The Suitability of the Karhunen-Lòeve Transform, For Real Time Multichannel Audio Data Reduction in Computer Game Systems.

Chris Green

Final Year Tonmeister Technical Project

March 2007

Abstract

The following project concerns itself with a real time use of the Karhunen-Lòeve Transform, specialising in computer game applications. A preliminary review of surround sound audio in conducted, concentrating on the heightened envelopmental qualities it produces. The information of this research is then related to computer game audio, and its requirements. The theory of the KLT is then introduced, along with its applications within the area of audio data reduction. Relevant previous systems are then reviewed in order to create main hypotheses regarding the suitability of the KLT to produce a real time multichannel audio decoder within a computer game system. The hypothesized system is then developed in a C++ programming environment. The produced system is then investigated through the use of performance and subjective based testing. The results gathered clearly show that the KLT is highly situated to real time decoding, and that further optimization of the produced program could have successfully implementation into computer game multichannel streaming systems. A summary of the results is then made and conclusion based around it. Finally further research and development within the area is discussed in order to further help proceeding researchers.

Table of Contents

ABSTRACT I
TABLE OF CONTENTS II
ACKNOWLEDGMENTS V

1. INTRODUCTION 1
  1.1 Overview of Project Outline 1
  1.2 Introduction Summary 2

2. SURROUND SOUND FUNDAMENTALS 3
  2.1 Initial Surround Sound Systems 3
  2.2 5.1 Surround Sound 5
  2.3 Summary 7

3. SURROUND SOUND IN COMPUTER GAMES 8
  3.1 The Need for Surround Sound 8
  3.2 Surround Sound Implementations 9
    3.2.1 Dolby Surround 9
    3.2.2 Dolby Pro Logic 11
    3.2.3 Further Developments 14
  3.3 Summary 16

4. MATHEMATICAL BACKGROUND OF THE KARHUNEN-LÒEVE TRANSFORM 17
  4.1 Arithmetic Mean 17
  4.2 Variance and Standard Deviation 17
  4.3 Covariance 18
  4.4 Correlation 19
  4.5 The Covariance and Correlation Matrices 20
  4.6 Eigenvectors and Eigenvalues 21
  4.7 Summary 21

5. PRINCIPLE COMPONENT ANALYSIS AND THE KARHUNEN-LÒEVE TRANSFORM 22
  5.1 Scatter Plots 22
  5.2 The Scree Plot 24
  5.3 Audio Applications Incorporating KLT 25
    5.3.1 Interchannel Redundancy Removal within Multichannel Audio 26
  5.4 Summary 27

6. THE USE OF REAL TIME KLT DECODING WITHIN A COMPUTER GAME 28
6.1 Justification of Hypotheses 28
6.2 The Premise 29
6.3 What IS and what is NOT known? 30
6.4 Hypotheses 30

7. DEVELOPMENTAL SECTION 31
7.1 Choice of Integrated Development Environment 31
7.2 Choice of Channel Amount in Audio Data Stream 33
7.3 Choice of Audio Excerpts 33
7.4 KLT Encoding of Audio Data Stream 34
7.4.1 Temporal-Adaptive Window Length 34
7.4.2 Binary Storage of Inverse-KLT Metadata 35
7.5 The Real Time Decoding System 35
7.5.1 Toolbox – C++ Libraries 35
7.5.2 Main Decoding System 36
7.6 Developmental System Testing 38
7.6.1 System Resource Test 39
7.6.2 Overall Latency Test 39
7.7 Summary 40

8. DEVELOPMENTAL RESULTS AND ANALYSIS 41
8.1 Summary 44

9. EXPERIMENTAL SECTION 45
9.1 Aspects Taken From Development 45
9.2 Choice of Testing Computer Game Consoles 45
9.3 The "Test Level" 46
9.4 What to Listen For? 47
9.5 The Listening Environment 47
9.6 Conducting the Experiment 48
9.7 Summary 49

10. EXPERIMENTAL RESULTS AND ANALYSIS 50

11. DISCUSSION AND CONCLUSION 53
11.1 Discussion of Results 53
11.2 Suggestions for Further Research and Development 53

REFERENCES 55

APPENDIX A: LISTENING TEST INSTRUCTIONS...58

APPENDIX B: MATLAB CODE FOR AN OFFLINE TEMPORAL-ADAPTIVE KLT MULTICHANNEL ENCODING ALGORITHM. 59

Acknowledgments

I would like to thank a lot of people with this section, but I guess I will have to keep it short. First I would like to thank the people at Electonic Arts for teaching me how to code, and therefore do pretty much everything this project. I would like to thank Lars Henning and Yu Jiao, “Joey” for their amazing expertise into the Karhunen-Lòeve Transform, help that this paper could not have been completed without. My final thanks goes to Pip, and my course mates for their constant support in the late night library sessions.

Thanks again.

1. Introduction

In the modern audio consumer market, surround sound has become widely used and available. Full 5.1 surround configurations can be brought for as little as £150, and 5.1 integration within new television systems is becoming more commonplace. Also within the same time scale the technology incorporated within computer game consoles has grown exponentially. With both these facts considered one could easily assume that the computer game technologies would be taking advantage of the 5.1 surround sound system. Though at the time of writing, the author could only research two systems that could playback 5.1 multichannel audio, and none that had the ability to stream it within the context of a game. With this taken into account it seems paramount that an encoding and decoding method is created that can facilitate 5.1 multichannel audio streams within a game.

This project attempts to develop an offline encoding, real time decoding system that satisfies the above need. To aid the flow and coherence of the project a layout overview is given below. The overview also contains brief summaries of each of the eleven chapters, and information about how they relate to each other.

1.1 Overview of Project Outline

In ***Chapter 2*** the initial development of surround sound systems is discussed and the widely used ITU-R BS. 775 5.1 standard introduced. The heightened envelopmental quality of these systems is then related to surround sound's use within computer games in ***Chapter 3***. This chapter then goes on further to argue the use of surround sound within computer games, by discussing its current implementations and limitations. During this, the encoding methods of Dolby Surround and Dolby Pro Logic are also discussed, due to their current use within computer game surround sound.

Chapter 4 goes on to give an overview of the mathematical techniques relevant to the Karhunen-Lòeve Transform (KLT) method of multichannel audio data reduction. This chapter then leads onto ***Chapter 5*** in which the KLT method is described, and its application within audio data reduction discussed.

In ***Chapter 6*** the above chapters are summarised and incorporated into hypotheses based on, the use of the KLT as a multichannel data reduction method for use within computer game programs.

Chapter 7 concerns itself with development and testing of the hypothesised systems, and describes the offline encoding and real time decoding programs that were produced. Within this chapter there is also a discussion into the choice of software development environment and the audio excerpts used. The system performance is then tested and results are shown in ***Chapter 8*** along with their analyses.

In ***Chapter 9*** the experimental research is further expanded with the use of a listening test. The listening test is based around the use of a "test level" computer game. For the remainder of this chapter the implementation of this "test level" is discussed, and the results shown and analysed in ***Chapter 10***.

From this development and experimentation a conclusion is created based on the original hypotheses. This conclusion is then stated and discussed in ***Chapter 11***, along with a summary of the two preceding stages. Finally at the end at this chapter further research and development within the area are deduced.

1.2 Introduction Summary

The project has been introduced along with the overall layout. Included with this was a small summary into each of the chapter that was presented consecutively. Within the project it is going to be investigated whether the Karhunen-Lòeve Transform will be suitable for an offline encoding, real time decoding system, and if this system could be incorporated within a computer game.

The first chapter that is to follow will start the process of reviewing the existing literature and discuss the fundamentals of surround sound.

2. Surround Sound Fundamentals

As mentioned in the introduction, this paper deals very closely with surround sound technologies and applications. Therefore it is the author's belief that in order to understand the proceeding discussions, it is important to first comprehend the basic background and ideologies of surround sound. Especially relevant is the necessity for a system with a greater perceived envelopment (referred to in later chapters). The following chapter and its subsections will discuss the development of surround sound resulting in the modern day use of the 3:2 ITU-R BS.775 Standard of 5.1 surround.

2.1 Initial Surround Sound Systems

Surround sound was first introduced to further the reality of previous two channel stereophony, and to enhance the cinema sound experience due to an increasing competition from home television systems [Rumsey 2007]. The system that was devised produced this effect by placing speakers both in front and behind the listener (***surrounding*** the listener), and therefore creating a more realistic auditory model of sound in a real space [Griesinger 1998]. While traditional two speaker stereophonic systems can create an accurate frontal image [Rumsey 2005] which is adequate for positioning sources in the stereo image, they have severe limitations in the areas of spaciousness and envelopment [Theile 1991]. This is due to the fact that the auditory cues for these two aspects of human hearing are based both in frontal and rear listening spaces [Howard 1996]. Also it is important to note that when listening to a two channel stereo playback system, the surround envelopment cues reach your ears through interaction with the listening environment [Griesinger 1997]. This causes the cues to contain information about the envelopment of the listening space and not the original source space, therefore resulting in a miss-match between perceived auditory and visual experiences.

The initial systems introduced by Warner and 20th Century Fox in the 1950's [Holman 2001] incorporated the use of one back or surround speaker which was used for surround content, and a further front speaker for centre image reinforcement in the frontal image (fig. 2.1). This is known as a 3:1 system (3 front speakers/channels and 1 surround speaker/channel).

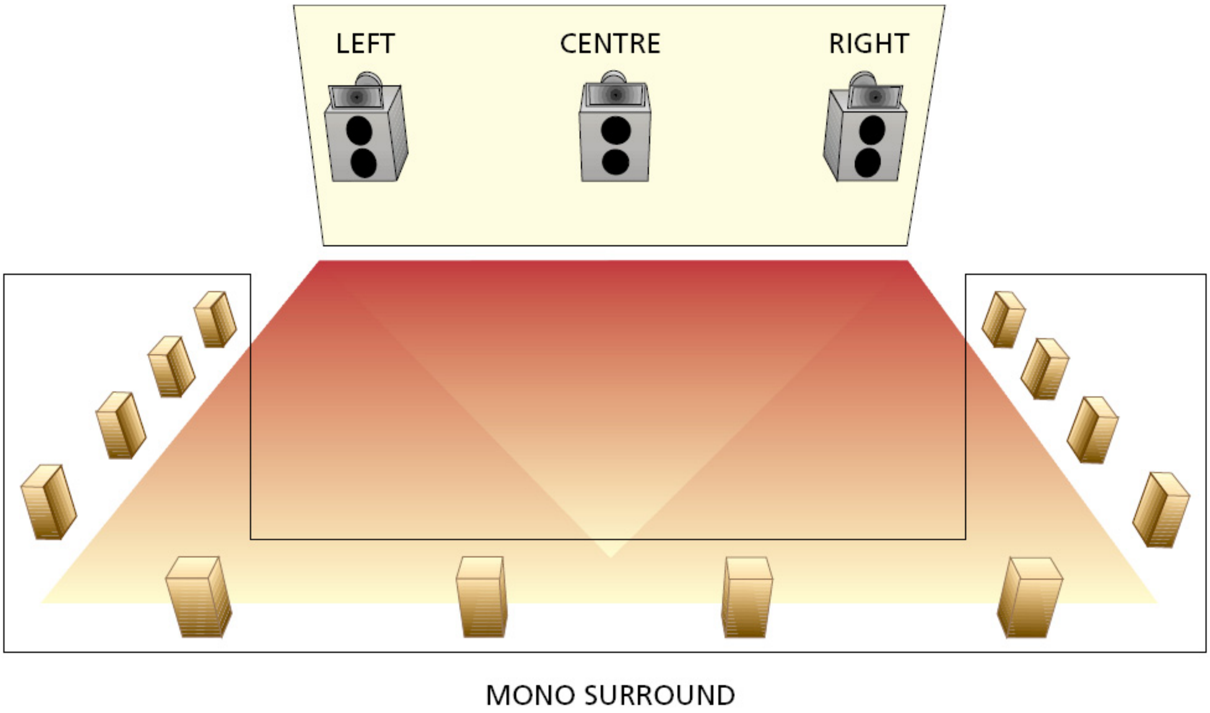

Figure 2.1: Graphical representation of a 3:1 surround system [Dressler 1997]

The systems marked a large step forward in the quality of envelopment due to the source material now being played back in both front and rear listening areas. This same 3:1 system was later adopted by Dolby and used in Dolby Stereo/Surround and Dolby Pro Logic. These systems will be discussed in more detail later in this paper, due to their use for surround content in current generation computer game consoles. While the added surround speaker increased the perception of envelopment over the traditional two speaker arrangement, the fundamental limitation was the fact that the surround content was monophonic. Woszczyk showed that this gave inadequate auditory envelopment and perceived spaciousness [Woszczyk 1993]. The next section of this paper will discuss the further development into extra surround channels to create a separate surround image to combat this restriction.

