_1 - 2\,Y_2)\,, \quad ...\ (4.18)$$

We substituted equation (4.11), (4.14), (4.15), and (4.16) in equation (4.9) to determine Yup.

Algorithm (4.3): A program of resampling by using Cubic spline interpolation method.

Inputs: (1) Nosamp= No. of input samples

(2) u = Ratio of up sampling

(3) yup() = Samples after decimation

(4) M = Up sampling rate -1

Out put: (1) Y() = Reconstructed samples

For I = 0, 1, . . . , Nosamp

Evaluated the coefficient a_0, a_1, a_2 , and a_3 from the Cubic eqaution

For j = 1 to M

$Y() = (U^* U * (a_0 * U + a_1) + a_2 * U + a_3)$

4.2.4 Lagrange Interpolation Method

A polynomial function is continuous and smooth everywhere. It would seem that if we can constructed a polynomial whose curve pass through the N+1 data points, our problem may be solved. For example, the Lagrange polynomial is the unique polynomial of degree N passing through these N+1 points. This polynomial interpolant can be thought of as an approximation of some other function passing through these N+1 points. Therefore better results are obtained from the approximation polynomial written in the form

$$P(x) = \sum_{i=0}^{n} I_i(\mathrm{x})\, f_i, \qquad \text{. . . (4.19)}$$

In out work the up sampling by using Lagrange interpolation method is done by taking pieces of four known samples surrounding the point to be samples. Let us consider four points then the Lagragian interpolation is begin

$$\mathrm{P}(\mathrm{x}) = \mathrm{I}_0(\mathrm{x})\, f_0 + \mathrm{I}_1(\mathrm{x})\, f_1 + \mathrm{I}_2(\mathrm{x})\, f_2 + \mathrm{I}_3(\mathrm{x})\, f_3, \qquad \text{. . . (4.20)}$$

Where

$$\mathrm{I}_0 = \frac{(x - \mathrm{x}_1)(\mathrm{x} - \mathrm{x}_2)(\mathrm{x} - \mathrm{x}_3)}{(x_0 - \mathrm{x}_1)(\mathrm{x}_0 - \mathrm{x}_2)(\mathrm{x}_0 - \mathrm{x}_3)}, \qquad \text{. . . (4.21)}$$

$$I_1 = \frac{(x - \mathrm{x}_0)(\mathrm{x} - \mathrm{x}_2)(\mathrm{x} - \mathrm{x}_3)}{(x_1 - \mathrm{x}_0)(\mathrm{x}_1 - \mathrm{x}_2)(\mathrm{x}_1 - \mathrm{x}_3)}, \qquad \text{. . . (4.22)}$$

$$I_2 = \frac{(x - \mathrm{x}_0)(\mathrm{x} - \mathrm{x}_1)(\mathrm{x} - \mathrm{x}_3)}{(x_2 - \mathrm{x}_0)(\mathrm{x}_2 - \mathrm{x}_1)(\mathrm{x}_2 - \mathrm{x}_3)}, \qquad \text{. . . (4.23)}$$

$$I_3 = \frac{(x - \mathrm{x}_0)(\mathrm{x} - \mathrm{x}_1)(\mathrm{x} - \mathrm{x}_2)}{(x_3 - \mathrm{x}_0)(\mathrm{x}_3 - \mathrm{x}_1)(\mathrm{x}_3 - \mathrm{x}_2)}, \qquad \text{. . . (4.24)}$$

Since the points (x_0,x_1,x_2,x_3) are equally spaces, then their relative position could be set (-1, 0, 1, and 2) respectively then equation (4.20) become

$$Yup\,(\mathrm{x}) = F_0 \times \mathrm{I}_0 + F_1 \times \mathrm{I}_1 + F_2 \times \mathrm{I}_2 + F_3 \times \mathrm{I}_3\,, \qquad ...\,(4.25)$$

Where Yup is interpolated (up sampled) data, F_0, F_1, F_2, and F_3 are the known surrounding samples.

We substitute the relative values of (x_0,x_1,x_2,x_3) in equations (4.21), (4.22), (4.23), and (4.24) to find I_0, I_1, I_2, and I_3 respectively:

$$I_0 = \frac{(\mathrm{x - 0})\,(\mathrm{x - 1})\,(\mathrm{x - 2})}{(0 - 1)\,(0 - 2)\,(0 - 3)} = -\frac{1}{6}\,\mathrm{x}\,(\mathrm{x - 1})\,(\mathrm{x - 2})\,, \qquad ...\,(4.26)$$

$$I_1 = \frac{(\mathrm{x} - (-1))\;\;(\mathrm{x - 1})\,(\mathrm{x - 2})}{(1 - 0)\,(1 - 2)\,(1 - 3)} = \frac{1}{2}\,(\mathrm{x + 1})\,(\mathrm{x - 1})\,(\mathrm{x - 2})\,, \qquad ...\,(4.27)$$

$$I_2 = \frac{(\mathrm{x} - (-1)\,(\mathrm{x - 0})\,(\mathrm{x - 2})}{(2 - 0)\,(2 - 1)\,(2 - 3)} = -\frac{1}{2}\,\mathrm{x}\,(\mathrm{x + 1})\,(\mathrm{x - 2})\,, \qquad ...\,(4.28)$$

$$I_3 = \frac{(\mathrm{x} - (-1))\,(\mathrm{x - 0})\,(\mathrm{x - 1})}{(3 - 0)\,(3 - 1)\,(3 - 2)} = \frac{1}{6}\,\mathrm{x}\,(\mathrm{x + 1})\,(\mathrm{x - 1})\,, \qquad ...\,(4.29)$$

Then we substitute equation (4.26), (4.27), (4.28), and (4.29) respectively in equation (4.25) to determine Yup.

Algorithm (4.4): A program for resampling by using Lagrange interpolation method.

Inputs: (1) Nosamp= No. of input samples

(2) u = Ratio of up sampling

(3) yup() = Samples after decimation

(4) M = Up sampling rate -1

Out put: (1) Y() = Reconstructed samples

Continue

```
For j = 1 to M
   Um1 = U - 1: Um2 = U - 2: Up1 = U + 1
   F (0, j) = -U * Um1 * Um2 / 6: F (1, j) = Up1 * Um1 * Um2 / 2
   F (2, j) = -U * Up1 * Um2 / 2: F (3, j) = U * Up1 * Um1 / 6
 Y() =  F((0, j) * Yp(0) + F(1, j) * Yp(1) + F(2, j) * Yp(2) + F(3, j) *
 Yp(3))
```

4.2.5 Bezier Interpolation Method

This method is one of the simplest methods for representing the curves. The mathematical relationship can be found in chapter three. The form of the Bezier functions can be given as:

$$P\,(\mathrm{x}) = \sum_{i=0}^{N} P_i\, W_i(N, i, \mu)\,, \qquad \ldots (4.30)$$

Where W is called Bernisten blending function and given by the relation

$$W(N, i, \mu) = [{}^{N}C_i]\, \mu^i\, (1-\mu)^N\,, \qquad \ldots (4.31)$$

$$[{}^{N}\mathrm{C_i}] = \mathrm{N!} / (\mathrm{i!}\ (\mathrm{N} - \mathrm{i})!), \qquad \ldots (4.32)$$

Where P_i is the parametric point ($P_i = P_0, P_1, \ldots, P_n$), μ^i is the value selected in the range [0, . . . , 1], N is the number of control points.

In out work the interpolation was done by choose the surrounding four points around the position that intended to be up sampled. We consider the relative position of the four points (0, 1, 2, and 3) then equation (4.30) become

$$P = \mathrm{P}_0\,(1-\mu)^3 + 3\,\mathrm{P}_1\,\mu\,(1-\mu)^2 + 3\,\mathrm{P}_2\,\mu^2\,(1-\mu) + \mathrm{P}_3\,\mu^3\,, \qquad \ldots (4.33)$$

Applying equation (4.33) on the four points (whose μ values are 0, $\frac{1}{3}, \frac{2}{3}, \textit{and}\ 1$ we will get:

$P_0 = y_0,$ . . . (4.34)

$P_3 = y_3,$ . . . (4.35)

$$y_1 = \frac{8}{27}P_0 + \frac{4}{9}P_1 + \frac{2}{9}P_2 + \frac{1}{27}P_3, \qquad \text{. . . (4.36)}$$

The solution of the four linear simultaneous equations will lead to:

$$P_1 = \frac{1}{3}(2A_1 - A_2), \qquad \text{. . . (4.37)}$$

$$P_2 = \frac{1}{2}(2A_2 - A_1), \qquad \text{. . . (4.38)}$$

Where

$$A_1 = \frac{1}{6}(27y_1 - 8y_0 - y_3), \qquad \text{. . . (4.39)}$$

$$A_2 = \frac{1}{6}(27y_2 - y_0 - 8y_3), \qquad \text{. . . (4.40)}$$

Thus we can use the equations (4.37), (4.38), (4.39) and (4.40) to determine the values of (P_0, P_1, P_2, P_3). Then the equation (4.33) could be used to interpolate the points between (x_1 and x_2) by using μ value $\mu = 1 + \frac{x - x_1}{x_2 - x_1}$.

Algorithm (4.5): A program of resampling by using Bezier interpolation method.

Inputs: (1) Nosamp= No. of input samples

(2) u = Ratio of up sampling

(3) yup() = Samples after decimation

(4) M = Up sampling rate -1

Out put: (1) Y() = Reconstructed samples

Continue

```
Evaluated the coefficient P1, and P2 from the Cubic eqaution
   For j = 1 to M
      U2 = U * U
     Uu = 1 - U: Uu2 = Uu * Uu
     Y() = Yp(0) * Uu2 * Uu + P1 * U * Uu2 + P2 * U2 * Uu +
Yp(3) * U2 * U
```

4.3 Quantization Processes

Quantization is a rounding off (approximation) method. By this process, the wide ranges of real numbers are mapped to a small set of integers which require less number of bits in representation (i.e. in storage or transmission). The quantization can be arranged in either a uniform fashion, i.e., uniformly distributed from the highest expected value to the lowest expected value, or non–uniformly distributed. Uniform quantizers allow the designer to designate a minimum value for the error of any quantized value, while non–uniform quantizers can give a significant increase in accuracy, especially when the statistics of the incoming signal are known. The block diagram shown in figure (4.2) illustrate the steps of implementing the quantization methods (uniform quantization, and non–unifiorm quantization).

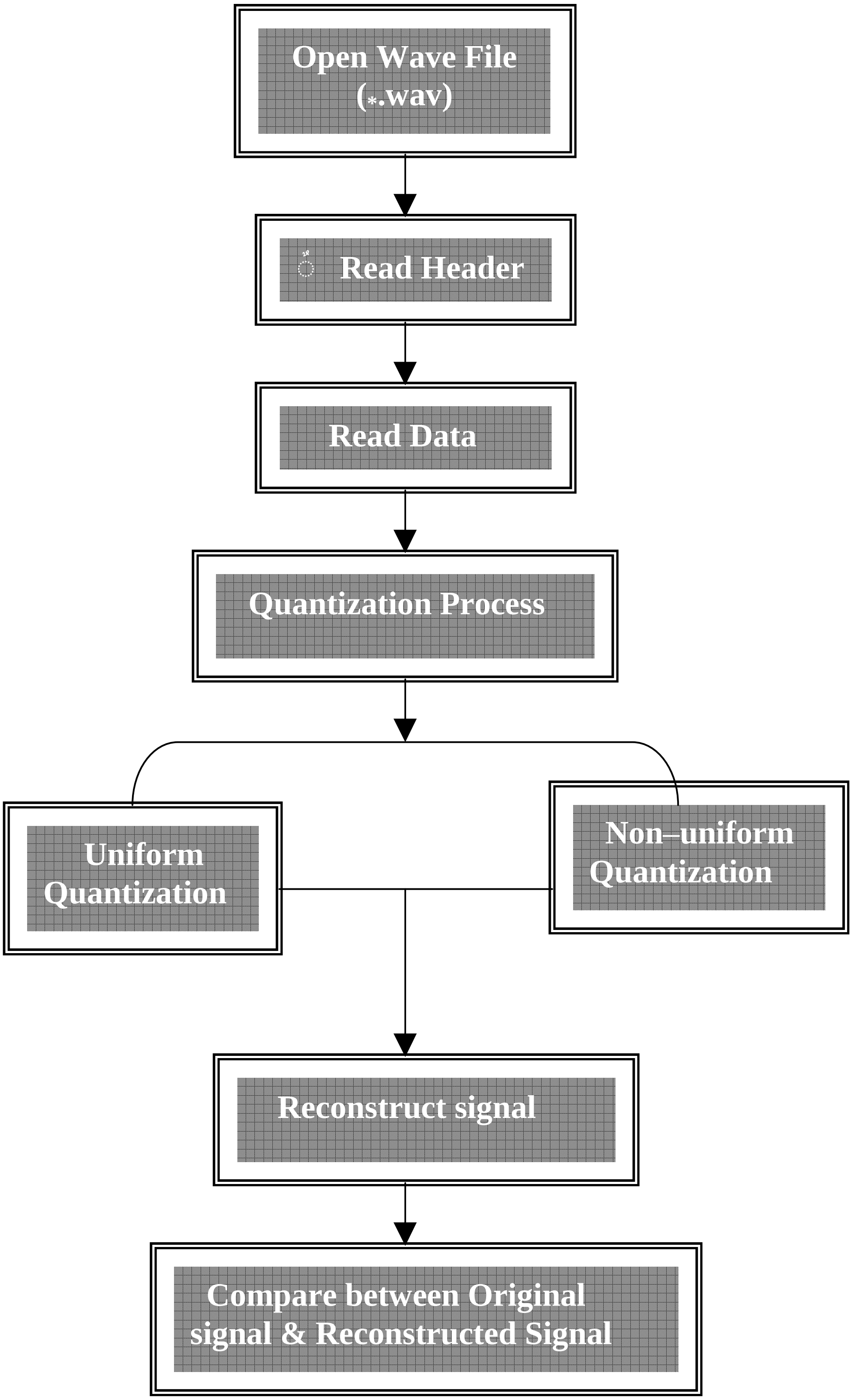


Figure (4.2) Block Diagram of de–quantization process

4.3.1 Uniform Quantization

The process of quantization and reconstruction (de–quantization) are extremely simple due to the linear relationship between the reconstructed values and the quantization indices (i). For computation of the index i from the signal value x, it is sufficient to divided the continuous value by the quantization step (Δ) and perform nearest integer rounding. Optionally, an offset shift can be compensated in the quantization step. To compute the reconstruction value (y), scaling of the index by (Δ) and reverse offset shift must be performed. A uniform quantization process determines the optimum index i and the reconstructed (y) as follows:

$$i = \mathrm{cint}\left[\frac{\mathrm{x - offset}}{\Delta}\right], \qquad \ldots (4.35)$$

$$y = \mathrm{i} \times \Delta + \mathrm{offset}\,, \qquad \ldots (4.36)$$

```
Algorithm (4.6): A program of Uniform Quantization.
   Inputs: (1) Nosamp = No. of input samples
           (2) qs = Quantization step
   Output: (1) Y () = Reconstructed samples
   For I = 0, 1, . . . , Nosamp
      j = Nosamp – 128
      Yq (I) =  j / qs
    End For
   For I = 0, 1, . . . , Nosamp
        Y() = Yq  * qs +128

   End For
```

4.3.2 Non–uniform Quantization

It is useful if quantization errors are perceived as more severe at low amplitude ranges. For computation of the index i from the signal value x, it is sufficient to apply histogram equalization method. The first step in this method is to find the accumulated probability density:

$$P_{acm}(i) = \frac{\sum_{j=0}^{i} \mathrm{H\,(j)}}{\sum_{j=0}^{255} \mathrm{H\,(j)}}, \qquad ...\ (4.37)$$

Where H (j) is the histogram value of the j th level of the audio signal, $P_{acm}(i)$ is the accumulated probability of the i th level. Then the requantized value $(i^{/})$ of the level (i) is determined from follows:

$$i^{/} = \mathrm{P_{acm}(i) \times N}, \qquad ...\ (4.38)$$

Where $i^{/}$ the requantized signal and N is is the total number of quantized levels.

Algorithm (4.7): A program of non–Uniform Quantization.

Inputs: (1) Nosamp = No. of input samples

(2) N = No. of quantization level

Output: (1) Y () = Reconstructed samples

For I = 0, 1, . . . , Nosamp

Determine the histogram of the samples

End For

For I = 0, 1, . . . , Nosamp

P_{acm} = his/max

End For

Continue

```
For I = 0, 1, . . . , Nosamp
    Y () = Pacm × N
End For
```

4.4 Criteria Measures

Due to the decimation/upsampling and to the quantization/ dequantization the speech signals tend to be corrupted by distortion known as ***Noise***. ***Noise*** is usually modeled as a random signal, which is combined (added) with the signal of interest. This ***noise*** is usually viewed as an additive signal independent of the speech signal. To measure the quality of reconstructed signal compared with the original ones. A common measure used for this purpose is the **peak signal to noise ratio** (PSNR). It is familiar to workers in the field; it is also simple to calculate [**Douglas 87**].

The ratio of PSNR is calculated by the following equation:

$$PSNR = 10 \times \log_{10} \frac{(255)^2}{\delta_d^2}, \qquad \ldots (4.38)$$

Where

$$\delta_d^2 = \frac{1}{N} \sum_{i=1}^{N} (x_i - x_i^{/})^2, \qquad \ldots (4.39)$$

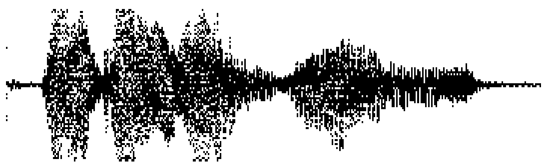
δ_d^2 is the Mean Square Error (MSE), x_i is original signal, and $x_i^{/}$ is reconstructed signal.

4.5 Resampling Test Results

Three files were adopted as test samples. Then the five types of interpolation methods are applied on each test samples to reconstruct the signal, and then we compute the values of PSNR and MSE for different down sample values.

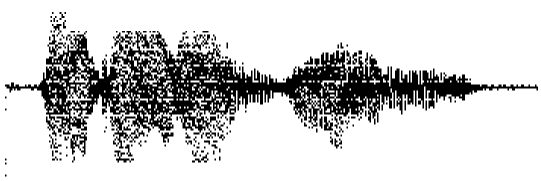
4.5.1 Test 1

The first test has 8–bit sampling resolution, (mono), sampling rate 22 kHz, and size 41080 bytes. Figure (4.3) illustrate the shape of the signal which has been tested.



Original signal

Sampled signal
with down sampling = 2



Reconstructed signal

Figure (4.3) Original and its sampled and reconstructed signal

The test results are shown in tables (4.1), (4.2), (4.3), (4.4), (4.5), (4.6), (4.7), (4.8), and (4.9).

Table (4.1) The results of 8–bit and size 41080 bytes with rate of down sampling =2

Interpolation Methods	MSE (dB)	PSNR (dB)
Linear	4.6	41.5
Lagrange	4.4	41.7
Cubic Spline	4.4	41.7
Quadratic	5.4	40.8
Bezier	4.4	41.7

Table (4.2) The results of 8–bit and size 41080 bytes with rate of down sampling =3

Interpolation Methods	MSE (dB)	PSNR (dB)
Linear	9	38.6
Lagrange	8.5	38.8
Cubic Spline	8.5	38.8
Quadratic	9.6	38.3
Bezier	8.5	38.8

Table (4.3) The results of 8–bit and size 41080 bytes with rate of down sampling = 4

Interpolation Methods	MSE (dB)	PSNR (dB)
Linear	14.1	36.7
Lagrange	12.5	37.2
Cubic Spline	12.5	37.2
Quadratic	14.5	36.5
Bezier	12.5	37.2

Table (4.4) The results of 8–bit and size 41080 bytes with rate of down sampling =5

Interpolation Methods	MSE (dB)	PSNR (dB)
Linear	21.2	34.9
Lagrange	18.2	35.5
Cubic Spline	18.2	35.5
Quadratic	21.3	34.9
Bezier	18.2	35.5

Table (4.5) The results of 8–bit and size 41080 bytes with rate of down sampling = 6

Interpolation Methods	MSE (dB)	PSNR (dB)
Linear	29.6	33.4
Lagrange	25.9	34
Cubic Spline	25.9	34
Quadratic	31.3	33.2
Bezier	25.9	34

Table (4.6) The results of 8–bit and size 41080 bytes with rate of down sampling =7

Interpolation Methods	MSE (dB)	PSNR (dB)
Linear	38	32.3
Lagrange	34.5	32.8
Cubic Spline	34.5	32.8
Quadratic	41.1	32
Bezier	34.5	32.8

Table (4.7) The results of 8–bit and size 41080 bytes with rate of down sampling = 8

Interpolation Methods	MSE (dB)	PSNR (dB)
Linear	50.4	31.1
Lagrange	48.2	31.3
Cubic Spline	48.2	31.3
Quadratic	54.2	30.8
Bezier	48.2	31.3

Table (4.8) The results of 8–bit and size 41080 bytes with rate of down sampling = 9

Interpolation Methods	MSE (dB)	PSNR (dB)
Linear	64.1	30
Lagrange	65.7	30
Cubic Spline	65.7	30
Quadratic	75.2	29.4
Bezier	65.7	30

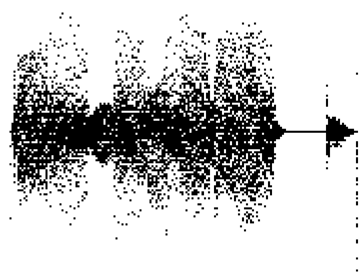
Table (4.9) The results of 8–bit and size 41080 bytes with rate of down sampling =10

Interpolation Methods	MSE (dB)	PSNR (dB)
Linear	77.9	29
Lagrange	83.7	29.4
Cubic Spline	83.7	29.4
Quadratic	95.1	29
Bezier	83.7	29.4

4.5.2 Test 2

The second test has 8–bit sampling resolution, (mono), sampling rate 22 kHz, and size 28213 bytes. Figure (4.4) illustrate the shape of the signal which has been tested.

Original signal



Sampled signal
with down sampling = 2

Reconstructed signal

Figure (4.4) Original and its sampled and reconstructed signal

The test results are shown in the tables (4.10), (4.11), (4.12), (4.13), (4.14), (4.15), (4.16), (4.17), and (4.18).

Table (4.10) The results of 8–bit and size 28213 bytes with rate of down sampling =2

Interpolation Methods	MSE (dB)	PSNR (dB)
Linear	6	40.3
Lagrange	4	42.2
Cubic Spline	4	42.2
Quadratic	5.8	40. 5
Bezier	4	42.2

Table (4.11) The results of 8–bit and size 28213 bytes with rate of down sampling =3

Interpolation Methods	MSE (dB)	PSNR (dB)
Linear	19.3	35.3
Lagrange	13.3	36.9
Cubic Spline	13.3	36.9
Quadratic	18.49	35.5
Bezier	13.3	36.9

Table (4.12) The results of 8–bit and size 28213 bytes with rate of down sampling = 4

Interpolation Methods	MSE (dB)	PSNR (dB)
Linear	41.6	32
Lagrange	33.3	33
Cubic Spline	33.3	33
Quadratic	44.4	31.6
Bezier	33.3	33

Table (4.13) The results of 8–bit and size 28213 bytes with rate of down sampling = 5

Interpolation Methods	MSE (dB)	PSNR (dB)
Linear	74.8	29.4
Lagrange	67.4	29.8
Cubic Spline	67.4	29.8
Quadratic	77.4	29.2
Bezier	67.4	29.8

Table (4.14) The results of 8–bit and size 28213 bytes with rate of down sampling = 6

Interpolation Methods	MSE (dB)	PSNR (dB)
Linear	115	27.5
Lagrange	110	27.7
Cubic Spline	110	27.7
Quadratic	130	27
Bezier	110.4	27.7

Table (4.15) The results of 8–bit and size 28213 bytes with rate of down sampling = 7

Interpolation Methods	MSE (dB)	PSNR (dB)
Linear	158.3	26.1
Lagrange	158.2	26.2
Cubic Spline	158.2	26.2
Quadratic	179.8	26
Bezier	158.2	26.2

Table (4.16) The results of 8–bit and size 28213 bytes with rate of down sampling = 8

Interpolation Methods	MSE (dB)	PSNR (dB)
Linear	204.7	25
Lagrange	206.4	25
Cubic Spline	206.4	25
Quadratic	234.5	24.4
Bezier	206.4	25