2.2 5.1 Surround Sound

In order to create a more enveloping and spacious audio experience, the use of two surround speakers in combination with three frontal speakers was decided upon. This 3:2 system quickly became widely adopted and is now the established modern surround standard for cinema, as well as television and home audio products [Rumsey 2005]. The standard is described in detail and specified by the ITU-R BS.775 Recommendation, which is shown below in Fig 2.2.

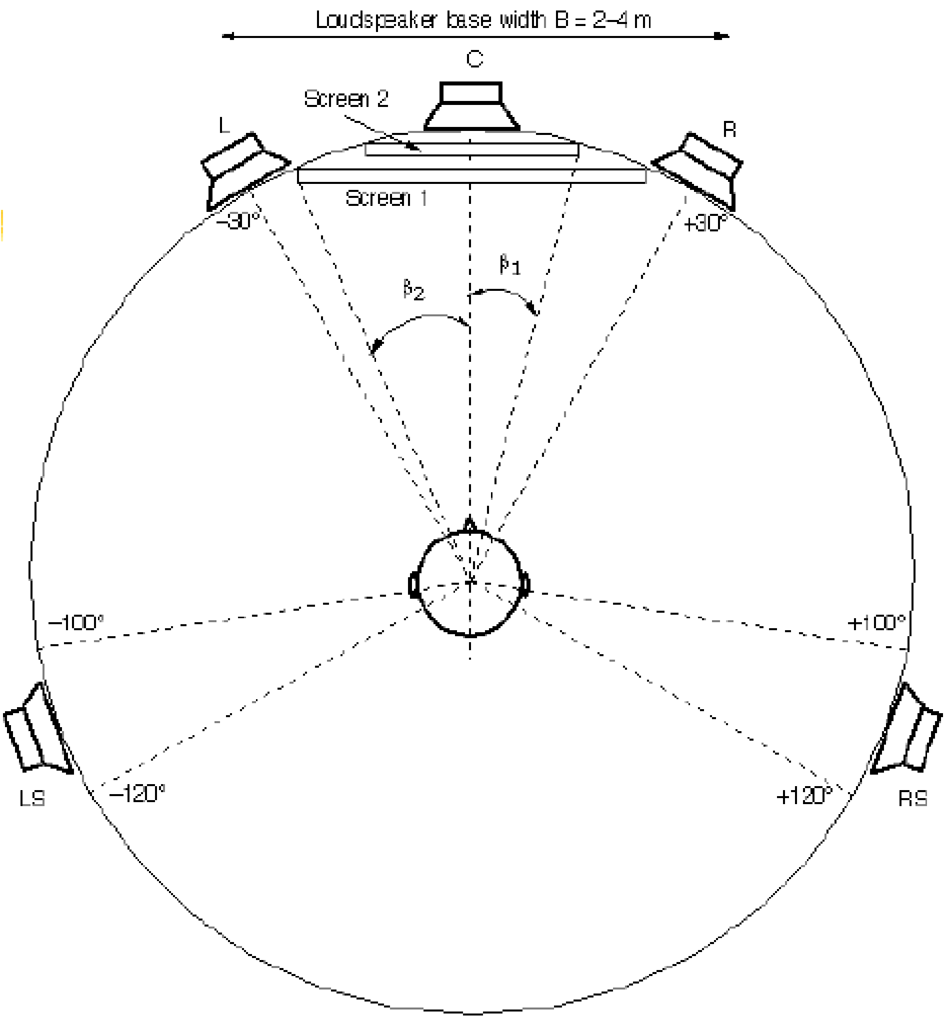

Figure2.2: Diagram of ITU-R BS.775 Standard [ITU-R 1993]

As with the previous 3:1 system the three front channels contain frontal stereo image and positional cues, with the two surround channels creating the perception of envelopment and rear spatial impression [Henning 2006]. The surround elements of the BS.775 5.1 system though are a compromise between rear spatial imaging and surround envelopment. This is because they are located approximately mid way between 90^o, which is the optimal angle for envelopment quality, and 135^o which is most favourable for rear localisation [Nakayama 1971]. The compromise causes the 5.1 system to have poor side imaging over a full bandwidth frequency range, with frequencies separating if a full bandwidth signal is panned between front and rear speakers [Holman 2001].

The .1 stated in the 5.1 title refers to the low frequency channel which has become known as the low frequency effects channel (LFE), due to its use to add low frequency rumble and bass hits to effects [Rumsey 2007]. The LFE was originally added to the system in order to increase the bass headroom. The need for this being that the human hearing system is less sensitive at low frequencies, and therefore needs an increased amount of decibels to create the same perceived level as the higher frequencies [Howard 1996]. The addition of the LFE means that the five main program speakers will not overload during the increased intensity bass content. Overall the inclusion of the LFE increases the bass headroom by 10dB. For the purpose of this research the need of the LFE channel is unnecessary considering its use for sound effects and low frequency content. It is for this reason that it is not included in any of the proceeding discussions and that all audio stream data reduction experiments are involving five streams.

It has been shown by Blauert that the correlation between each channel, especially the front and surround sets, controls the amount of experienced envelopment and spaciousness within the program material [Blauert 1997]. It should be noted that it is this correlation which is exploited and used for the basis of the Karhunen-Lòeve Transform method of data reduction, which is mentioned in the introduction and researched in later sections of this paper.

2.3 Summary

It can be concluded that the main motivation for the research and development of surround sound systems, is to create a more enveloping and spacious audio configuration, which will result in a more involving experience and closer representation to a real sound space. It has been shown though that the 5.1 system is a compromise especially in the surround speakers and it has been concluded at various AES conventions [Bosi 2000] that additional surround speakers would produce a higher quality performance. The ITU-R BS.775 Recommendation though is widely regarded as the standard surround sound configuration and will be used for this paper's research and experimentation into multichannel audio.

The next section will review the current literature associated with computer game surround audio, while relating the research back to the findings stated here.

3. Surround Sound in Computer Games

Since the first computer game Spacewar in the 1960's [Russell 1960], computer games in partnership with their relevant technologies have been rapidly developing into a more involving experience. Along with this and an elevated availability of high quality surround sound systems in the home, the need for surround sound, and high quality audio within computer games has risen exponentially.

3.1 The Need for Surround Sound

The importance of surround sound within a computer game is routed in the involvement, and envelopment that the computer game creates towards the player. The game models another reality outside of the real world that the player's consciousness is drawn into. When considering the role audio plays, be it sub-consciously or consciously, it is important to first judge the perceptual effect of the audio on the player. It has been shown by Zieliński et al. that the person participating in the computer game becomes less receptive to changes in the quality of the audio accompanying the game [Zieliński 2003]. This research though was made using temporally static quality degradations rather than temporally dynamic ones. The importance of this is that research by Mued et al, shows that time or syncing differences would have resulted in much higher receptivity [Mued 2003], due to the human brain being very sensitive in situations where auditory and visual events are unsynchronised. With this taken into account it can be shown that the most important feature is that the audio mirrors the visual cues, in both temporal and environmental aspects. Therefore by referring to the surround sections earlier in this paper, it can be deduced that the heightened envelopment quality present in surround sound systems would be of high importance to the area of computer game audio.

3.2 Surround Sound Implementations

The limitations of current generation games consoles mean that full five channel surround sound streaming is impossible within the context of a real time game [Morris 2007]. This may seem an implausible statement due to the fact that a Digital Versatile Disk (DVD) can stream high quality video and 5.1 audio content simultaneously, but this is in a linear system which only needs to stream this fixed data. In a computer game application a DVD can be expected to stream 3D model meshes, environmental textures, speech audio, surround music and ambiences, in a random access means [Morris 2007]. Therefore it is unrealistic to stream all needed audio data within the approximately 5.28 MB/s (megabytes per second, based on Sony Playstation 2 DVD x4 speed drive) data rate available for all streamed data [Blu Ray Founders 2004]. The only solution is that any surround sound information needs to be encoded before transmission. In current generation systems this encoding is applied in pre-processing, due to real time processing restrictions of the relevant consoles. The encoded data in then read out and decoded by external hardware, again releasing the gaming console from any real time CPU decoding processes. The encoding schemes that are in use at the time of writing this paper are Dolby Pro Logic I and II. Various experimental systems such as Format B Ambisonics and MPEG-2 have been tested by Electronic Arts [Morris 2007], but neither of these reached the consumer market. For this reason the following section of this paper will concern itsself with the two end user encoding systems of Dolby Pro Logic.

3.2.1 Dolby Surround

The Dolby Stereo system was developed in 1976 by Dolby Laboratories as a method for transmitting four channels of surround data over two channels of analogue audio. It was created for use with 35mm motion picture prints and became a world leader in cinema surround sound [Dressler 1997]. The emulation of this system for the home environment is referred to as Dolby Surround and was introduced in 1982. The relevance of this encoding method within the context of this paper is that it was later evolved by Dolby into Dolby Pro Logic.

Dolby Surround uses a matrix encoding method known as the Dolby Motion Picture (MP) Matrix. The Matrix is fed four input signals that represent left (L), right (R), centre (C) and surround (S) channels, and produces two output signals, left total (L_T) and right total (R_T).

The main L and R signals are fed straight to the L_T and R_T outputs, with the C channel being divided evenly between the outputs with a 3dB reduction. The surround channel experiences bandwidth limiting from 100Hz to 7KHz, and is encoded with a modified Dolby B-Type noise reduction technique. After these processes a +90° phase shift is applied to the signal summed with the L_T and a -90° phase shift to the one summed with the R_T, thus creating an 180° phase difference between the signal component present in both outputs [Dressler 1997]. This is shown diagrammatically in figure 3.1 below.