Table (4.17) The results of 8–bit and size 28213 bytes with rate of down sampling = 9

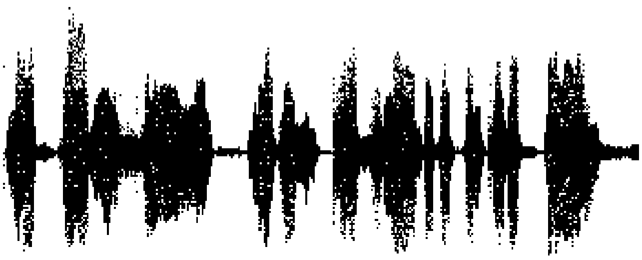
Interpolation Methods	MSE (dB)	PSNR (dB)
Linear	249.5	24
Lagrange	254.8	24.2
Cubic Spline	254.8	24.2
Quadratic	279.3	23.7
Bezier	254.8	24.2

Table (4.18) The results of 8–bit and size 28213 bytes with rate of down sampling = 10

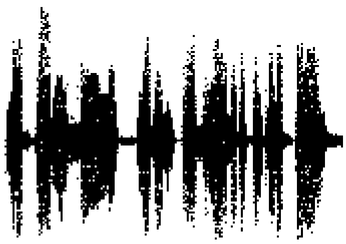
Interpolation Methods	MSE (dB)	PSNR (dB)
Linear	299.7	23
Lagrange	310.3	23.2
Cubic Spline	310.3	23.2
Quadratic	334.1	22.9
Bezier	310.3	23.2

4.5.3 Test 3

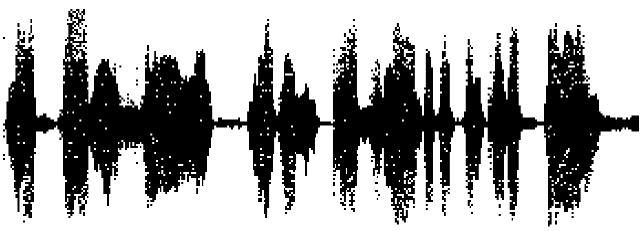
The third test has 8–bit sampling resolution, (mono), sampling rate 22 kHz, and size 117114 bytes. Figure (4.5) illustrate the shape of the signal which has been tested.



Original signal



Sampled signal
with down sampling = 2



Reconstructed signal

Figure (4.5) Original and its sampled and reconstructed signal

The test results are shown in the tables (4.19), (4.20), (4.21), (4.22), (4.23), (4.24), (4.25), (4.26) and (4.27).

Table (4.19) The results of 8–bit and size 117114 bytes with rate of down sampling = 2

Interpolation Methods	MSE (dB)	PSNR (dB)
Linear	6.1	40.3
Lagrange	5	41.2
Cubic Spline	5	41.2
Quadratic	6.2	40.2
Bezier	5	41.2

Table (4.20) The results of 8–bit and size 117114 bytes with rate of down sampling = 3

Interpolation Methods	MSE (dB)	PSNR (dB)
Linear	16.3	36
Lagrange	13.2	37
Cubic Spline	13.2	37
Quadratic	16.4	36
Bezier	13.2	37

Table (4.21) The results of 8–bit and size 117114 bytes with rate of down sampling = 4

Interpolation Methods	MSE (dB)	PSNR (dB)
Linear	32.3	33
Lagrange	27.7	33.7
Cubic Spline	27.7	33.7
Quadratic	34.3	32.8
Bezier	27.7	33.7

Table (4.22) The results of 8–bit and size 117114 bytes with rate of down sampling = 5

Interpolation Methods	MSE (dB)	PSNR (dB)
Linear	52.8	31
Lagrange	47.3	31.4
Cubic Spline	47.3	31.4
Quadratic	58.2	30.5
Bezier	47.3	31.4

Table (4.23) The results of 8–bit and size 117114 bytes with rate of down sampling = 6

Interpolation Methods	MSE (dB)	PSNR (dB)
Linear	79.4	29.1
Lagrange	77.5	29.2
Cubic Spline	77.5	29.2
Quadratic	92	28.5
Bezier	77.5	29.2

Table (4.24) The results of 8–bit and size 117114 bytes with rate of down sampling = 7

Interpolation Methods	MSE (dB)	PSNR (dB)
Linear	107.4	27.3
Lagrange	114	27.7
Cubic Spline	114	27.7
Quadratic	128.7	27
Bezier	114	27.7

Table (4.25) The results of 8–bit and size 117114 bytes with rate of down sampling = 8

Interpolation Methods	MSE (dB)	PSNR (dB)
Linear	136.4	26
Lagrange	150.7	26.3
Cubic Spline	150.7	26.3
Quadratic	163.1	26
Bezier	150.7	26.3

Table (4.26) The results of 8–bit and size 117114 bytes with rate of down sampling = 9

Interpolation Methods	MSE (dB)	PSNR (dB)
Linear	167.5	25.9
Lagrange	184.4	25.5
Cubic Spline	184.4	25.5
Quadratic	195.3	25.2
Bezier	184.4	25.5

Table (4.27) The results of 8–bit and size 117114 bytes with rate of down sampling = 10

Interpolation Methods	MSE (dB)	PSNR (dB)
Linear	198.8	25.1
Lagrange	212.4	24.8
Cubic Spline	212.4	24.8
Quadratic	223.1	24.6
Bezier	212.4	24.8

The results of the three tested samples show that the resampling method which they are Lagrange, Cubic spline, and Bezier give the same results (i. e) there is no difference between them. Also the results show that when increases the rate of down sample the PSNR will be decreases this lead to the difference in the audio quality. The results are different because it depend on the conditions and the parameters of each method.

4.6 Listening Test

Also the results are tested subjectively. The listening test is done to the resampling algorithm for it importance, the test shows when increase the rate of down sampling there is perceptually noticeable difference to the listener. The four tested sample listens are from different area includes student in the master science, and ordinary people. The choice of the listing is random. The listening test results are shown in the tables (4.28), (4.29), and (4.30).

Table (4.28) Listening test results of 8–bit and size 41080 bytes

Down sampling rate	Person 1	Person 2	Person 3	Person 4
2	Excellent	Excellent	Excellent	Excellent
3	Excellent	Excellent	Excellent	Excellent
4	Excellent	Very good	Very good	Excellent
5	Excellent	Very good	Very good	Excellent
6	Very good	Very good	Good	Very good
7	Very good	Good	Good	Very good
8	Good	Bad	Good	Good
9	Bad	Bad	Bad	Bad
10	Bad	Bad	Bad	Bad

Table (4.29) Listening test results of 8–bit and 28213 bytes

Down sampling rate	Person 1	Person 2	Person 3	Person 4
2	Excellent	Excellent	Excellent	Excellent
3	Excellent	Excellent	Excellent	Excellent
4	Very good	Very good	Very good	Very good
5	Very good	Very good	good	Very good
6	good	Very good	Good	good
7	good	Good	Good	good
8	Bad	Bad	Good	Bad
9	Bad	Bad	Bad	Bad

Table (4.30) Listening test results of 8–bit wand size 117114 bytes

Down sampling rate	Person 1	Person 2	Person 3	Person 4
2	Excellent	Excellent	Excellent	Excellent
3	Excellent	Excellent	Excellent	Very good
4	Excellent	Very good	Very good	Very good
5	Excellent	Very good	Very good	good
6	Very good	Very good	Good	good
7	good	Good	Good	Bad
8	Bad	Bad	Good	Bad
9	Bad	Bad	Bad	Bad
10	Bad	Bad	Bad	Bad

4.7 Dequantization Test Results

Three files are adapted as test samples. Then we applied two types of quantization methods on each test samples to reconstruct the signal, and then we compute the value of PSNR and MSE.

4.7.1.a Test 1

The first test has 8–bit sampling resolution, (mono), sampling rate 22 kHz, and size 41080 bytes. Figure (4.6) illustrate the shape of the signal which has been tested.

Original signal

Quantized signal with quantization step = 2

De-quantized signal

Figure (4.6) Original and its quantized and reconstructed signal

The test results of uniform quantization are shown in the table (4.31).

Table (4.31) Results of 8–bit and size 41080 bytes with different steps of uniform quantization

Step of quantization	MSE (dB)	PSNR (dB)
2	0.3	54
3	0.6	50
4	1.4	46.7
5	1.8	45.5
6	2.8	43.6
7	3.5	42.7
8	4.7	41.4
9	5.6	40.6

Step of quantization	MSE (dB)	PSNR (dB)
10	7.1	39.6
11	8.2	39
12	9.8	38.2
13	11.2	37.6
14	13	37

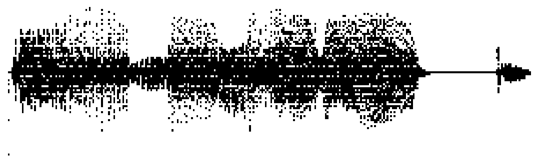
Table (4.32) Listening test results of 8–bit and size 41080 bytes of uniform quantization

Step of quantization	Person 1	Person 2	Person 3	Person 4
2	Excellent	Excellent	Excellent	Excellent
3	Excellent	Excellent	Excellent	Excellent
4	Excellent	Excellent	Very good	Excellent
5	Excellent	Excellent	Very good	Excellent
6	Very good	Excellent	Very good	Very good
7	Very good	Very good	Very good	Very good
8	Very good	Very good	Very good	Good
9	Very good	Very good	Good	Good
10	Very good	Very good	Good	Good
11	Very good	Good	Good	Good
12	Good	Good	Good	Good
13	Good	Bad	Bad	Bad
14	Bad	Bad	Bad	Bad

4.7.2.a Test 2

The second test has 8–bit sampling resolution, (mono), sampling rate 22 kHz, and size 28213 bytes. Figure (4.7) illustrate the shape of the signal which has been tested.

Original signal



Quantized signal with quantization step = 2

De-quantized signal

Figure (4.7) Original and its quantized and reconstructed signal

The test results of uniform quantization are shown in the table (4.33).

Table (4.33) Results of 8–bit and size 28213 bytes with different steps of uniform quantization

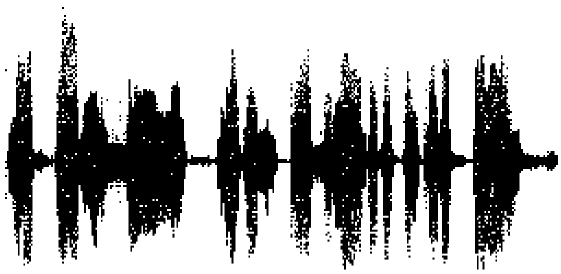
Step of quantization	MSE (dB)	PSNR (dB)
2	0.5	51.3
3	0.6	50.1
4	1.3	47
5	1.7	45.7
6	2.8	43.8
7	3.4	42.7
8	4.7	41.4
9	5.6	40.6
10	7.1	39.6
11	8.3	39
12	10	38.1
13	11.4	37.5
14	13.3	36.9

Table (4.34) Listening test results of 8–bit and size 28213 bytes of uniform quantization

Step of quantization	Person 1	Person 2	Person 3	Person 4
2	Excellent	Excellent	Excellent	Excellent
3	Excellent	Excellent	Excellent	Excellent
4	Excellent	Excellent	Very good	Excellent
5	Excellent	Excellent	Very good	Excellent
6	Very good	Excellent	Very good	Very good
7	Very good	Excellent	Very good	Very good
8	Very good	Excellent	Very good	Very good
9	Very good	Very good	Good	Very good
10	Very good	Very good	Good	Very good
11	Good	Very good	Good	Very good
12	Good	Very good	Good	Good
13	Good	Good	Bad	Good
14	Bad	Bad	Bad	Bad

4.7.3.a Test 3

The third test has 8–bit sampling resolution, (mono), sampling rate 22 kHz, and size 117114 bytes. Figure (4.8) illustrate the shape of the signal which has been tested.



Original signal

Quantized signal with quantization step = 2

De-quantized signal

Figure (4.8) Original and its quantized and reconstructed signal

The test results of uniform quantization are shown in the table (4.35).