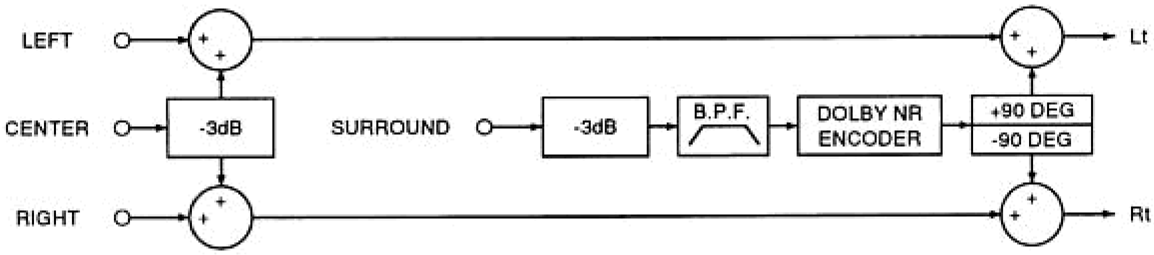

Figure 3.1: Diagram of Dolby MP Encoder [Dressler 1997].

The Dolby MP Matrix system theoretical results in no loss of separation between the centre and surround signals. This can be mathematically proven; because the surround channel is derived from the difference between the L_T and R_T outputs, the centre signal that is identically present in both signals will be cancelled out in the recovered surround signal. The problem is that if during transmission the amplitude and phase characteristics change between the two output signals (L_T and R_T) the centre channel information will no longer be cancelled. This will result in undesired crosstalk between the centre and surround channels.

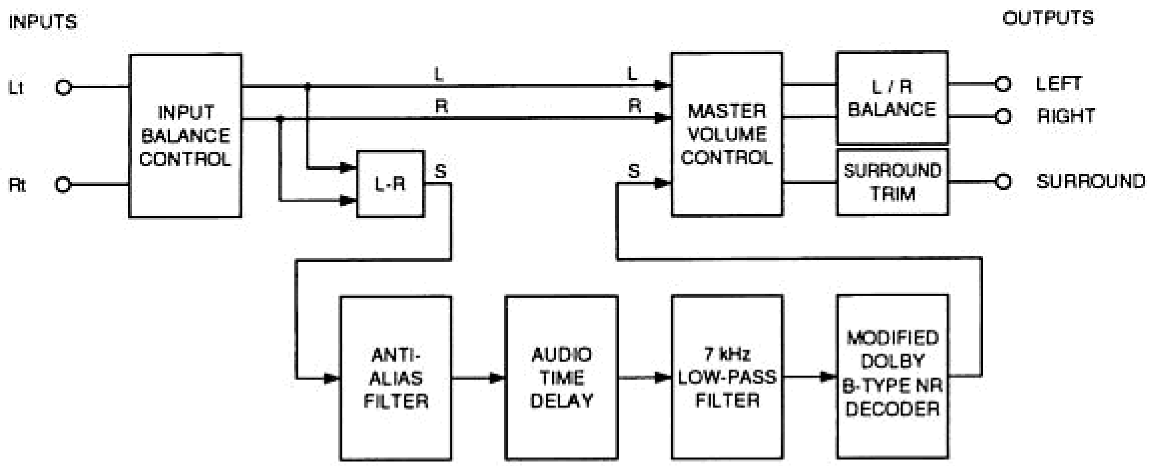

Figure 3.2: Diagram of Dolby Surround passive decoder [Dressler 1997].

The Dolby Surround decoder shown above (fig. 3.2) uses a passive decoding scheme and works in a reverse way to the encoding method. The interesting parts to note are the addition of a time delay to the recovered surround signal, and the fact that the surround signal is not removed from the produced left and right channels. The use of the time delay has been shown by Dressler to help ensure that any elements of the front channels that are present in the surround channel are replayed later in the surround speakers [Dressler 1997]. With reference to the research of Haas effect it can be shown that this method will help prevent the surround image being drawn away from the screen. The fact that the out of phase surround information is not removed from the frontal image is worrying because it will diffuse the stereo image [Dressler 1997] and cause poor low frequency response [Howard 1996]. If these psychoacoustic effects are taken into account it can be proved that the Dolby Surround method, while perceptually satisfactory is a significant compromise to the spatial qualities of the source material.

3.2.2 Dolby Pro Logic

In 1987, Dolby laboratories improved upon the limited ability of the Dolby Surround system to localise sounds, and create isolation between front and surround signals. The system reads the same Dolby Surround encoded signals, but incorporates the use of an active (logical dominance decision) matrix in the decoder stage, hence the name Dolby Pro Logic.

At the centre of the new system is the "Pro Logic" Adaptive Matrix which continuously analyses the two encoded signals, detecting the soundtrack dominance in both magnitude and direction. The calculation of dominance means that the system can steer the decoded soundfield toward the dominant element within the encoded soundtrack, therefore increasing the front to surround isolation and improving the localised precision [Rumsey 2005]. It does this producing a signal dominance vector in which the x-axis represents the left/right dominance pair, and the y-axis the centre/surround dominance pair. Dressler proved that if the magnitudes of these two signals are then resolved along each axis, and then converted from rectangular to polar coordinates. The angle of the resultant vector corresponds to the angle (direction) of the dominant sound, while the magnitude of the vector its relative dominance within the soundtrack [Dressler 1997]. This idea is shown graphically below in figure 3.3.

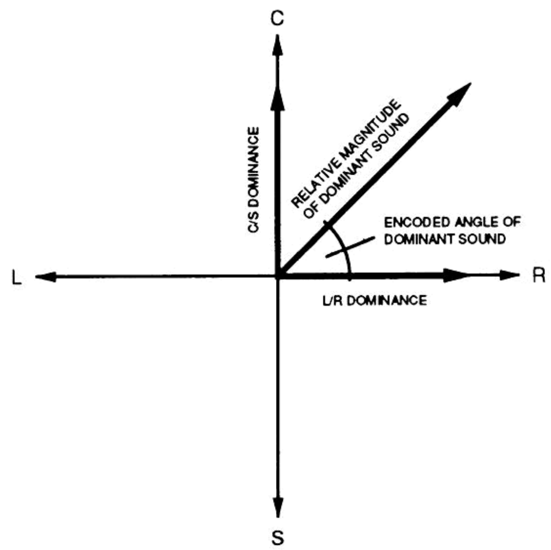

Figure 3.3: Diagram representation of dominance vector [Dressler 1997].

The signal dominance vector is created by first removing strong low frequency signals which contain no directional cues, and lowering high frequencies that contain uncertain phase information due to transmission errors. Then, two bipolar control voltages are produced by full wave rectifying the principal signals (L, R, C and S), passing the corresponding DC pairs (left/right, centre/surround) through log conversion, and lastly obtaining the difference between them. The resulting control voltages represent the required left/right and centre/surround dominance magnitudes.

Calculations are then performed on the control voltages to split then into their positive and negative elements, cumulating in four unipolar control voltages. These control voltages are then finally used to control eight voltage controlled amplifies (four amplifying L_T and four amplifying R_T), which in combination with the encoded L_T and R_T signals are combined to create the four channel decoded output [Dressler 1997]. This complicated active system is shown diagrammatically below (fig. 3.4).

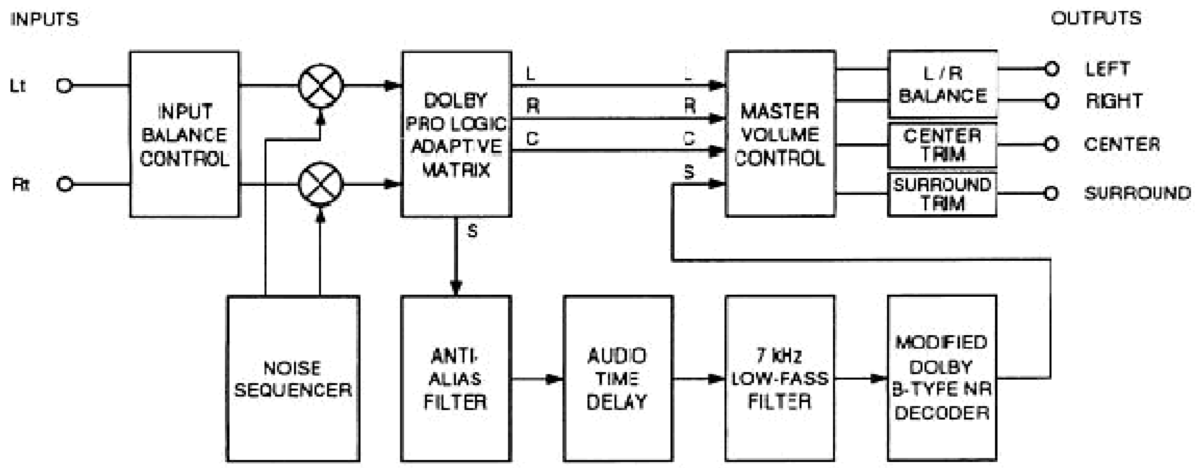

Figure 3.4: Diagram of Dolby Pro Logic Decoder [Dressler 1997].

While the Dolby Pro Logic system addresses some of the flaws within the Dolby Surround decoding method, it still creates and 3:1 stereophonic image and not 5.1 surround. Therefore it still suffers from a monophonic surround channel and the envelopment issues mentioned in earlier sections of this paper.

The Pro Logic II decoder (shown in fig. 3.5) developed after and used in the most recent computer games, only emulates 5.1 surround and still does not allow for a two channel surround signal to be encoded as multiple channels [Dressler 1998]. It is the author's belief that this limitation is due to the fact that all the systems use an encoding method that was developed in the 1970's, and to allow for backward compatibility only the decoding method was ever changed.

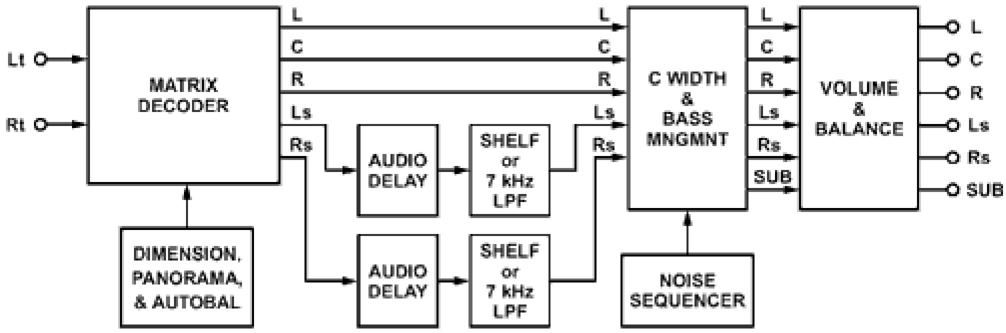

Figure 3.5: Diagram of Dolby Pro Logic II [Dressler 1997].

3.2.3 Further Developments

In the recent years new computer games consoles known as 'next generation' have been developed. These modern systems incorporate faster, more complex central processing units and a larger amount of random access memory (RAM), as well as embracing the newly produced media type of Blu Ray [Altizer 2007]. Below the Blu Ray media type is discussed, and it is shown that while it does provide faster data transfer rates compared to previous systems, it is not in proportion with the increased amount of data needed to be transferred. The next generation consoles do though allow for more complex decoding techniques to be used, and introduce the idea of real time decoding due to increased processor ability [Microsoft 2006].

3.2.3.1 Blu Ray

The Blu Ray Disk (BD) was created with the objective of producing a new media type that could store greater than two hours of High Definition Television (HDTV), in combination with multichannel audio. It was decided that based upon the broadcast standard data rate of 24Mb/s (Megabits per second), two hours of HDTV would require a data capacity of 22 GB [Blu Ray Founders 2004]. The method in which BD stores the increased amount of data is to use a blue-violet laser (400nm wavelength) which has a smaller wavelength to the current red laser (650nm wavelength) used in DVD technologies. This in turn causes the area of the beam that hits the disk to become smaller (fig. 3.6), which results in a higher density of data being able to be stored on disk.