Table (4.35) Results of 8–bit and size 117114 bytes with different steps of uniform quantization

Step of quantization	MSE (dB)	PSNR (dB)
2	0.5	51.1
3	0.6	50
4	1.4	46.7
5	1.8	45.5
6	2.8	43.6
7	3.5	42.7
8	4.7	41.4
9	5.6	40.6
10	7.1	39.6
11	8.2	39
12	9.8	38.2
13	11.2	37.6
14	13	37

Table (4.36) Listening test results of 8–bit and size 117114 bytes of uniform quantization

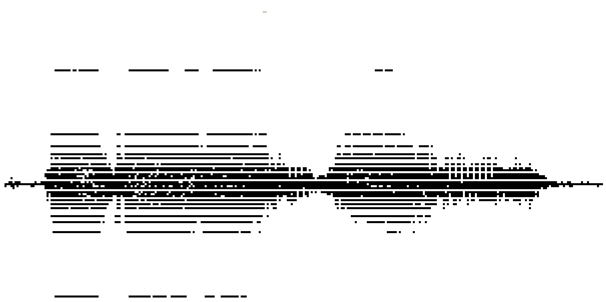
Step of quantization	Person 1	Person 2	Person 3	Person 4
2	Excellent	Excellent	Excellent	Excellent
3	Excellent	Excellent	Excellent	Excellent
4	Excellent	Very good	Very good	Excellent
5	Very good	Very good	Very good	Excellent
6	Very good	Very good	Very good	Very good
7	Very good	Very good	Very good	Very good
8	Very good	Good	Very good	Very good
9	Good	Good	Good	Very good
10	Good	Good	Good	Very good
11	Good	Good	Good	Good
12	Good	Bad	Good	Good
13	Bad	Bad	Good	Good
14	Bad	Bad	Bad	Bad

4.7.1.b Test 1

The first test has 8–bit sampling resolution, (mono), sampling rate 22 kHz, and size 41080 bytes. Figure (4.9) illustrate the shape of the signal which has been tested.

Original signal

Quantized signal with
No. of level = 30



De-quantized signal

Figure (4.9) Original and its quantized and reconstructed signal

The test results of non–uniform quantization are shown in the table (4.37).

Table (4.37) Results of 8–bit and size 41080 bytes with different levels of non–uniform quantization

No. of level	MSE (dB)	PSNR (dB)
30	41.9	32
26	53.2	30.9
22	86.3	28.8
18	111.3	27.7
14	179	25.6
10	279.4	23.7
8	436	21.7
7	436	21.7
5	756	19.3
4	1722.4	15.8

Table (4.38) Listening test results of 8–bit and size 41080 bytes of non–uniform quantization

No. of level	Person 1	Person 2	Person 3	Person 4
30	Excellent	Excellent	Excellent	Excellent
26	Excellent	Excellent	Excellent	Excellent
22	Excellent	Very good	Very good	Excellent
18	Excellent	Very good	Very good	Very good
14	Very good	Very good	Very good	Very good
10	Very good	Very good	Very good	Good
8	Good	Good	Very good	Good
7	Good	Good	Good	Good
5	Bad	Bad	Good	Good
4	Bad	Bad	Bad	Bad

4.7.2.b Test 2

The second test has 8–bit sampling resolution, (mono), sampling rate 22 kHz, and size 28213 bytes. Figure (4.10) illustrate the shape of the signal which has been tested.

Original signal

Quantized signal with
No. of level = 30

De-quantized signal

Figure (4.10) Original and its quantized and reconstructed signal

The test results of non–uniform quantization are shown in the table (4.39).

Table (4.39) Results of 8–bit and size 28213 bytes with different levels of non–uniform quantization

No. of level	MSE (dB)	PSNR (dB)
30	0.8	49.2
26	0.8	49.1
22	0.8	40.2
18	110.3	27.7
14	322.8	23
10	540	20.8
8	540.8	20.8
7	540.8	20.8
5	1163	17.5
4	1987	15.1

Table (4.40) Listening test results of 8–bit and size 28213 bytes of non–uniform quantization

No. of level	Person 1	Person 2	Person 3	Person 4
30	Excellent	Excellent	Excellent	Excellent
26	Excellent	Excellent	Very Good	Excellent
22	Very good	Excellent	Very good	Very good
18	Very good	Very good	Very good	Very good
14	Very good	Good	Very good	Good

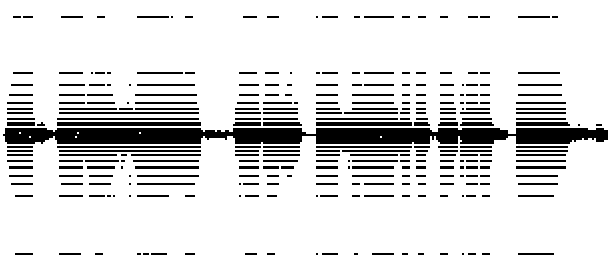
No. of level	Person 1	Person 2	Person 3	Person 4
10	Good	Good	Very good	Good
8	Good	Good	Good	Good
7	Good	Good	Good	Good
5	Bad	Bad	Good	Good
4	Bad	Bad	Bad	Bad

4.7.3.b Test 3

The third test has 8–bit sampling resolution, (mono), sampling rate 22 kHz, and size 117114 bytes. Figure (4.11) illustrate the shape of the signal which has been tested.

Original signal

Quantized signal with
No. of level = 30



De-quantized signal

Figure (4.11) Original and its quantized and reconstructed signal

The test results of non–uniform quantization are shown in the table (4.41).

Table (4.41) Results of 8–bit and size 117114 bytes with different levels of non–uniform quantization

No. of level	MSE (dB)	PSNR (dB)
30	37	32.4
26	52.2	31
22	67	30
18	93.8	28.4
14	139.1	26.7
10	238.5	24.3
8	347.2	22.7
7	347.2	22.7
5	619.8	20.2
4	1358.6	16.8

Table (4.42) Listening test results of 8–bit and size 117114 bytes of non–uniform quantization

No. of level	Person 1	Person 2	Person 3	Person 4
30	Excellent	Excellent	Excellent	Excellent
26	Excellent	Excellent	Excellent	Excellent
22	Excellent	Very good	Very good	Excellent
18	Excellent	Very good	Very good	Very good
14	Very good	Very good	Very good	Very good
10	Very good	Good	Very good	Good
8	Good	Bad	Good	Good
7	Good	Bad	Good	Good
5	Bad	Bad	Good	Bad
4	Bad	Bad	Bad	Bad

Chapter One
General Introduction

1.1 Introduction

Over the past two decades, improvements in technology have changed the way of recording the music and the used digital media. Today, we use computers to record audio and save it on of CDs, or other storage devices. In order to transform sound into a digital format, one must sample the sound. This process takes place while one is recording. The computer takes a snapshot of the sound level at small time intervals. The number of samples taken in each second is called the sampling rate. The more samples that are taken, the better sound quality. For instance, audio sampled at 44 kHz is better than audio sampled at 22 kHz. It also means more storage space is required to record higher quality digital sounds

Bandlimited interpolation of discrete–time signals is a basic tool has extensive applications in digital signal processing. In general, the problem is to correctly compute signal values at arbitrary continuous times from a set of discrete-time samples of the signal amplitude. In other words, we must be able to interpolate the signal between samples. Shannon's sampling theorem tells us the signal can be exactly and uniquely reconstructed for all time from its samples by bandlimited interpolation [**Smi 04**].

Signal requantization is applied in digital audio systems whenever the word−length of the audio samples needs to be reduced. This is the case for instance when an audio signal has to be stored on a CD and was originally produced from the output of a digital audio system that operates with more than 16 bit precision. In some applications, like multimedia, gaming, or mobile communication devices, requantization to 8 bit or 12 bit could be an economically interesting alternative to other forms of data compression because requantized data can be send directly to the D/A converter, while encoded data requires a decoder. Signal requantization inevitably introduces an error, which can cause two types of audible problems. The first is a background noise that may be audible by itself. It can usually occur when (part of) the error signal is uncorrelated with the original audio. When the error is correlated with the signal, linear or nonlinear distortions may cause alterations in the perceived quality of the signal itself. At low signal levels, the second problem is usually much more serious. Dither (it means how well it remove quantization distortion whenever there's some requantization going on), noise can be used to remove the correlation between the error and the signal at the expense of increased noise energy [**Kon 003**].

1.2 Review of Previous Works

Among the massive published research work in the literature concerned with speech analysis, the following list of recent researches illustrate some the important research work conducted in the field:

1. **Mclain (1976)** describe a method for smooth interpolation in one dimension between data provided at a set of points arbitrarily distributed. Because the method ensures the continuity of the resulting points, and its first two derivatives, it is suitable for graphical application. This method is called spline and its coefficients are not found from the values at the nodes, as in the usual applications of bicunbic spline when the data are given, but are calculated using a statistical least squares fit, so that resulting curves fits as closely as possible with the data. The curves produced by this technique are, of course, smooth but it will not in general pass through all the data points [**Mcl 76**].

2. **Collins (1998)** described that real–time synthesis methods deal with some form of manageable control data as a handle to the job of producing a continuous output stream of digital audio in the time domain. Changes in the control data have an immediate effect on the synthesis, though the relation between audio result and the parameter changed can be obscure. Interfacing to low level digital audio requires a manageable representation of control data to specify the shape of a waveform. The model solved this by the use of interpolating splines defined by an ordered list of control points [**Col 98**].
3. **Shykula and Seleznjev (2000)** considered quantization of a signal (or random process) in a probabilistic framework. The presented quantization method can be applied to signal coding and storage capacity problems. In order to demonstrate the general approach, the uniform quantization of a Gaussian process was studied in more detail. They investigated asymptotic properties of some accuracy characteristics, such as rate and distortion, in terms of correlation structure of the original random process when quantization cellwidth tends to zero [**Shy 00**].
4. **Koning and Verhelst (2003)** presented the idea of using Least Squares (LS) theory for optimal noise shaping of audio signals; they indicated that the suggested approach provides shorter and more straightforward proof of known properties of dithered and nondithered noise shaping. In contrast with the standard theory, this approach shows how noise shaping filters that attain the theoretical optimum can be designed in practice. Also they presented some produced results from an experimental noise shaping system for minimally audible signal requantization that is based on the suggested filter design method and a simple masking model. In listening experiments, this system was

unanimously preferred over the alternatives which included straightforward requantization, dithered requantization and optimized fixed noise shaping [**Kon 03**].

5. **Simth (2004)** described a technique for resampling algorithm which evaluates a signal at any time specifiable by a fixed point number. In addition, one low pass filter was used, regardless of the sampling rate conversion factor. The algorithm effectively implements the “analog interpretation” of rate conversion, in which a certain low pass filter impulse response must be available as a continuous function. Continuity of the impulse response is simulated by linearly interpolating between samples of the impulse response stored in a table. Due to the relatively low cost of memory, the method is quite practical for hardware implementation [**Sim 04**].

1.3 Aim of the Thesis

The present work aims to investigate the performance of some selected resampling methods on the digital audio signal, which they are Linear, Quadratic, Cubic spline, Lagrange, and Bezier in order to reduce the number of samples while the audio quality is maintained. Also the present work aims to investigate the performance of some uniform and non–uniform quantization methods in order to make the requantization levels of the digital audio data so small such that the audio quality is maintained.

1.4 Thesis Layout

In addition to chapter one, there are four chapters, which deal with the ways of resampling and requantization of the digital wave.

Chapter Two Entitle ‘‘***Digital Audio Processing***’’

This chapter includes some basic of signal processing concepts dealing with digital audio wave as a digital signal.

Chapter Three Entitle ‘‘***Resampling and Requantization***’’

This chapter presents a short description for some selected resampling methods, which they are linear, Lagrange, Cubic spline, Quadratic, Bezier, and there is some descrition of each one of them. The chapter also contains a description of requantization methods which they are uniform and non–uniform methods.

Chapter Four Entitle ‘‘***Experimental Results***’’

It includes a summary of the practical current research work. Also, the analysis results were presented in form of tables.

Chapter Five Entitle‘‘***Conclusions and Future Work***’’

It includes some of conclusions derived from the investigation of test results, which present in chapter four. Also, this chapter presents some future work suggestions concerned with the field of resampling and requantization for audio data.

Chapter Three
Resampling and Requantization

3.1 Introduction

The discrete–time signals is a basic way for representing signals in digital form, it has an extensive applications using digital signal processing. In general, the problem is to correct computer signal values at arbitrary continuous times from a set of discrete–time samples of the signal amplitude. In other words, we must be able to interpolate the signal between samples. Since the original signal is always assumed to be band limited to half the sampling rate. *Shannon's sampling theorem* tells us the signal can be exactly and uniquely reconstructed for all time from its samples by ***interpolation*** [**Sim 04**].

The concept of interpolation is the selection of a function f (x) from a given class of functions in such a way that the graph of y=f (x) passes through a finite set of given data points. Interpolation method has a number of important uses. Its primary use is to furnish some mathematical tools that are used in developing methods in the areas of approximation theory, numerical integration, and the numerical solution of differential equations. A second use is in developing means for working with functions that are stored in a tabular form [**Mcl 79**].

When the **sample** is assigned into a numeric value that the computer or digital circuit can use or store in a process called **quantization.** The number of available values is determined by the number of bits used for each sample. Each additional bit doubles the number of values available (1–bit samples have 2 values, 2–bit samples have 4 values, etc.). When a sample is quantized, the analog amplitude has to be rounded off to the nearest available digital value. This rounding–off process is called **approximation** [**Has 01**].

3.2 Resampling

The process of converting from digital back to analog is called reconstruction a good example is given by audio CDs. The music is stored in a digital form from which a CD player reconstructs the continuous (analog) waveform that we listen to. The reconstruction process is basically one of interpolation [**Mcc 98**].

Thus, an understanding of sampling and reconstruction is a good foundation for producing good–quality signals.

In the sampling/reconstruction problem we have to deal with three distinct signals: the continuous signal f, the discrete signal f_d, and the reconstructed signal f_r. Ideally we aimed to make the reconstructed signal equal to continuous signal ($f_r = f$) when this happens we say the reconstruction is exact. Exact reconstruction is not always possible.

The aim of reconstruction techniques is to minimize the error $|f - f_r|$. Reconstruction techniques are very important in the manipulation of signals in the computer, for at least two reasons:

1. In the solution of certain problem we need a continuous representation of the signal.

2. A good knowledge of the reconstruction techniques used by a given output device is important in the creation or choice of a algorithms to process the signal to be displayed on that device [**Gom 97**].

3.2.1 Linear Interpolation

The simplest method of deforming an object is to create in–between a series of transitional stages between two static positions. The in–between, or the sequence of intermediate shapes, are all generated from the given beginning and final static positions; these are also called **key positions** or **extremes**. ***Linear interpolation*** is the method of calculating any number of new values between two existing values [**Ker 86**].

Linear interpolation is definitely the most popular and most widely used reconstruction method. The reasons for this are that it is simple and pretty straight forward to implement, and the results are usually not so linear. Linear interpolation in one dimension results it is simply connecting sampling points using straight lines.

The simplest kind of interpolation is ***linear interpolation***. Assuming some desired function f (x), which is continuous and differentiable at all points. Thus, for n+1 different values of x, not necessarily evenly spaced, we are given the corresponding values of f (x). We assume here that both the x_i and the corresponding f (x_i) are given either exactly, or within some specified accuracy. Figure (3.1) shows the function f (x) and the corresponding values of x. Which are shown as heavy black points on the curve.

To use **linear interpolation**, we draw a straight line between two points one on each side of the unknown point x; in this case, we draw a straight line AD between the points at x_3 and x_4.

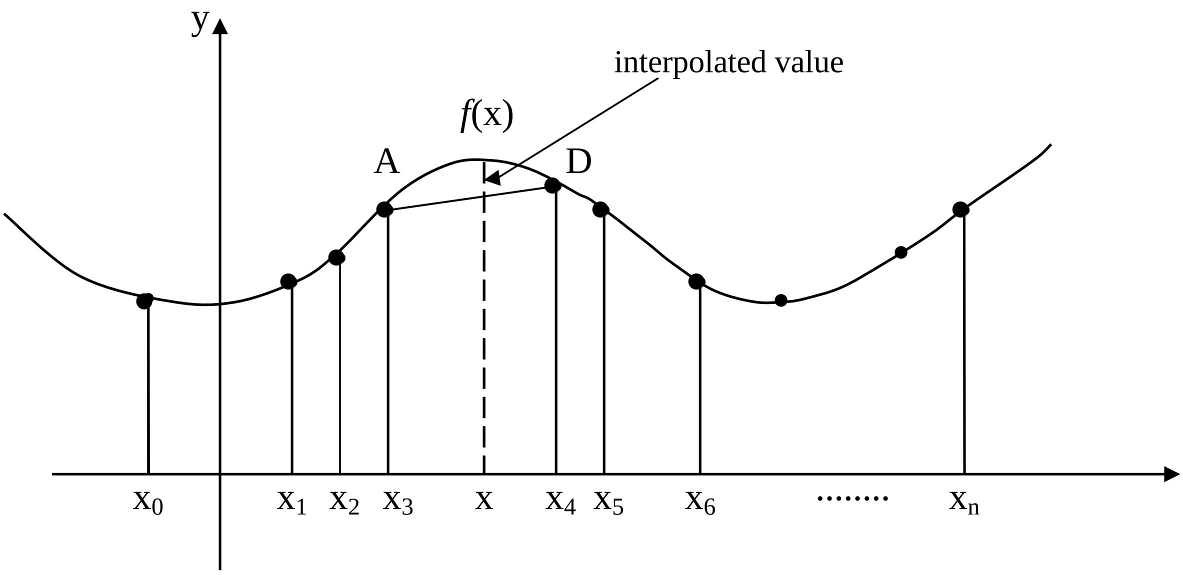


Figure (3.1) Linear interpolation

Having drawn this line, as in figure (3.1), we now approximate the curve in the region between, in this case, x_3 and x_4 by the straight line, which is shown magnified in figure (3.2) using similar triangles, we can get the proportion

$$\frac{BC}{AC} = \frac{DE}{AE} \quad , \qquad \ldots (3.1)$$

Which we can solve for BC:

$$BC = \frac{AC}{AE} DE \quad , \qquad \ldots (3.2)$$

$$f(x)_{int} - f(x_3) = \frac{x - x_3}{x_4 - x_3} [f(x_4) - f(x_3)] \, , \qquad \ldots (3.3)$$

So, the resulting interpolation value for f (x) will be:

$$P(x) = f(x)_{\text{int}} = f(x_3) + \frac{x - x_3}{x_4 - x_3}[f(x_4) - f(x_3)], \qquad \ldots (3.4)$$

Where P (x) is the interpolating approximation to f (x). In general, suppose we wish to find the value of f (x) for some x located between x_i and x_{i+1} [**Sta 70**]

Then the interpolated value p (x), which is only an approximation for f (x), is given by

$$P(x) = f(x_i) + \frac{x - x_i}{x_{i+1} - x_i}[f(x_{i+1}) - f(x_i)], \qquad \ldots (3.5)$$

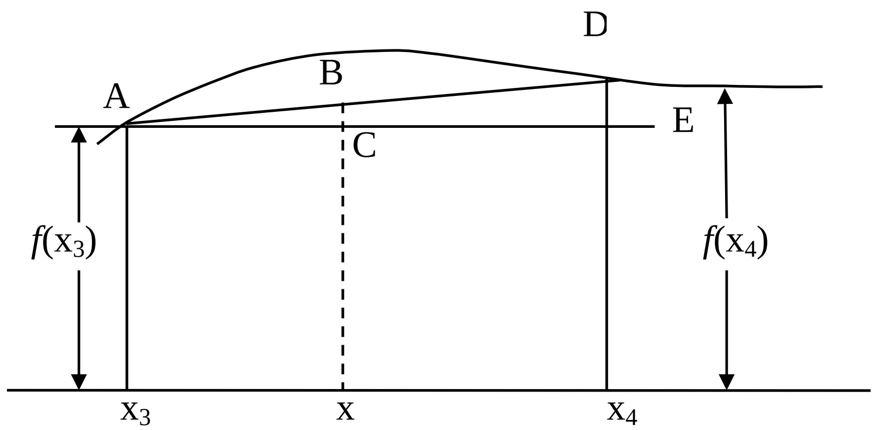


Figure (3.2) Derivation of linear interpolation formula

3.2.2 Lagrange Interpolation

Consider the problem of determining a polynomial of degree 2 that passes through the distinct points (x_0,y_0) and (x_1,y_1) [**Bur 85**].

Consider the polynomial

$$P(x) = \frac{(x - x_1)}{(x_0 - x_1)} y_0 + \frac{(x - x_0)}{(x_1 - x_0)} y_1 , \qquad \ldots (3.6)$$

When $x = x_0$, then

$$P(x_0) = y_0 = f(x_0) , \qquad \ldots (3.7)$$

And when $x = x_1$, then

$$P(x_1) = y_1 = f(x_1) , \qquad \ldots (3.8)$$

For the case we need to construct (for each k = 0, 1, . . ., n) a quotient $L_{n,k}(x)$ with the property that $L_{n,k}(x_i)=0$ when $i \neq k$ and $L_{n,k}(x_k) = 1$. To satisfy that $L_{n,k}(x_i)=0$ for each $i \neq k$ requires that numerator of $L_{n,k}$ contain the term $(x - x_0)(x - x_1) \ldots (x - x_{k-1})(x - x_{k+1}) \ldots (x - x_n)$.

To satisfy $L_{n,k}(x_k)=1$, the denominator of L_k must be equal to (1) when $x = x_k$. Thus,

$$L_{n,k}(x) = \frac{(x-x_0)\ldots(x-x_{k-1})(x-x_{k+1})\ldots(x-x_n)}{(x_k-x_0)\ldots(x_k-x_{k-1})(x_k-x_{k+1})\ldots(x_k-x_n)} = \prod_{\substack{i=0\\ i\neq k}}^{n} \frac{(x-x_i)}{(x_k-x_i)}, \qquad \ldots (3.9)$$

If $x_0, x_1, \ldots, x_n$ are (n+1) distinct numbers and f is a function whose values are given at these numbers, then there exists a unique polynomial p of degree at most n with property that

$$f(x_n) = f(x_k) \qquad \text{for each } k = 0, 1, \ldots, n , \qquad \ldots (3.10)$$

$$p(x) = f(x_0)\,L_{n,0}(x) + \cdots + f(x_n)\,L_{n,n}(x) = \sum_{k=0}^{n} f(x_k)\,L_{n,k}(x), \qquad \ldots (3.11)$$

Where

$$L_{n,k}(x) = \frac{(x - x_0)(x - x_1)\ldots(x - x_{k-1})(x - x_{k+1})\ldots(x - x_n)}{(x_k - x_0)(x_k - x_1)\ldots(x_k - x_{k-1})(x_k - x_{k+1})\ldots(x_k - x_n)}, \qquad \ldots (3.12)$$

$$= \prod_{\substack{i=0 \\ i \neq k}}^{n} \frac{(x - x_i)}{(x_k - x_i)} \qquad \text{for each } k = 0, 1, \ldots, n, \qquad \ldots (3.13)$$

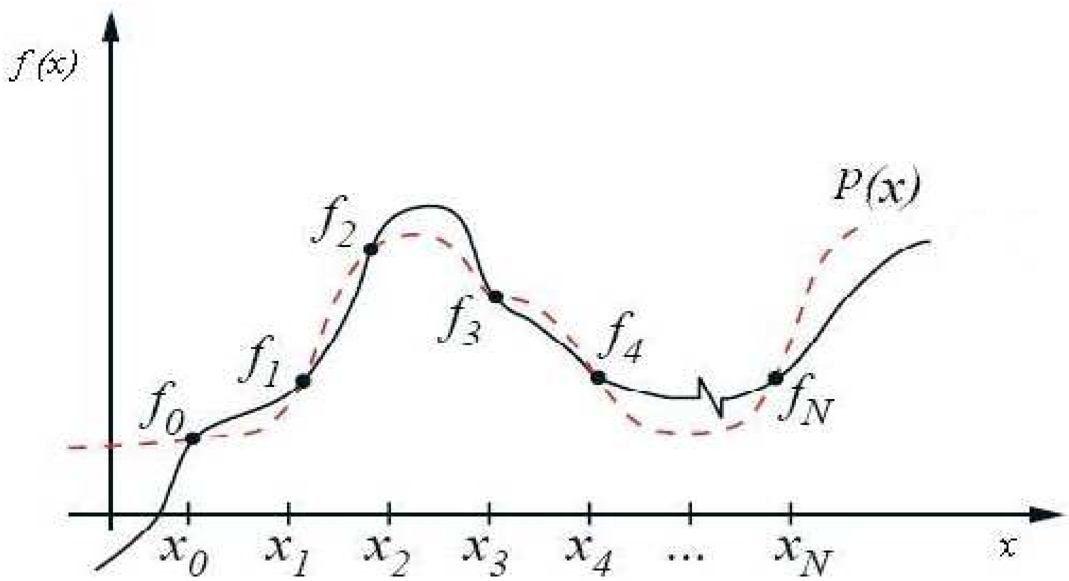


Figure (3.3) Lagrange interpolation

$f(x)$=exact function of which only N+1 discrete values are known and used to an interpolating or approximating function p(x).