The problem is that the current boundary of 10,000 RPM for spinning a dual layer optical disk [Bennett 2006], has limited the data transfer rate of a BD to be 8.58MB/s based on a x2 speed BD drive (x1 transfer rate 4.29MB/s). If you compare the ratio of data capacity to data transfer rate (table 3.1) it is possible to shown that data transfer speeds have not been able to match the increases in data capacity. Also with the Blu Ray being used on next generation systems the amount of data required to stream from disk will also increase. This is due to the increase in complexity and resolution of the 3D model meshes and environmental textures mentioned in earlier sections of this paper.

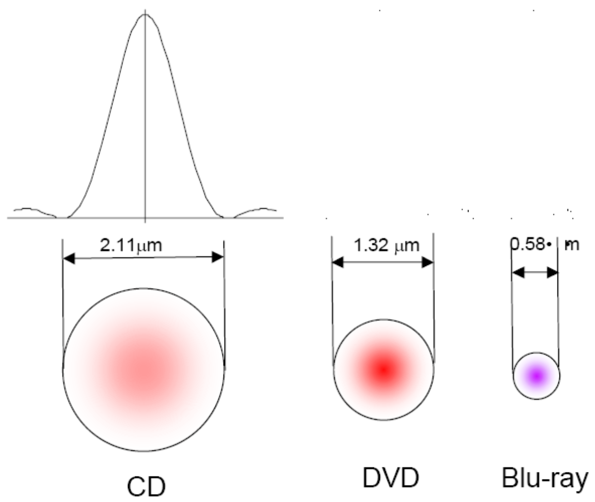

Figure 3.6: Diagram of Blu Ray beam size area compared to current media types [Blu Ray Founders 2004].

Media	Transfer Rate	Capcity	Ratio
DVD –DL x4	5.28 MB/s	8.5GB	1:1610
Blu Ray x2	8.58MB/s	50GB	1:5828
HD DVD x2	8.71MB/s	30GB	1:3444

Table 3.1: Comparision in transfer and capacity of different media types.

3.3 Summary

As this chapter of the paper has stated, the increased envelopment and spaciousness of surround sound systems shown by Blauert, adds to the overall involvement of a computer game. The problem is that limitations of current generation systems mean that full 5.1 capabilities are inaccessible and only 3:1 stereophony or emulated 5.1 can be realised, which were shown to be a high compromise when compared with full non-encoded 5.1 material. Even with new developments in storage media such as Blu Ray, it was proved that the higher data rate was not in proportion with the increased storage amount. Therefore with the amplified amount of data needed to be streamed from the disk, it would still be implausible to stream five channels of audio within the media's data rate. This facilitates the need for an encoding system that can incorporate 5.1 multichannel audio, while remaining straightforward in the decoding stage. The decoding aspect is required to be simple so that the real time decoding does not introduce latency between audio and visual events, which were shown by Mued to be highly perceivable.

In section 4 the background mathematics behind Principle Component Analysis and the Karhunen-Lòeve Transform will be introduced. This will provide a strong grounding into the multichannel audio data reduction method discussed in section 5.

4. Mathematical Background of the Karhunen-Lòeve Transform

In order to understand the methods of Principle Component Analysis (PCA) and the Karhunen-Lòeve Transform (KLT) discussed in the subsequent chapters, it is first important to understand the elementary mathematical topics behind it. Therefore in this chapter and its subsections, the multivariate statistical vocabulary of mean, variance and covariance will be introduced, along with the matrix terminology of Eigenvalues and Eigenvectors. These areas will provide a strong grounding in to the mathematical techniques that make up PCA and the KTL.

4.1 Arithmetic Mean

The most fundamental aspect of a data set needed to be understood is the idea of the set's mean. The mean represents the arithmetical average for a given set of values or data. It is very important to note that the mean is not the middle value or the most likely value, these being the median and mode respectively. The arithmetic mean is calculated through the use of the formula below (fig. 4.1), and given the symbolic representation $\bar{X}$.

$$\bar{X} = \frac{\sum_{i=1}^{n} X_i}{n} \qquad (4.1)$$

The arithmetic mean does not hold major information about the data independently, but when incorporated into other formulas it becomes very significant.

4.2 Variance and Standard Deviation

Even when two data sets have the same calculated arithmetic mean, the data within the sets can vary greatly. Therefore it is useful to have a value which represents the spread of the data, or to define it another way the average distance of data away from the mean value [Smith 2002]. In the area of statistics there are two values that hold such information, variance and standard deviation. The relevant formulas for calculating these two statistical values are shown below, in figure 4.2 and 4.3.

$$s^2 = var(X) = \frac{\sum_{i=1}^{n}(X_i - \bar{X})(X_i - \bar{X})}{(n-1)} = \frac{\sum_{i=1}^{n}(X_i - \bar{X})^2}{(n-1)} \qquad (4.2)$$

$$s = \sqrt{\frac{\sum_{i=1}^{n}(X_i - \bar{X})^2}{(n-1)}} \qquad (4.3)$$

It is simple to notice the close relationship between these two formulas, with the standard deviation being the square root of the variance equation. The reason for this is that the variance equation produces a measurement in units squared which is illogical for use in most applications [Field 2005]. Consider calculating the spread of students within a college, it would not make sense to use the measurement students squared. The reason for including both equations within this paper is because it is important to understand the close similarities between the two, and also the relationship between variance and covariance discussed in the proceeding section.

4.3 Covariance

Both the previously discussed values of variance and standard deviation looked solely at single dimension data sets, that is data sets which contained only one variable. For the remainder of this paper the amount of variables within the data set will be referred to as the amount of dimensions, for example if the data set contains four variables it will be called 4-dimensional. Many statistical applications will concern themselves with multiple dimension data sets, and in these situations it would be highly valuable to be able to calculate the variance or relationship between these dimensions. The name given to this type of calculation is covariance ("co" meaning joint) [Smith 2002]. The covariance equation (fig. 4.4) is always measured between two dimensions, and it is important to note at this stage that variance is covariance with both dimensions being the same, a dimension's covariance with itself (shown by comparing the two equations) [Everitt 2001]. It is this reason that creates the variance measurement of units squared discussed in the previous section.

$$cov(X,Y) = \frac{\sum_{i=1}^{n}(X_i - \bar{X})(Y_i - \bar{Y})}{(n-1)} \qquad (4.4)$$

In all of the discussed "spread" equations *(n – 1)* has been used to divide by rather than *n,* which was used in the original arithmetic mean formula. The reason for this lies in the fact that *n* represents the number of samples within the data set, also known as the number of degrees of freedom. If *n* is used when dealing with a sample of a population and not the whole population, then having all degrees of freedom will introduce bias in the given result. This is because one parameter is being held constant but all degrees of freedom are being incorporated [Field 2005]. When *(n-1)* is utilised a more "true" estimate of the entire population is given, especially in situations when the number of samples is much lower than the total population [Henning 2006]. Therefore *n* should only be used when dealing with the whole population.

4.4 Correlation

The statistical value of correlation if very closely related to covariance expect for the fact that it has a normalised output [Everitt 2001]. The available digits for a correlation value vary from -1 to 1. The correlation value of -1 represents complete correlation but of out phase, the value of 0 indicates complete decorrelation (or no correlation), and the value of 1 represents complete correlation with identical polarity. The formula uses the standard deviation of the two dimensions to normalise the covariance by dividing the covariance measurement by the both. This is shown in the correlation equation below (fig. 4.5).

$$corr(X,Y) = \frac{\sum_{i=1}^{n}(X_i - \bar{X})(Y_i - \bar{Y})}{s_1 s_2 (n-1)} \qquad (4.5)$$

4.5 The Covariance and Correlation Matrices

If the amount of dimensions are increased even higher than the two mentioned above, then it will give rise to a situation in which multiple covariance calculations are required, to represent the covariance of the data set. An example of this would be a data set with three dimensions *X, Y, Z*. Within this data set three different covariance values could be calculated, *cov(x, y)*, *cov(x, z)* and *cov(y, z)*. There are only three differing values because corresponding dimensions (e.g. *cov(x, y)* and *cov(y, x)*) will produce equal results. Therefore if a data set with n dimensions is considered it can be shown that $^{n}/_{(n-2)*2}$ separate covariance values could be produced [Smith 2002].

The covariance matrix is the matrix representation of all the covariance values available for calculation; this includes the identical values. For example the three dimensional data set mentioned above would create the covariance matrix shown below in figure 4.6.

$$C = cov(X,Y,Z) = \begin{bmatrix} cov(x,x) & cov(x,y) & cov(x,z) \\ cov(y,x) & cov(y,y) & cov(y,z) \\ cov(z,x) & cov(z,y) & cov(z,z) \end{bmatrix} = \begin{bmatrix} var(x) & cov(x,y) & cov(x,z) \\ cov(y,x) & var(y) & cov(y,z) \\ cov(z,x) & cov(z,y) & var(z) \end{bmatrix}$$

(4.6)

It is important to note that the matrix produced is symmetrical about the main diagonal, due to the *cov(x, y)* and *cov(y, x)* etc. values being identical. Also as shown by the second matrix in figure 3.7 the values in main diagonal are produced by the covariance between the dimension and itself, therefore resulting in the variance value of that dimension.

The closely related correlation matrix is created by a similar method to the above covariance matrix. The difference being that in the correlation matrix all element values will fall between -1 and 1 due to the normalising by standard deviation that accrues within the equation.

4.6 Eigenvectors and Eigenvalues

Eigenvectors, or characteristic vectors (the German word "eigen" meaning "inherent" or "characteristic") are a unique set of vectors associated with a square linear transformation matrix (one which contains $n*n$ elements). Relating to a linear transformation matrix there will be one corresponding eigenvector for each dimension, therefore an $n*n$ matrix will produce n eigenvectors. These corresponding dimension eigenvectors will also be orthogonal, i.e. at right angles to each other [Smith 2002].

The calculation of eigenvectors is highly useful within the areas of mathematics and science because the eigenvector contains information about the dimensions of a transformed multidimensional data set [Henning 2006]. The eigenvector achieves this descriptive quality because after the linear transformation it is left unaffected or simply multiplied by a scale factor. The scale factor produced is known as the eigenvalue and represents the magnitude of the eigenvector, with an eigenvalue of 1 representing the unaffected state mentioned above. With this taken into account it is possible to see that an eigenvector will always be calculated in pair with its corresponding eigenvalue. The complex iterative method of calculating the eigenvectors and eigenvalues of large transformation matrices is beyond the range of this paper, and for the proceeding research and experimentation they will be calculated with the use of the mathematical computer program *Matlab*. It is important to note that all offline (non real time) calculations of the KLT will be produced through the use of Matlab for the duration of this paper.

4.7 Summary

Within this section the basic statistical mathematics of mean, variance and covariance were established. It was then shown how the covariance measurement could be used with multi-dimensional data to form a covariance matrix. Fianlly, the matrix calculations of eigenvectors and eigenvalues which discussed as a method to gather information from a transformation matrix.

These mathematical techniques will now be applied to the area of component analysis in order to simplify a large data set. This statistical technique will then be shown to have application within the area of multichannel audio data reduction.

5. Principle Component Analysis and the Karhunen-Lòeve Transform

When dealing with large multidimensional data sets, it is often very useful to have a method for simplifying the set and expressing it in basic components [Field 2005]. One such method is Pearson's technique of Principle Component Analysis first put forward in 1901. In his paper he introduces the ideas of using line of best fit and planes to describe a multidimensional data set [Pearson 1901]. In recent years this method has been developed into new techniques that allow for more applications, for example Kramer and Mathews' data compression of correlated audio signals [Henning 2006] in their 1956 speech vocoder [Henning 2006]. The Karhunen-Lòeve Transform was more of an adaptation of Pearson's original PCA technique than a separate development. With it Karhunen and Lòeve demonstrate the effectiveness of PCA beyond that of analysing multivariate statistics. It has been shown by Zieliński and Henning that the KLT is more suitable in the context of audio signal processing and therefore will be used in the proceeding research and experimentation. It is important to note though that PCA and KLT are essentially identical [Henning 2006].

5.1 Scatter Plots

When introducing the idea of KLT it is first important to consider a graphical representation of the method. In the example below a two dimensional data set is displayed in a scatter plot (fig. 5.1). In this scatter plot the x dimension/variable (first dimension) is plotted against the y dimension/variable (second dimension) in order to assign each value within the data set a Cartesian coordinate (x, y). In the example data a degree of correlation is present so the scatter plot takes the form of an ellipse, which can also be represented graphically (noted in fig. 5.1). As also shown in figure 5.1 (lines e^1 and e^2) it is possible to represent the dimensions and orientation of the ellipse with two orthogonal vectors. These can be referred to as the "line of best fit" and the "line of worst fit", as in the diagram from Pearson's 1901 paper which is shown in figure 5.2 [Henning 2006].

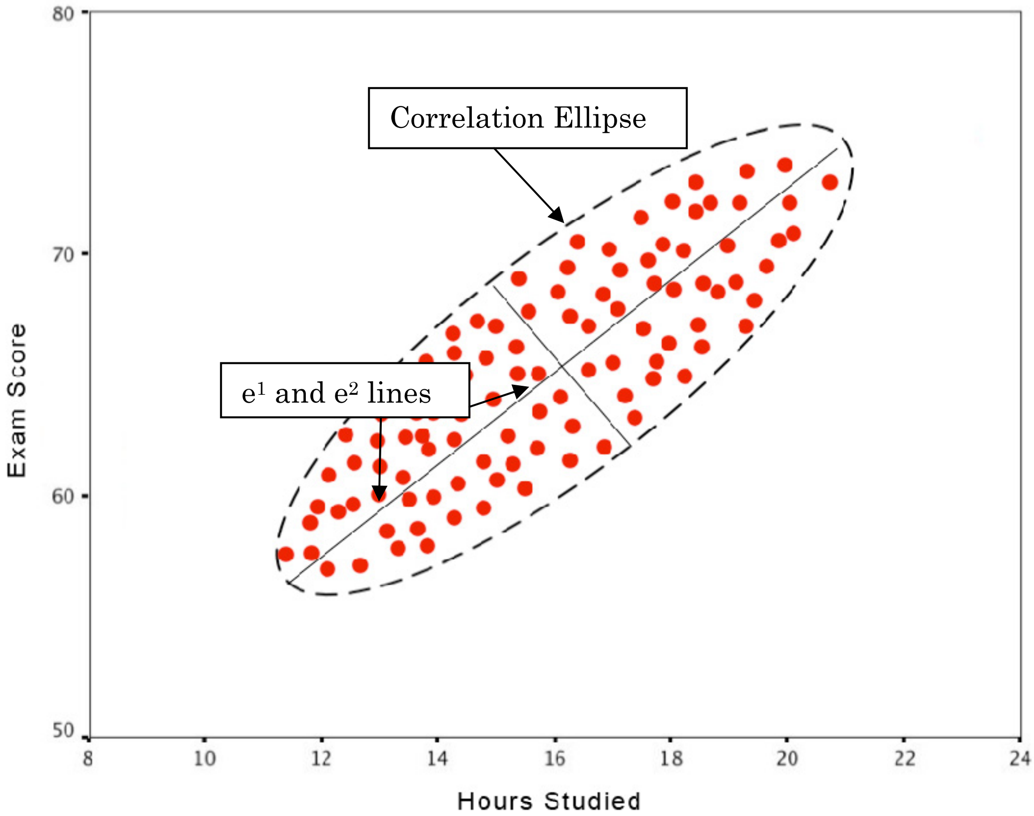

Figure 5.1: An example scatter plot [adapted from Henning 2006]

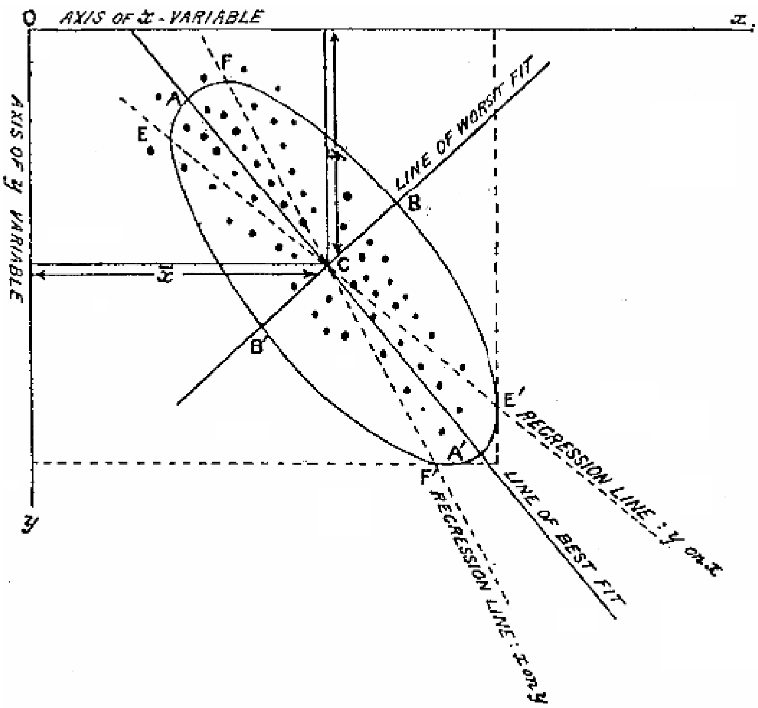

Figure 5.2: A graphical representation of Principle Components Analysis [Pearson 1901].

It is interesting that if the covariance matrix for this data set is determined, and then the eigenvectors and eigenvalues of this matrix are calculated, the perpendicular eigenvectors created will correspond to these two lines (line of best fit and line worst fit) and (as stated above) represent the orientation of the ellipse. The magnitude of these eigenvectors (their paired eigenvalues) will in turn correspond to the length and width of the ellipse (the ellipse's dimensions).

In the above example it is possible to see that one eigenvector is of greater magnitude than the other, therefore its paired eigenvalue will also be higher. It is simple to see that the ratio between the eigenvalues describes the shape of the ellipse, and therefore is a representation of the correlation between the two dimensions [Henning 2006]. Using the available eigenvalues, the statistical significance of each corresponding eigenvector can be calculated based upon the compared percentage. Therefore it is possible to see that if the correlation with in the data set was high, one eigenvalue would hold a particularly higher percentage when compared with the rest. In this case the total variance within the matrix could be expressed through the use of only this single eigenvector.

In summary it can be shown by the above example that if correlation between dimensions within a data set is high, the set can be represented by a single eigenfunction, or to use Pearson's thinking "Principal Component".

5.2 The Scree Plot

In a situation where a number of eigenfunctions or "Principal Components" are being used to characterise a large multi-dimensional data set, it is useful to have a graphical way of representing each eigenfunction's or component's statistical contribution. As mentioned in the previous subsection, the statistical significance of a component is indicated by its paired eigenvalue. Therefore it is simple to show that a numerically ordered plot of the eigenvalues in reference to their components would prove an adequate solution. This type of plot is referred to as a "Scree Plot". Henning states that the word "scree" refers to debris fallen from a mountain and present at its base [Henning 2006].

The scree plot is used to analyse which of the components are considered statistically insignificant. This is done by introducing a limit called the point of inflexion. Any components which have an eigenvalue which is lower than this point are deemed insignificant and discarded [Field 2005]. The general value for the point of inflexion is when eigenvalues are less or equal to 1, and can be seem graphically as an "elbow" at which the line connecting the scree values levels out (fig 5.3).

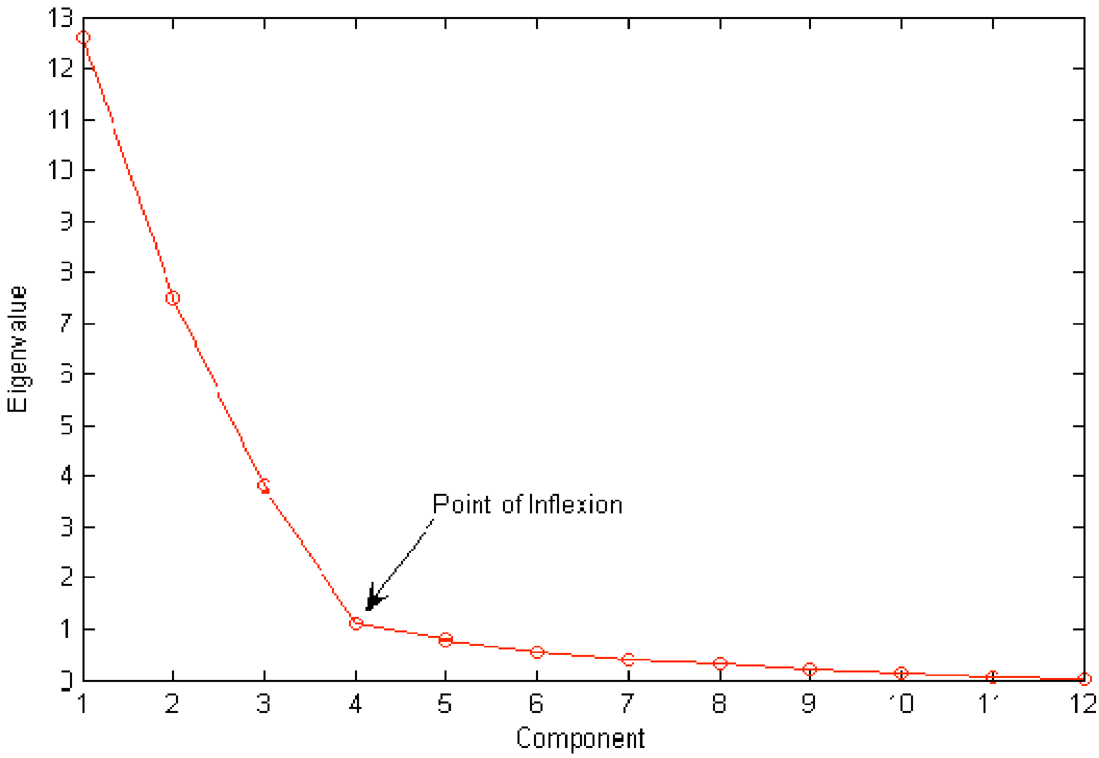

Figure 5.3: An average scree plot show the point of inflexion [Henning 2006]

5.3 Audio Applications Incorporating KLT

The Karhunen-Lòeve Transform has been widely used within the areas of data analysis and statistics in order to simplify large data structures. Additionally in recent times with the support of increasingly complex computer systems, the ability of the KLT to indentify data patterns and indicate its similarities has become increasingly useful [Smith 2002]. Such applications include the work of Turk and Pentland who were among the first to use the KLT in the area of facial recognition [Turk 1991], and Lu's work on noise reduction in seismic imaging. While these applications can be helpful in researching areas of audio in which the KLT could be incorporated [Henning 2006], it is not within the scope of this paper to discuss them further. Instead this chapter will concern itself with the well documented audio application of interchannel redundancy removal within multichannel audio, due to its direct relevance to discussions in earlier chapters.