P(x)=approximating or interpolating function. This function will pass through all specified N+1 interpolation points (also referred to as data points or nodes) [**Ron 02**].

The interpolated curves tend to oscillate about the exact result. Smooth functions are treated more accurately than oscillatory ones or ones with concentrated curvature. For this reason, Lagrange interpolation with more than three or four points is rarely used. Piecewise Lagrange interpolation offers some improvement, but suffers from having discontinuous derivation at the points that join the segments and many cause trouble if the result is to be differentiated [**Fer 81**].

This approximation to the function is not "smooth" (smoothness usually refers to the continuity of the derivatives) because, at the end–points (sometimes known as nodes) of each subinterval the derivative of the approximation is discontinuous. We can try to make the approximation smoother by using piecewise quadratic, rather than piecewise ***Linear***, approximation. A ***quadratic*** has three free parameters, two of which are determined by the function values at the ends of the subinterval, leaving the third free to be used to smooth the approximation. Unfortunately, there are not enough free parameter to ensure smoothness over the whole interval; the approximation cannot match the derivatives of the function at the end–points of the interval. This can be achieved, however, by using ***Cubic spline*** [**Atk 87**].

3.2.3 Cubic Spline Interpolation

Cubic spline is an equation of degree seven. Splines are drafting aids used to draw smooth curve passing through a set of points. Weights are attached at the points to be connected and a flexible stripe is shaped around the weights. A polynomial fitted to many data points could exhibit erratic behavior. Splines are smooth and continuous across the interval.

Cubic splines have the advantage of sufficient free parameters to ensure continuity of first and second derivatives throughout the interval, and to satisfy a derivative condition at the ends of the interval. The disadvantage of approaching an approximation problem is that at each of the end points of the subintervals, there is no assurance of differentiability, which, in a geometric context, means that the interpolating function is not "smooth" at these points.

It is important to note that the construction of a ***Cubic spline*** does not assume that the derivatives of the interpolant agree with those of the function any where except, perhaps, at the ends of the interval [**Burden & Faires 85**].

(i) $s^{//}(\mathrm{x}_0) = \mathrm{s}^{//}(\mathrm{x}_\mathrm{n}) = 0$, . . . (3.14)

(ii) $s^{//}(x_0) = f_0^{/}$ and $\mathrm{s}^{/}(x_n) = f_n^{/}$, . . . (3.15)

When condition (i) is satisfied the spline is called a **natural spline**. The condition (ii) is called a **clamped spline** [**Atk 87**].

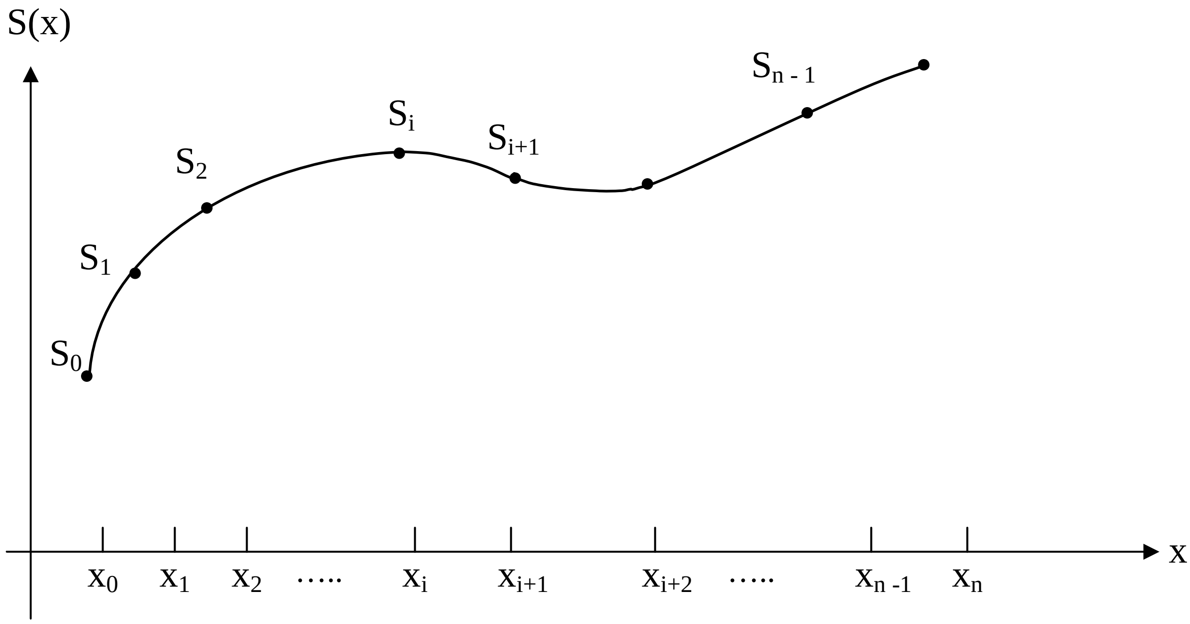


Figure (3.4) Cubic spline interpolation

We turn now to the specific problem of obtaining a ***Cubic spline*** function which interpolates the function f at $x_0, x_1, \ldots, x_N$. It will be convenient to introduce the following notation; in each of the subintervals $I_i = [x_i, x_{i+1}]$ of the interpolation range, S is a polynomial of degree at most three; denote this polynomial by s_i then we have

$$s(x) = s_i(x) \qquad x \in I_i,\ i = 0, 1, \ldots, N-1, \qquad \ldots (3.16)$$

A convenient formulation of s_i will be in terms of the distance of x from the two ends of the interval I_i, and so we define new variables u_i by

$$u_i = x - x_i \qquad \text{for } i = 0, 1, \ldots, N, \qquad \ldots (3.17)$$

Observe that $du_i / dx = 1$ for every i, and so differentiation or integration with respect to x and with respect to u_i will be equivalent. We denote the step lengths between the knots by

$$h_i = x_{i+1} - x_i = u_i - u_{i+1}, \qquad \ldots (3.18)$$

The conditions which must be satisfied are that s must interpolate f at $x_0, x_1, \ldots, x_N$ and s', s'' must be continuous at the interior knots $x_1, x_2, \ldots, x_{N-1}$. We will begin with the last of these conditions, the continuity of s''. On each of the intervals I_i, and so s'' is the first–degree polynomial s''_i. Let us denote its (as yet unknown) values at the knots by

$$s''(x_i) = A_i \qquad i = 0, 1, \ldots, N, \qquad \ldots (3.19)$$

It follows that $s''_i(x_i) = A_i$ and $s''_i(x_{i+1}) = A_{i+1}$, and since $s''_i(x_i)$ is a linear function, we have, for each i,

$$s''_i(x) = \frac{A_{i+1}(x - x_i) - A_i(x - x_{i+1})}{h_i} = \frac{A_{i+1}u_i - A_i u_{i+1}}{h_i}, \qquad \ldots (3.20)$$

We may integrate equation (3.20) twice to get

$$s_i(x) = \frac{A_{i+1}u_i^3 - A_i u_{i+1}^3}{6h_i} + cx + d, \qquad \ldots (3.21)$$

Where c and d are constants of integration. This can be conveniently written in the form

$$s_i(x) = \frac{A_{i+1}u_i^3 - A_i u_{i+1}^3}{6h_i} - B_i u_{i+1} + C_i u_i, \qquad \ldots (3.٢٢)$$

Consider first the interpolation condition at the point x_i, we have $u_i = 0$ and $u_{i+1} = -h_i$. Denoting $f(x_i)$ by f_i and substituting these values into (3.22), we get

$$f_i = \frac{A_i h_i^2}{6} + B_i h_i \qquad (i = 0, 1, \ldots, N-1), \qquad \ldots (3.٢٣)$$

Similarly at x_{i+1}, we have

$$f_{i+1} = \frac{A_{i+1} h_i^2}{6} + C_i h_i \qquad (i = 0, 1, \ldots, N-1), \qquad \ldots (3.24)$$

Solving these two for B_i and C_i yields:

$$B_i = \frac{f_i}{h_i} \square \frac{A_i h_i}{6}, \qquad \ldots (3.25\ a)$$

$$C_i = \frac{f_{i+1}}{h_i} - \frac{A_{i+1} h_i}{6}, \qquad \ldots (3.25\ b)$$

The final system of equations is derived from the first−derivative continuity condition. These equations are obtained by differentiating equation (3.22) with respect to x (remembering that differentiations with respect to x, or with respect to u_i or u_{i+1} are the same operation).

We obtain

$$s_i^{/}(x) = \frac{A_{i+1} u_i^2 - A_i u_{i+1}^2}{2h_i} - B_i + C_i, \qquad \ldots (3.26)$$

From which we may deduce that

$$s_i^{/}(x_i) = C_i - B_i - \frac{A_i h_i}{2}, \qquad \ldots (3.27)$$

and, similarly,

$$s_i^{/}(x_{i+1}) = C_i - B_i + \frac{A_{i+1} h_i}{2}, \qquad \ldots (3.28)$$

The continuity of s' will be guaranteed if, for every interior knot x_i, we have

$s'_i(x_i) = s'_{i-1}(x_i)$ which, on comparing (3.27) with (3.28) for i–1, yields the equation

$$\frac{(h_{i-1} + h_i)A_i}{2} + B_i - C_i - (B_{i-1} - C_{i-1}) = 0, \quad \text{for } i = 1, 2, \ldots, N-1, \qquad \ldots (3.29)$$

We can subtract the two equations (3.25 a and b) to obtain:

$$B_i - C_i = \frac{(A_{i+1} - A_i)h_i}{6} - \frac{f_{i+1} - f_i}{h_i}, \quad (i = 0, 1, \ldots, N-1), \quad \ldots (3.30)$$

The final term here is just the divided $f[x_i, x_{i+1}]$ which we will denoted by d_i. With this notation and substituting (3.30) for both i and i−1 into (3.29) we get

$$\frac{h_{i-1}A_{i-1}}{6} + \frac{(h_{i-1} + h_i)A_i}{3} + \frac{h_i A_{i+1}}{6} = d_i - d_{i-1}, \quad (i = 1, 2, \ldots, N-1) \qquad \ldots (3.31)$$

This is a system of N–1 equations with N−1 unknown A_i 's. As was commented above, there are many ways of using these two extra degree of freedom. One of the simplest ways is to simply set

$$A_0 = A_N = 0, \qquad \ldots (3.32)$$

Which gives rise to the so–called ***natural cubic splines*** [**Buc 92**].

3.2.4 Quadratic Interpolation

To define a ***Quadratic*** function, we need three data points. So each piece of the piecewise function will actually be defined over two consecutive data intervals. The following pints explains how to evaluate the piecewise quadratic interpolant at a point x [**Atk 03**].

1. Determine the **two consecutive intervals** that contain the position x.
2. Determine the parabola that passes through the three data points that define the intervals.
3. Evaluate that parabola at x.

Most data arise from graphs that are curved rather than straight. Assume that three data points (x_0, y_0), (x_1, y_1) and (x_2, y_2) are given with x_0, x_1, x_2 distinct points [**Atk 03**].