5.3.1 Interchannel Redundancy Removal within Multichannel Audio

The intention of interchannel redundancy removal is to exploit the correlation between channels within a multichannel audio signal, as a form of data reduction. Taking the research of previous chapters it can be shown that the KLT would be a highly suitable method for indentifying such correlation. Initial research by Yang has proved that multichannel audio which contains a large amount of correlation between channels, provides an exceptional candidate for an interchannel redundancy technique incorporating the use of the KLT. The method Yang uses involves the calculation of the multichannel signal's covariance matrix. This matrix in then utilised in the KLT computation of decorrelated "eigenchannels" [Yang 2003]. The KLT will produce an amount of eigenchannels equal to the number of channels present in the multichannel signal, for example if the five main channels of a 5.1 recording are used five eigenchannels will be created. The research showed that depending upon the amount of correlation between channels it is possible to describe a five channel signal, with the use of as little as two, three or four eigenchannels. The eigenchannels can then be decoded, using the relatively simple inverse Karhunen-Lòeve Transform, into multichannel audio for playback with only a small perceptual change. In 2003 Yang et al. further developed the use of the KLT in the coding of multichannel audio signals, by incorporating its use within an Advanced Audio Coding (AAC) algorithm [Yang 2003]. The AAC algorithm is an example of lossy audio coding, due to the fact the decoded audio signal will have "lost" a proportion of its signal information when compared to the original. The created codec was named Modified Advanced Audio Coding with Karhunen-Lòeve Transform (MAACKLT), and uses the KLT in the processing before (pre-processing) creating decorrelated eigenchannels that are compressed by the real time AAC. AAC employs the use of a Modified Discrete Cosine Transform, and only contains the simple use of "joint coding" to reduce interchannel redundancy [Yang 2003]. It was shown that the additional interchannel redundancy removal of the pre-processing KLT stage achieved a higher quality result than the original Advanced Audio Coding scheme [Yang 2003]. Detailed descriptions of the AAC scheme are beyond the scope of this paper. The main difference between the KLT section of the MAACKLT codec and Yang's original KLT research was the use of a temporal-adaptive KLT that took "snapshots" of the original signal, and calculated a separate KLT eigenchannels for each. The perceptual qualities of both the original KLT and the newly developed temporal-adaptive version will be discussed in the proceeding section.

5.3.1.1 Perceptual Effect of Interchannel Redundancy Removal

In 2006 Henning undertook research into the perceptual effect of interchannel redundancy removal in multichannel audio signals, using the original KLT method developed by Yang. The research concerned itself with offline processing of a five channel audio signal, and tested the process with the use of MUSHRA listening tests in a controlled ITU-R BS.775 standard listening room. It was shown that the perceptual effects of removing the two most statistically insignificant eigenchannels (the two with the smallest paired eigenvalues) were undetectable. Therefore it was concluded by Henning that the KLT was highly successful in the area of multichannel data reduction on highly correlated multichannel audio signals. The perceptual effects of KLT within interchannel redundancy removal were further reinforced by Jiao in 2006 during his research into Hierarchical Bandwidth Limitation of Surround Sound [Jiao 2006]. In this research Jiao concluded that the KLT was a "promising method in the context of saving bandwidth of multichannel audio" [Jiao 2006]. At the time of writing this paper there was no published research available into the perceptual effect of the temporal-adaptive KLT, but it is the author's knowledge that research is currently being undertaken at the Surrey Institute of Sound Recording. Therefore the only listening based testing which has been conducted on the temporal-adaptive version is the single test present in the original MAACKLT research. This test involved just four listening subjects and only the basic audio quality was commented upon.

5.4 Summary

Firstly in this section a graphical representation of Principle Component Analysis and the KLT was shown and commented upon. The importance of eigenvector and eigenvalues was then discussed and the idea of "eigenchannels" was put forward. Yang's audio application of the KLT which then introduced in the area of interchannel redundancy removal within multichannel audio. Finally the perceptual effects where agured with the use of Henning's and Jiao's research.

In the proceeding section the above method is used to generate a hypothesis into a real time decoding method, for use within computer games.

6. The Use of Real Time KLT Decoding within a Computer Game

This chapter will summarise the information given in the above literary research, and then based off this information propose a real time use of the KLT for interchannel redundancy removal within the 5.1 audio of a computer game. The final subsections will concern themselves with what IS and what is NOT known, in which the proposed system will be analysed to form original experimental hypotheses.

6.1 Justification of Hypotheses

The above review of permanent literature showed that, while two channel stereophonic systems can create an accurate frontal image they have severe limitations in the area of envelopment. It was then proven through the research of Blauert that multichannel surround systems can provide increased envelopment due to their playback of sound within both the frontal and rear (surround) auditory spaces, which introduced the widely used ITU-R BS.775 5.1 playback system. The heightened quality of envelopment present in surround systems was revealed to be of high significance within the area of computer games, but that the current console technology could not recreate it without high compromise. The Karhunen and Lòeve adaptation of Pearson's Principle Component Analysis method was then introduced along with its background mathematics. This Karhunen-Lòeve Transform was discussed to have qualities that allowed for its application by Yang within the area of interchannel redundancy removal in multichannel audio signals. This data reduction use of the KLT was shown by Henning's and Jiao's separate research to be almost perceptually undetectable through the use of MUSHRA based listening test [Henning 2006].

6.2 The Premise

Firstly audio stream data will be created through the use of an offline application of the KLT. In this stage a covariance matrix of the multichannel audio will be created. This matrix will then be used in the extraction of matrices containing the eigenvalues and eigenvectors. This data can then be used to produce the principle components of the original signal, also known as the eigenchannels [Yang 2003]. By removing a certain number of the least significant eigenchannels, it will be possible to create audio stream data that contains fewer channels than the original multichannel data. This will be referred to as the KLT encoded multichannel audio data stream. This data stream can then be stored on the relevant media type in preparation for transmission. It is important to note that information concerning the inverse-KLT will also need to be created and stored along with the audio stream. The real time decoding algorithm will take the encoded audio stream data and use the information about the inverse-KLT stored on the media to create a multichannel audio signal suitable for playback. A graphical overview of this system is shown in figure 6.1.

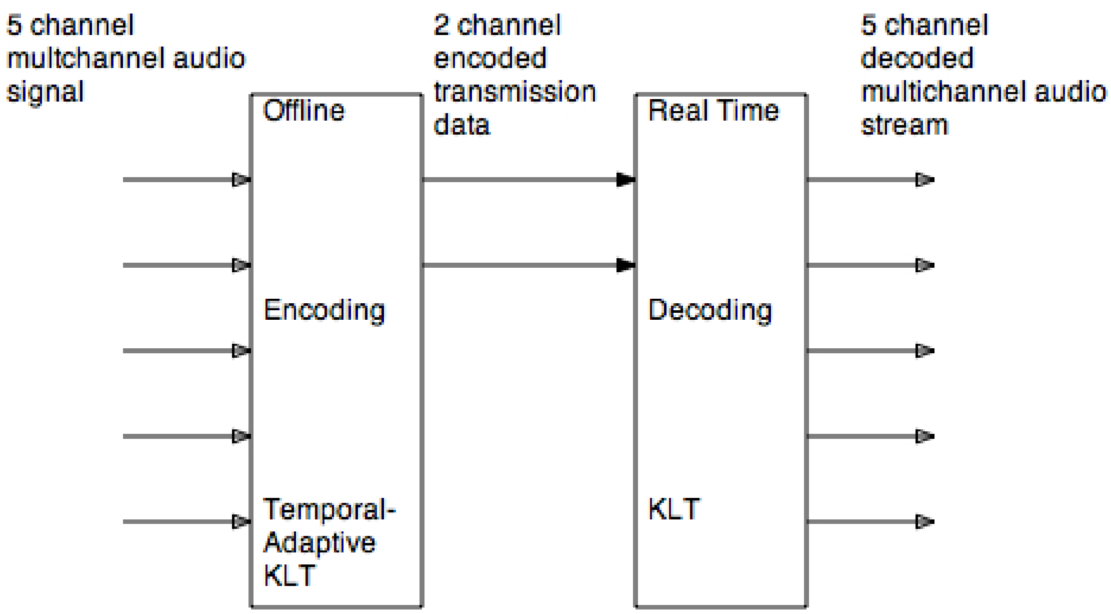

Figure 6.1: The KLT coding method [Adapted from Zielinski 2005].

6.3 What IS and what is NOT known?

The KLT has been shown by separate research to be highly adaptable for use within the area of interchannel redundancy proposed above. The created eigenchannels describe the variance within the signal and in combination with their paired eigenvalues can be used to judge the statistical significance of each component. It IS known that this measurement of statistical importance can be used to predict the perceptual importance of each eigenchannel, and therefore create a nearly perceptual undetectable data reduction method [Henning 2006; Jiao 2006]. All the current uses of the KLT within this area though use an offline (non real time) application of the method in both the encoding and decoding stages, therefore it is NOT known is the KLT will be suitable for the real-time decoding use above.

6.4 Hypotheses

Based upon the research and deduction above, it is hypothesised that the Karhunen-Lòeve Transform *will* be suitable for use in offline data reduction and real-time decoding of a multichannel audio stream. Further to this, it is hypothesised that the KLT *will* produce adequate results that allow for this newly devised system to be incorporated within a computer game system, for the two channel streaming and full playback of 5.1 multichannel surround audio. In the following sections of this paper an offline data reduction technique and real time decoding system will be developed. This system will then be tested though the use of performance analysis and formal listening trials, in order to assess these hypotheses.

7. Developmental Section

This chapter will discuss the processes that were undertaken in the development of a real time decoding system that incorporates the use of the KLT. The initial stages that were involved consisted of, the choice of software development environment including programming language, decision of channel amount in audio data stream and selection of what audio excerpts were to be used. An offline encoding algorithm was then produced in Matlab and the real time inverse-KLT decoding program designed.

7.1 Choice of Integrated Development Environment

Before any steps could be taken into the development of the system, a decision needed to be made as to the environment in which the programming will take place. Firstly the computer operating system (OS) was decided upon. It was shown through research of the market, support and amount of documentation that Microsoft Windows XP Service Pack 2 would produce the most satisfactory and convenient OS. This was mainly due to two factors; the amount of open source programming libraries and documentation available, and the large choice of integrated development environments (IDE's).

After further research into the area of computer game console programming, it was found that in nearly all applications the programming language of C++ was used. In order to develop a compatible system in keeping with the hypotheses stated in chapter 6, it was determined that C++ would be used for all real time development. This therefore limited the choice of IDE to the products that could support the C++ language. The main three IDE's available are shown below in table 7.1 with their respective advantages and disadvantages.

IDE	Advantages	Disadvantage
Microsoft Visual Studio	Very well supported. Lots of documentation. Facility for open source plug ins. MatLab support.	Expensive.
C++ Builder	Lots of features and built in libraries. Adequate documentation.	Very expensive.
KDevelop	Freeware. Open source development.	Missing for more advanced features.

Table 7.1: IDE advantages and disadvantages.

Based on the information shown is this table Microsoft Visual Studio was selected due to its OS integration, large available amount of documentation and Matlab support. A screenshot of the Microsoft Visual Studio graphical user interface is shown below (fig 7.1).