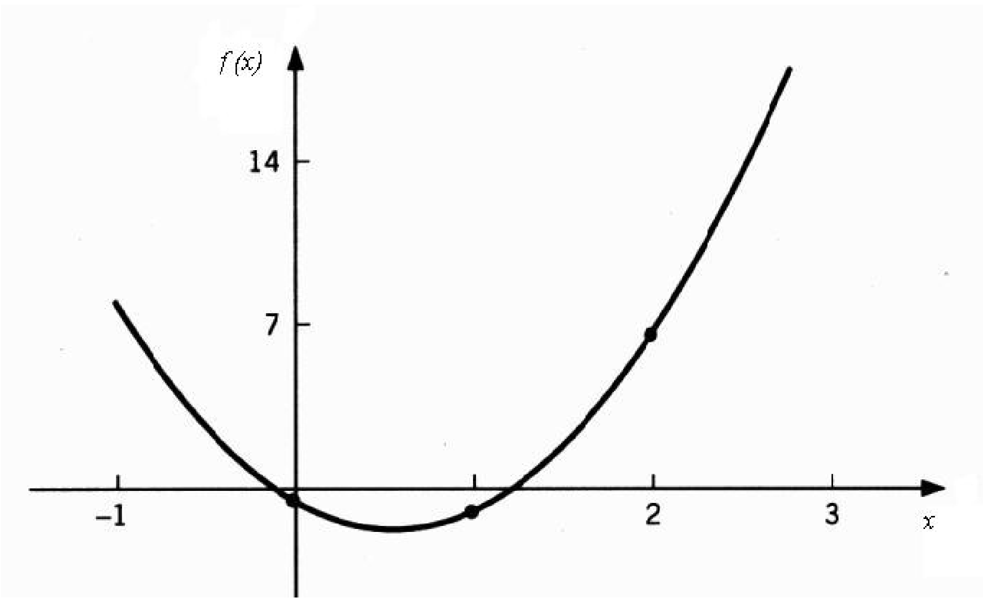


Figure (3.5) Quadratic interpolation

With linear interpolation, it was obvious that there was only one straight line passing through two given data points. But with three data points it is less obvious that there is only one quadratic interpolation whose graph pass through the points. It would be expected that a ***quadratic*** interpolation would yield better accuracy interpolation [**Atk 03**].

The given three points at a time construct an arc of a ***quadratic*** curve, perhaps parabola, circle or ellipse, to join them. Let us consider the three points A, B, C, with position vector n_1, n_2, n_3 respectively. ***Quadratic*** interpolation formula is written using shape functions, which vary according to the parameter values being used. Thus in general we have:

$$I(x) = M_1(d)\, n_1 + M_2(d)\, n_2 + M_3(d)\, n_3 , \qquad ..(3.34)$$

Where $M_1(d)$, $M_2(d)$ and $M_3(d)$ are shape functions. For ***quadratic*** interpolation the shape functions involve squared terms like d^2 and expressions like $d^2 + 3d + 2$, and thus the overall result is an arc of a ***quadratic*** curve fitting the points [**Ema 01**].

3.2.5 Bezier Interpolation

So far we have considered curve definitions that interpolate given data. Another approach is to provide a good smooth representation of a surface that approximates given data. In such a case there is no definable best fit, but the quality of a fit depends primarily on the designer's judgment. It is thus logical to use an interactive technique in which the user can experiment with a variety of shapes without having to know anything about the mathematical principles involved. However, certain smoothness condition should a priori be built into the class of curves the designer will experiment with. The most interesting approach probably being that developed by ***Bezier*** [**Wol 78**].

Bezier defines the curve p(u) in terms of the locations of n+1 control points p_i

$$P(u) = \sum_{i=0}^{n} p_i \, B_{i,n}(u) \,, \qquad \ldots (3.35)$$

Where $B_{i,n}(u)$ is a blending function

$$B_{i,n}(u) = C(n,i)\, u^i \,(1-u)^{n-i} \,, \qquad \ldots (3.36)$$

And C(n,i) is the binomial coefficient,

$$C(n,i) = n!/\,(i!(n-i)!) \,, \qquad \ldots (3.37)$$

The particular curve shown in figure (3.7) uses four control points, connected in the illustration to form an open polygon.

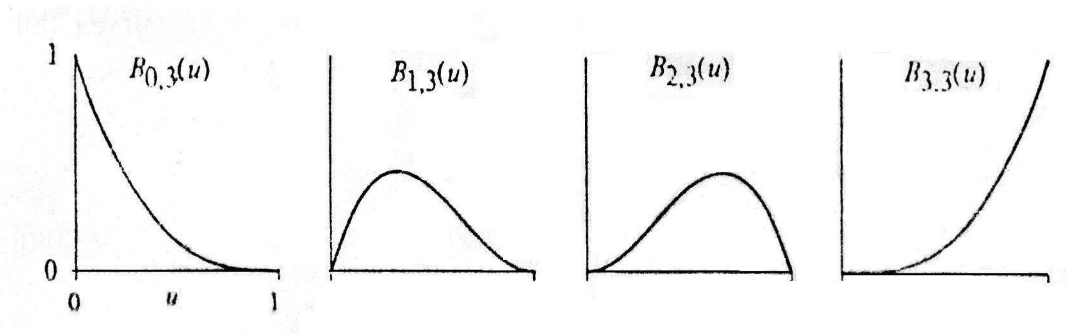


Figure (3.6) The four Bezier blending functions for n=3

The blending functions are the key to the behavior of Bezier curve. Figure (3.7) shows the four blending functions that correspond to a ***Bezier*** curve with four control points. These curves represent the influence that each control point exerts on the curve for various values of u. The first control point, p_0 corresponding to $B_{0,3}$, is most influential when u=0 in fact. Locations of all other control points are ignored when u=–0, because their blending functions are zero. The situation is symmetric for p_3 and u=1. The middle control points p_1 and p_2 are most influential when u=1/3 and 2/3, respectively [**New 79**].

In ***Bezier*** curve generally only the first and last control points are interpolated. The intermediate control points influence the curve's shape in a different way, acting more like magnets. There are various ways to adjust the influence of the control points. One could repeat some points, i.e., list them more than once, but increasing the number of points also increases the degree of the resulting curve. Another restriction inherent to the ***Bezier*** approach is the fact that the curves change totally as soon as one control point is moved [**Mul 00**].

3.3 Requantization

Quantization is the step which allows a continuous amplitude signal to be represented in the discrete amplitude increments available in a digital computer this is performed by an ADC, Which takes as input a constant analogue voltage (performed by the sampler) and generates a corresponding binary value as output [**Embree 91**].

Signal requantization is applied in digital audio systems whenever the word−length of audio samples needs to be reduced. This is the case for instance when an audio signal has to be stored on a CD and was originally produced at the output of a digital audio system that operates with more than 16 bit precision. In some applications, like multimedia, gaming, or mobile communication devices, requantization to 8 bit or 12 bit could be an economically interesting alternative to other forms of data compression because requantized data can be send directly to the ADC converter, while encoded data requires a decoder. Signal requantization inevitably introduces an error, which can cause two types of audible problems. The first is a background noise that may be audible by itself. It can usually occur when (part) of the error signal is uncorrelated with the original audio. When the error is correlated with the signal, linear or nonlinear distortions may cause alterations in the perceived quality of the signal itself. At low signal levels, this second problem is usually much more serious. Dither means how well does it remove quantization distortion whenever there's some requantization going on, small noise can be used to remove the correlation between the error and the signal at the expense of increased noise energy [**Kon 03**].

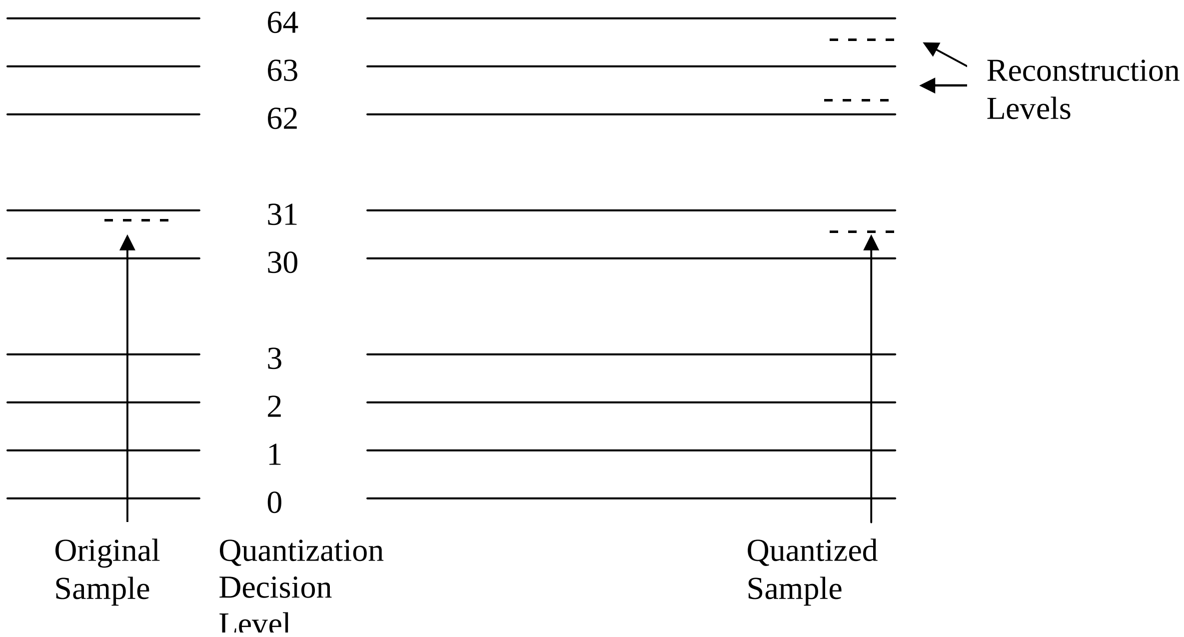


Figure (3.7) Quantization operation

3.3.1 Uniform Quantization

Uniform quantization is the most commonly used technique for digital signal representation. The goal of the quantizer is to provide minimum possible average distortion to its input under some constraint. The quantizer output signal, which indicates the minimum amount of information needed to reconstruct the output, is generally used as a constraint. The simplest quantization correspondence is uniform quantization, where the amplitude range is split into equal regions by points termed quantization levels, and the output is a binary representation of the nearest quantization level to the input voltage. An example of a 1-dimensional uniform quantization is shown in figure (3.9):

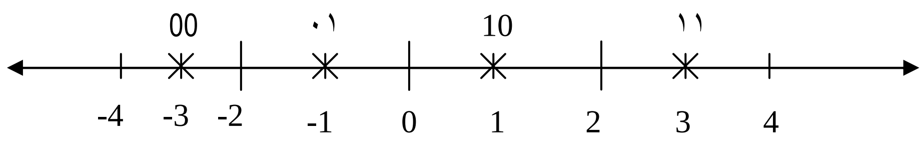


Figure (3.8) Uniform quantization

Here, every number less than -2 is approximated by 00. Every number between -2 and 0 are approximated by 01. Every number between 0 and 2 are approximated by 10. Every number greater than 2 is approximated by 11.

The amplitudes of the samples are quantized by dividing entire amplitude range into a finite set of amplitude ranges and assigning the same amplitude value to all samples falling in a given range. This is shown in figure (3.10) for an 8-level quantizer. For all values of x (n) between x_1 and x_2 the output of the quantizer is q (n) = Q [x (n)] = q_2. each of the quantizer level is labeled with a 3–bit binary codeword which serves as a symbolic representation of that amplitude level [**Wit 82**].

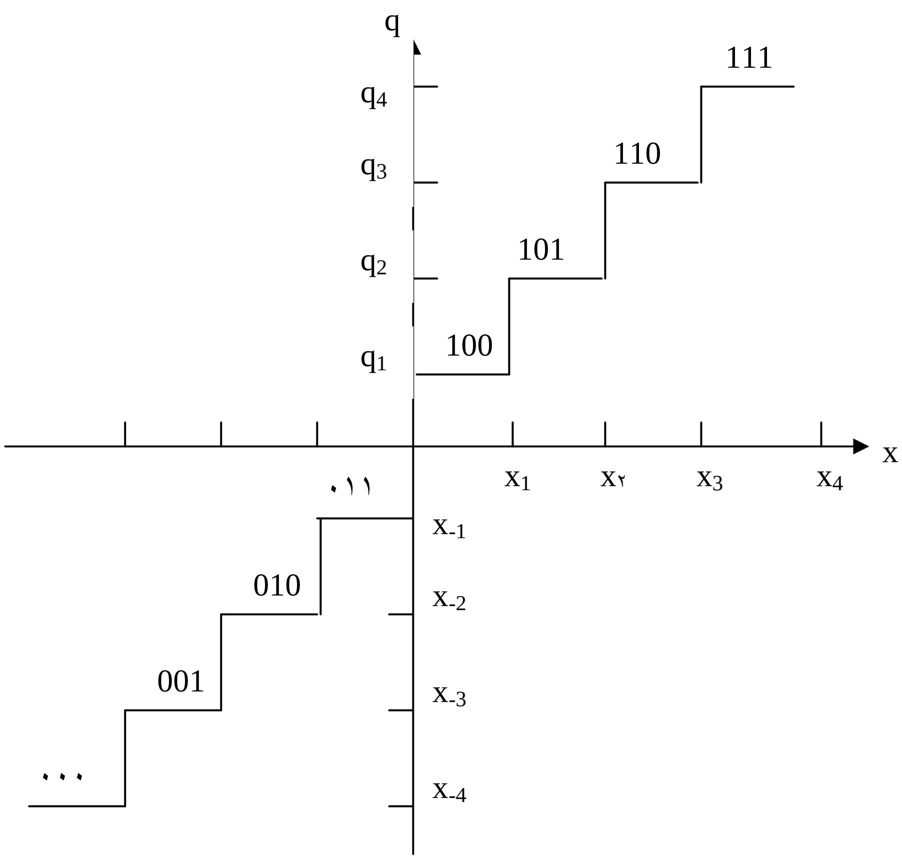


Figure (3.9) Input–output characteristic of a 3–bit quantizer

3.3.2 Non–Uniform Quantization

When the input source signal is uniformly distributed all the quantization intervals are of the same width, i.e, the source does not prefer any particular quantization interval. This may not be true in general for a source with an arbitrary distribution of values. In this general case it would make more sense to assign more levels in the ranges of values that occur more often and fewer quantization levels to ranges that are infrequent. This type of quantization is referred to as **non–uniform** quantization.