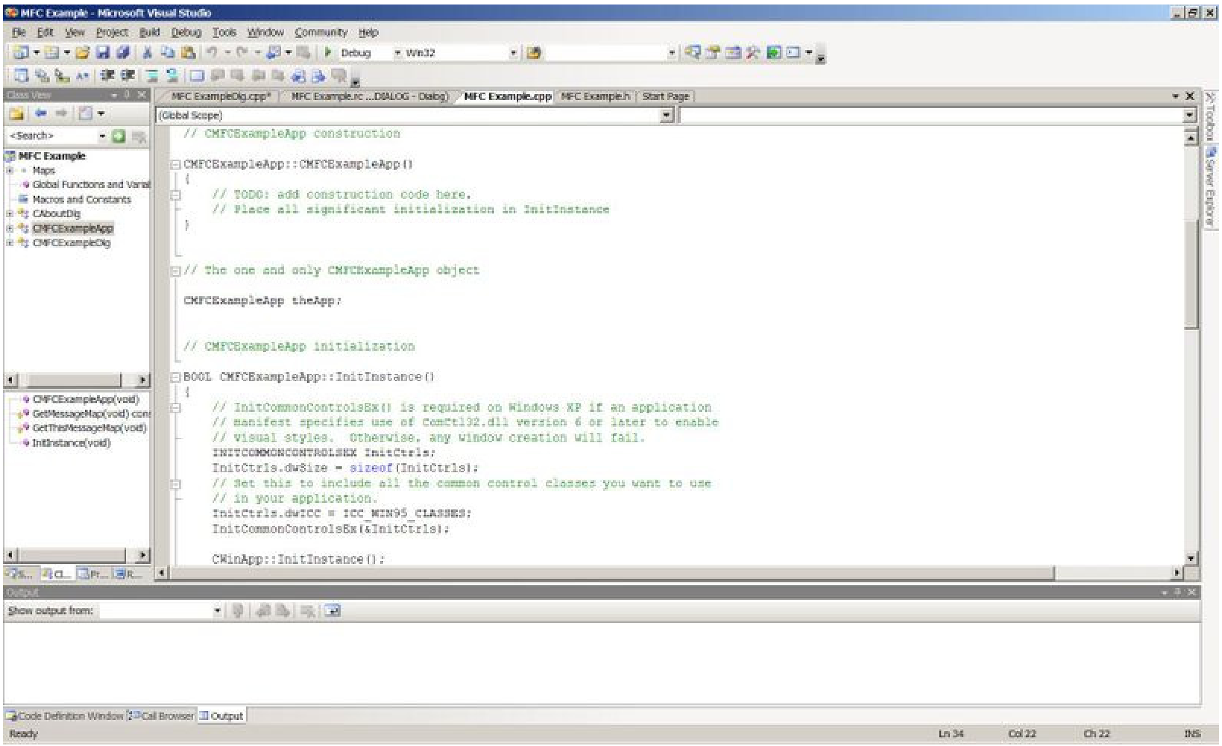

Figure 7.1: Screenshot of Microsoft Visual Studio [Microsoft 2007].

7.2 Choice of Channel Amount in Audio Data Stream

In the earlier research of chapter 2, it was shown that current and next generation computer game consoles have the ability to stream two channel audio data. Therefore it seems sensible to conduct an experiment into encoded audio streams for transmission on these consoles, using a method that is within the tested limit of the systems. For this reason it was decided that the number of channels to transmit in the audio data stream would be two. The implication this had on the developmental stage, was that the three least significant eigenchannels needed to be removed in the KLT encoding algorithm. This in turn created limitations within the choice of test audio which will be discussed in the next section.

7.3 Choice of Audio Excerpts

Four separate audio recordings were chosen for use within the development and testing stages. The content and type of each recording was selected based upon on set of certain criteria. The first criterion was that the content of the selected audio would be relevant, to the audio type that is present within computer game streaming systems. Research showed that in computer game development, sound effects are read from the console memory and panned about the stereo image, while ambiences and music needed to be streamed. Also because the ambiences and music are not panned, they require streaming as a full multichannel audio signal. For example if a 5.1 ambience is to be used, it must be streamed as a five channel audio signal, which as stated in the research above is implausible. With this taken into account it can be shown that 5.1 ambience and music recordings, would prove an excellent candidate to satisfy this first criterion.

The audio excerpts chosen were also required to contain high correlation between channels. This was deemed necessary because of the decision made above, to transmit only two encoded audio channels. It was shown by Henning that multichannel audio with high interchannel correlation, faired most highly in perceptual based listening tests where the three least significant eigenchannels were removed. This particular criterion also meant that the statistical significance of the first two eigenchannels representing the audio needed to be a high percentage of the total variance.

7.4 KLT Encoding of Audio Data Stream

The encoding of the audio data stream was created using an offline temporal-adaptive KLT [adapted from code produced by Jiao]. A commented version of the code is included for reference purposes in appendix B.

Firstly the original multichannel audio data (one of the audio excerpts chosen above) was extracted into a six column matrix within the Matlab workspace. In this matrix each column represents a discrete channel of the original multichannel audio signal. The Low Frequency Effect channel was removed from the matrix due to its insignificance in respect to this experiment (mentioned in chapter 1), and the DC offset of the resultant five column matrix subtracted (in order to create a mean value of zero). A Hann Sine Squared time domain window with half window overlapping was then used to take multiple “snapshots” of the data within the matrix. The length of the snapshot was controlled by the temporal-adaptive window length, discussed in the following subsection. Next the covariance matrix of the separate “snapshots” was calculated, and then used to extract the eigenvalues and eigenvectors into matrices. These matrices were then used to produce eigenchannels that represented the principle components of the five original audio channels. Based on the paired eigenvalues, the three least significant eigenchannels were removed. This resulted in a two channel audio stream which was then written to disk as a wav file. Finally the matrices containing the produced eigenvalues and eigenvectors were used to create the inverse-KLT information. How the inverse-KLT information in handed and written to disk is discussed in the proceeding subsection.

7.4.1 Temporal-Adaptive Window Length

At the time of writing this paper there was no available research into the perceptual effect that variance of the window length will have on the listener. Therefore the length of the temporal-adaptive window was dictated by the size of the audio memory buffer. This was decided upon because the audio buffer could now be sent data that was the same size as the window “snapshot”, making the overall process more optimised and efficient.

7.4.2 Binary Storage of Inverse-KLT Metadata

As mentioned in "The Premise" section of chapter 6, information about the inverse-KLT decoding section is also required to be stored with the encoded audio stream data. This is due to the KLT being a data driven method, and therefore fixed coefficients cannot be used in decoding [Henning 2006]. With this taken into account it was necessary to devise a method to store this information in binary metadata. Both Matlab (binary writer) and Visual Studio (binary reader) use the same data storage methods. What is meant by this is that a floating point value will be allocated the same amount of bytes in both data storage systems. Using this it was possible to write a binary file in Matlab, which contained an integer value within its header indicating the amount of floating point values that follow. When this binary file was read into Visual Studio the integer value, along with some simple arithmetic, was used to set up an iteration that read the linear binary data into a structure that mirrored that of the original Matlab matrix. This produced matrix was then used to calculate the inverse-KLT of the encoded data, in order to create multichannel audio for playback.

7.5 The Real Time Decoding System

The real time decoding system was written in the programming language C++. As mentioned in earlier sections, the availability of open source C++ libraries was of great importance when developing this system. The C++ libraries used are discussed in this section, along with an overview of the created system, including the overall architecture.

7.5.1 Toolbox – C++ Libraries

In the programming language of C++, a Library is the name given to groups of external classes and functions that are included for use within a program. In order for the classes and functions to be called within the main program they must be included within the header file. The advantage of using C++ Libraries (especially open source), is that extra functionality can be created within the program quickly and efficiently. During development the mathematical calculations and matrix data structures used with the main program, were created through the use of the MatLab C++ Math Library. Once compiled and included within the program header file, the MatLab C++ Math Library allowed for complex matrix, and the use of matrix data structures that mirrored that of the MatLab encoding algorithm

Reading of multichannel audio in the system was achieved with the use of the open source C++ Library for Audio and Music (CLAM) [C++ Library for Audio and Music 2007]. This open source (freeware) Library was created for audio analysis and synthesis applications. The functionality it provided was a simple and optimised method of reading multichannel audio data into the Matlab matrix structure comment on above. The final C++ library incorporated within the developmental system was the Open Source Audio Library Project (OSALP). OSALP was used for help with audio streaming and buffer management sub systems of the main program [Open Source Audio Library Project 2007].

7.5.2 Main Decoding System

First the memory and streaming management systems were initialised using the constructor of the management class. The purpose of the management class was to encapsulate all the functions and classes, relevant to controlling the transfer of data throughout the system. The management class was the only object within the program that had access and vision to all the aspects of the program. The initialisation of the management class triggered a chain of initialisation within all other aspects of the program. This is chain is shown diagrammatically in figure 7.2 below.

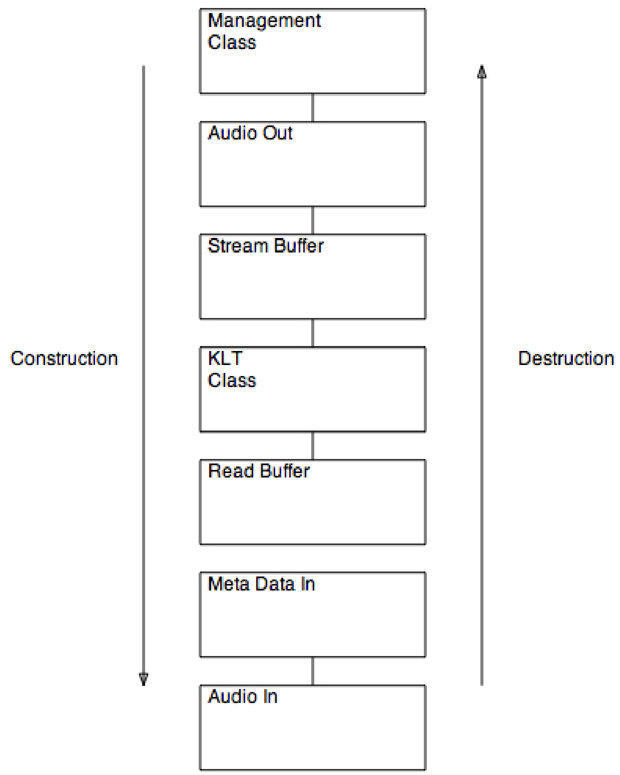

Figure 7.2: Construction and destruction chain for real time decoding system.

The separate parts of the program are set up in a backwards fashion (end data systems set up first) this insures that data is not transferred to sections before they are initialised. It is important to note that the architecture of the system was such, that all aspects of data transfer within the program were control and monitored by the management class (as stated above). The next stage was to open a file stream in order to read audio data off disk and into the read buffer matrix. At the same time, the metadata was read from disk into a separate matrix stored within the same read buffer class. The read methods and matrix structures used were designed to mirror the corresponding functions within Matlab. For example the read audio data was formatted into a matrix where, the discrete audio channels were represented by separate columns, as in the MatLab encoding method (discussed above). The incorporation of the data within the same class facilitated the easy synchronisation of both sets of data. This data was then sent to the KLT class as a signal matrix array structure to optimise memory management. The KLT class though use of the MatLab Library matrix functions then calculated the multichannel audio data for playback. The audio data was then sent to the playback stream buffers which in turn streamed it to the audio playback device. The start of all data transfer control functions was callback data sent from the audio playback class. This meant that the audio playback class never received a saturation of data, or no data ("ran dry"). The overall system architecture is shown graphically in figure 7.3.