There are two advantages to using non-uniform spacing of quantization levels. First, it is possible to significantly increase the dynamic range that can be accommodated for a given number of bits of resolution by using a suitably chosen non-uniform quantizer. Second, it is possible to design a quantizer tailored to the specific input statistics so that it is considerably superior, in terms of (SNR) levels, compared to the uniform quantization case [**Gar 02**].

It is sufficient to apply histogram equalization method. The first step in this method is to find the accumulated probability density:

$$P_{acm}(i) = \frac{\sum_{j=0}^{i} \mathrm{H}\,(\mathrm{j})}{\sum_{j=0}^{255} \mathrm{H}\,(\mathrm{j})}, \qquad \ldots (3.38)$$

Where H (j) is the histogram value of the j th level of the audio signal, $P_{acm}(i)$ is the accumulated probability of the i th level. Then the requantized value $(i^{/})$ of the level (i) is determined from follows:

$$i^{/} = \mathrm{P}_{acm}(\mathrm{i}) \times \mathrm{N}, \qquad \ldots (3.39)$$

Where $i^{/}$ the requantized signal and N is is the total number of quantized levels.

Examination Committee Certificate

We certify that we have read the thesis entitled
"Effect of Resampling and Requantization on the Compression of Digital Audio Data"
and as Examining Committee, examined the student
SHEREEN ABDUAL QADIR AL-Samara' i
in its contents and what is related to it, and that in our opinion it is adequate as standard of thesis, with ***Very Good*** standing of Degree of Master of Science
in ***Physics***

Signature:
Name: **Dr. Kais J. AsL-Jumaily**
Title: Assist Professor (Chairman)
Address: Al-Mustansiriyah University
Date: / /2005

Signature:
Name: **Dr. A. R. T. Ziboon**
Title: Assist Professor (Member)
Address: Technology University
Date: / /2005

Signature:
Name: **Dr. Salah A. H. Saleh**
Title: Assist Professor (Member)
Address: Al-Nahrain University
Date: / /2005

Signature:
Name: **Dr. Laith A. Al-Ani**
Title: Assist Professor (Supervissor)
Address: Al- Nahrain University
Date: / /2005

Signature:
Name: **Dr. Loay E. George**
Title: Assist Professor (Supervisor)
Address: Ministry of Higher Education
Date: / /2005

Approved by the University Committee of Postgraduate Studies

Signature:
Name: **Dr. Laith A. Al-Ani**
(Dean of the college of science)
Date: / /2005

Dedicated

To

My Parents

And

Brother and Sisters

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

نَرفَعُ دَرَجَتٍ مَّن نَّشَآءُ وَفَوقَ كُلِّ ذِى

عِلمٍ عَلِيمُ

صَدَقَ اللَّهِ العَظِيمْ

(يُوسُفَ :٧٦)

الخلاصة

ان دراسة اعادة الاعتيان والتكميم للبيانات الصوتية الرقمية تعتبر من المواضيع المهمة حيث تستخدم لاغراض ضغط البيانات.
في هذا البحث تم دراسة بعض طرق اعادة الاعتيان بواسطة تقليل عدد العينات مع الحفاظ على نوعية الصوت. ومن هذه الطرق المدروسة:

‘‘ Linear, Lagrange , Cubic Spline , quadratic, and Bezier’’

وتم بحث معدل تقليل العينات بواسطة حذف جزء كبير من العينات ومن ثم اعادة العينات التي تم حذفها بواسطة طرق الاستكمال التي تم الاشارة اليها وتم تحديد درجة كفاءة كل طريقة من الطرق المدروسة باستخدام مقاييس معيارية منها نسبة تقييس الاشارة العظمى الى الضوضاء (Peak Signal to Noise Ratio) واثبتت طرق Lagrange, Cubic spline, and Bezier انها من افضل الطرق وتعطي نفس النتائج . كذلك تم في هذا البحث دراسة اعادة التكميم بواسطة الطريقة المتجانسة وغير المتجانسة وتم دراسة مراحل التكميم لكل طريقة مع الحفاظ على نوعية الصوت وكانت نتائج التكميم للطريقة المتجانسة افضل من الطريقة الغير متجانسة.بالاضافة الى ذلك فقد تم اختبار النتائج سمعياً وكانت عينة المستمعين من خلفيات مختلفة وتم استنتاج انه كلما زاد معدل تقليل البيانات ومراحل التكميم سوف تقل نوعية الصوت.

Chapter One General Introduction

Chapter Two Digital Audio Processing

Chapter Three Resampling and Requantization

Chapter Four Experimenal Results

Chapter Five Conclusion and Future Work

References

References

[**Atk 83**]

L. V. Atkinson & P. J. Harley, ‘‘**An Introduction to Numerical Methods with Pascal**’’, Library of Congress Cataloging in Publication Data, 1983.

[**Atk 03**]

S. Atkinson, ‘‘**Piecewise Polynomial Interpolation**’’,
http://www.math.Pitt.edu/~ sussmanm/2070 Fall02/lab-08, 2003.

[**Buc 92**]

J. L. Buchanan & P. R. Turner, ‘‘**Numerical Methods and analysis**’’, McGraw–Hill, Inc., 1992.

[**Bur 85**]

R. L. Burden & J. D. Faires, ‘‘**Numerical Analysis**’’, PWS Publishers, 1985.

[**Cav 00**]

T. J. Cavicchi, ‘‘**Digital Signal Processing**’’, John Wiley & Sons, Inc, 2000.

[**Col 98**]

N. Collins, ‘‘**An Interface to Low Level Digital Audio**’’, http://www.cus.cam.ac.uk/~nc272/papers/pdfs, 1998.

[**Con 00**]

M. Concrete, ‘‘**Coding of Multimedia Signals**’’, http://www.ece.ucdavis.edu/~mihaela, 2000.

[**Dou 87**]

O. Douglas, ‘‘**Speech Communication, Human and Machines**’’, Wesley series, 1987.

[**Ema 01**]

K. A. Emad, ‘‘**Analytical Studies for Geometric Transformation Methods**’’, M. Sc. Baghdad University, 2001.

[**Emb 91**]

P. M. Embree, ‘‘ **Digital Signal Processing**’’, Prentice–Hall, Inc., 1991.

[**Fer 81**]

J. H. Ferziger, ‘‘**Numerical Methods for Engineering Application**’’, Library of Cataloging in Publication Data, 1981.

[**Gar 02**]

V. Garousi , ‘‘**Study on effects of Different Quantization Schemes in the NAP Turbo Decoding Algorithm**’’, http://www.cst.uwaterloo.ca/~garousi, 2002.

[**Gom 97**]

J. Gomes & L. Velho, ‘‘**Digital Signal Processing** ’’, Springer–Verlag New York, Inc., 1997.

[**Has 01**]

J. Hass, ‘‘**Principles of Digital Audio**’’, http://www.indiana.edu /~emusic/ digital_audio.htm, 2001.

[**Jut 94**]
D. Jutta, ‘‘**Digital Speech Compression**’’, IEEE, Signal Processing Society, PP. 30, 1994.

[**Ker 86**]
I. V. Kerlow & J. Rosebush, ‘‘**Computer Graphics for Designers and Artists**’’, Van Nostrand Reinhold Company, 1986.

[**Kie 98**]
T. Kientzle, ‘‘**A Programmer's Guide to sound**’’, An Imprint of Addison Wesley Longman Inc., 1998.

[**Kon 03**]
D. D. Koning & Verhelst, ‘‘**Least Squares Noise Sharping for Audio Requantization**’’,
http://www.etro.vub.ac.be/research/DSSP/publications/int_conf/ICASSP, 2003.

[**Mcc 98**]
J. H. McClellan & R. W. Schafer, ‘‘**A Multimedia Approach**’’, Library of Congress Cataloging in Publication Data, 1998.

[**Mcl 79**]
D. H. Mclain, ‘‘**One Dimensional interpolation from Random Data**’’, Computer Journal, Vol. 19, No. 2, 1979.

[**Mul 00**]
C. Mulcahy, ‘‘**The Basic Curves and Surfaces of Computer Aided Geometric Design**’’, http:// www.moshplant.com/direct-or/bezier, 2000.

[**New 79**]

W. M. Newman & R. F. Sproull, ‘‘**Principle of Interactive Computer Graphics**’’, McGraw–Hill, Inc., 1979.

[**Opp 75**]

A. V. Oppenheim & R. W. Schafer, ‘‘**Digital Signal Processing**’’, Prentice–Hall, International, Inc., London, 1975.

[**Rab 78**]

L. R. Rabiner & R. W. Schafer, ‘‘**Digital Processing of Speech Signals**’’, Prentice–Hall, Inc., Englewood Cliffs, New Jersey, 1978.

[**Ron 02**]

G. Ron, ‘‘**Lagrange Interpolation** ’’, http://www.nd.edu, 2002.

[**Sal 98**]

D. Salomon, ‘‘**Data Compression**’’, Springer–Verlag New York, Inc., 1998.

[**Sco 95**]

J. Scott, ‘‘**Visual Basic Multimedia’’**, Adventure Set, An Imprint of Coriolis Group Inc., 1995.

[**Shy 00**]

M. Shykula & O. Seleznjev, ‘‘**Uniform quantization of Random Processes**’’,

http://www. matstat.umu.se/personal/Oleg/Personal/Quantization.pdf, 2000.

[**Sim 04**]

J. O. Simth, ‘‘**What is Bandlimited Interpolation**’’,
http://www.ccrma.Stanford.edu, 2004.

[**Sta 70**]

P. A. Stark, ‘‘**Introduction to Numerical Methods**’’, Collier–Macmillan Canada, 1970.

[**Sto 93**]

A. Stolz, ‘‘**The Sound Blaster Book**’’, Abacus, USA, 1993.

[**Vet 00**]

R. Vetter & J. Kraus, ‘‘**Robust Speech Recognition using Missing Feature Theory and Vector Quantization**’’,
http://www. www.tsp.ece.mcgill.ca, 2000.

[**Wit 82**]

I. H. Witten, ‘‘**Principles of Computer Speech**’’, Academic Press Inc. Ltd, 1982.

[**Wol 78**]

K. G. Wolfgang, ‘‘**Interactive Computer Graphics**’’, Prentice–Hall, Inc., Englewood Cliffs, New Jersey, 1978.

[**Web 03**]

T. J. Weber, ‘‘**The Wave file Format**’’, http:// www.ora.com/centers, 2003.

[**Wil 03**]

J. Wiliams, ‘‘**Wave PCM Sound File Format**’’,

http://www.Standford.edu/ccrma/courses/422/project/wavefornat, 2003.

جمهورية العراق

جامعة النهرين

كلية العلوم

تأثيراعادة الاعتيان والتكميم على ضغط البيانات الصوتية الرقمية

رسالة

مقدمة إلى كلية العلوم، جامعة النهرين كجزء من متطلبات نيل شهادة الماجستير في علوم الفيزياء

من قبل

شيرين عبد القادر مهدي السامرائي

بكالوريوس ٢٠٠٢

المشرفون

د. ليث عبد العزيز العاني

د. لؤي ادور جورج

ربيع الاول

١٤٢٦

نيسان

٢٠٠٥