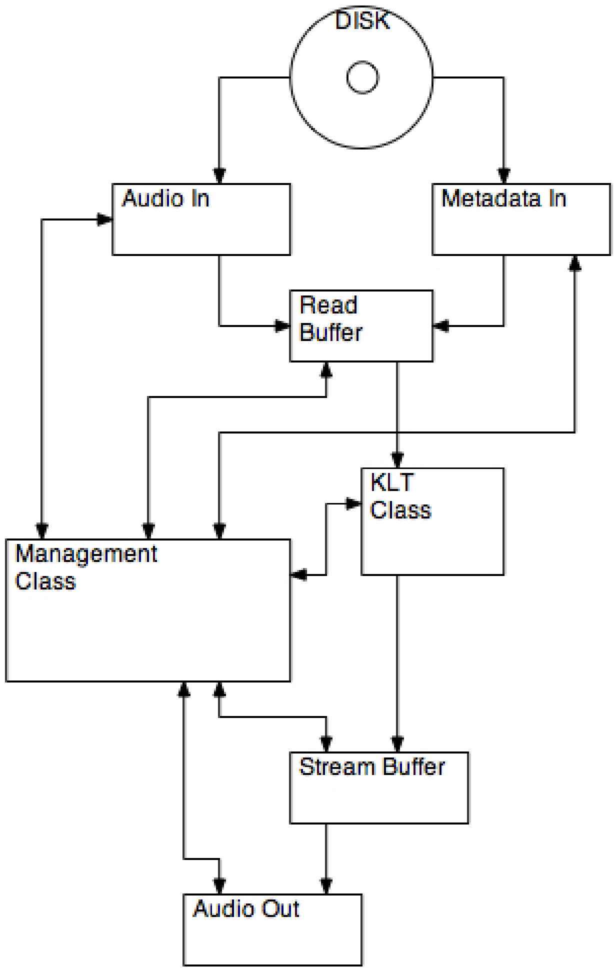

Figure 7.3: Graphical representation of system architecture.

7.6 Developmental System Testing

Assessment of the developed system was very important to gain information about its predicted performance, and as a basis for further development. In the proceeding sections the performance test procedure will be discussed. The variable that is altered throughout the testing stages is the size of the audio buffer used. The buffer was varied linearly from a value of ten samples to two thousand samples.

It is important to note that due to the close relationship between the audio buffer size and the temporal-adaptive window length, the audio stream was re-encoded at the start of each test in order to match the window length with the buffer size.

7.6.1 System Resource Test

The first test in order to investigate the hypotheses stated in chapter 6 was the general performance of the decoder and its use of available system resources. For this test the metric readout of the software development environment was used. This readout contained data about memory use and allocation, percentage amount of the central processor used and disk transfer rate. Also with the addition of open source "plug-ins" it was possible to display a predictive version of system resource usage on certain computer game consoles.

7.6.2 Overall Latency Test

It was decided that while the above test produced highly satisfactory result into analysis console performance. It did not give any insight into the latency produced within the audio stream, and has this latency varied with buffer size. To test this aspect of the decoder's performance, a time monitoring and readout class was built into the developed program. This timer was trigger by the entry of data into the program (audio read in), and stopped when data left the system (audio streamed out). The integration of this system is shown in figure 7.4.

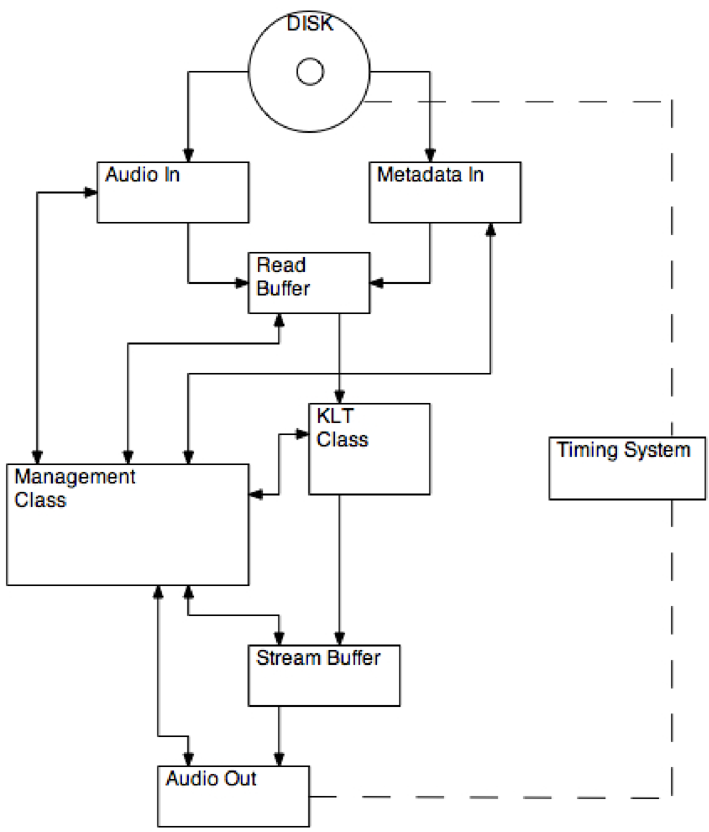

Figure 7.4: The integration of the timing system in order to measure latency.

7.7 Summary

Development was undertaken into a system that could satisfactorily encoded data through offline processes while, decoding data in real time though the use of the Karhunen-Lòeve Transform. Firstly, Microsoft Visual Studio was chosen as a software environment for the development of a C++ coded real time decoder. Audio excerpts were then decided upon based on the fact that only two channels of encoded data were to be transmitted. This meant that highly correlated ambience and music sources were chosen. The decoding system was then created, based on an architecture built around a central management class. This system was then tested to investigate it performance within this application.

In the next section the results of the above testing will be shown and analysed, in order to create a conclusion towards the stated hypotheses.

8. Developmental Results and Analysis

The results shown in this section are split into two groups, system usage and audio latency. Firstly the system usage results will be shown and commented upon. The tables in this section show what percentage of the overall system resources were used by the decoding program, while the audio buffer size was changed.

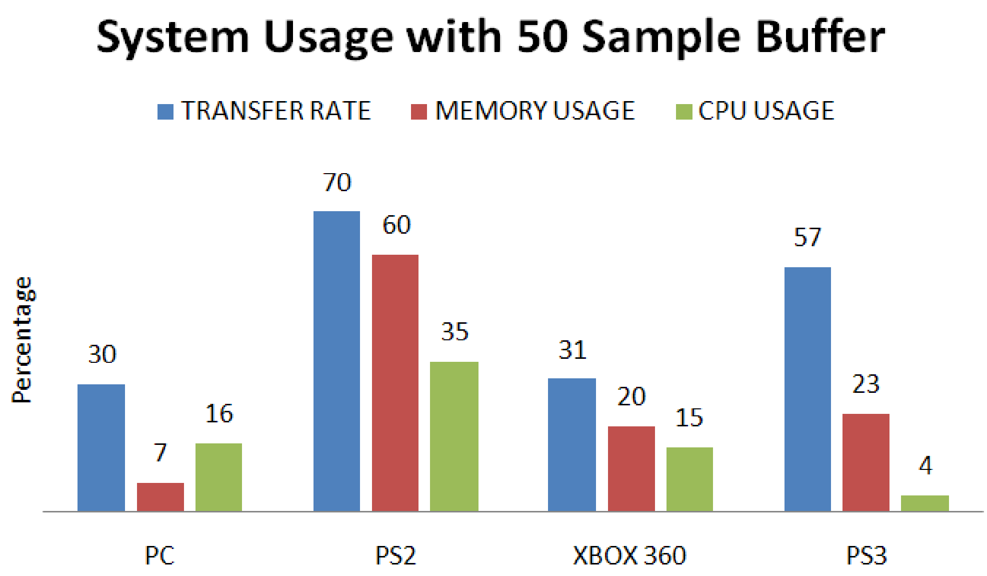

Figure 8.1: Percentage system usage with a 50 sample audio buffer.

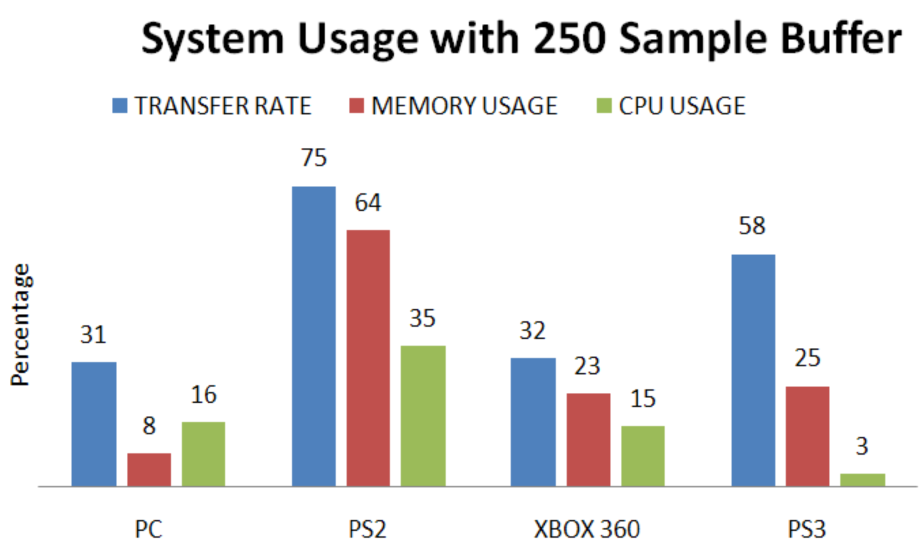

Figure 8.2: Percentage system usage with a 250 sample audio buffer.

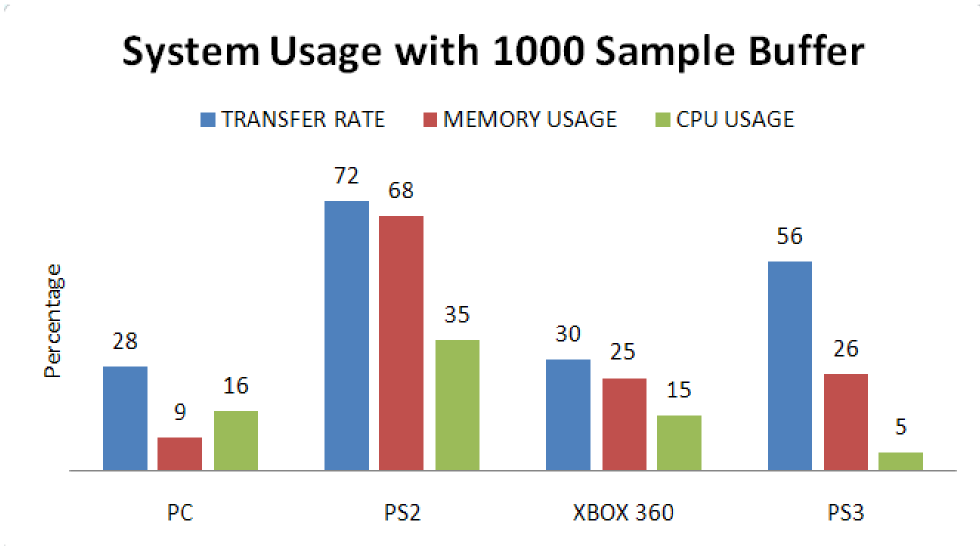

Figure 8.3: Percentage system usage with a 1000 sample audio buffer.

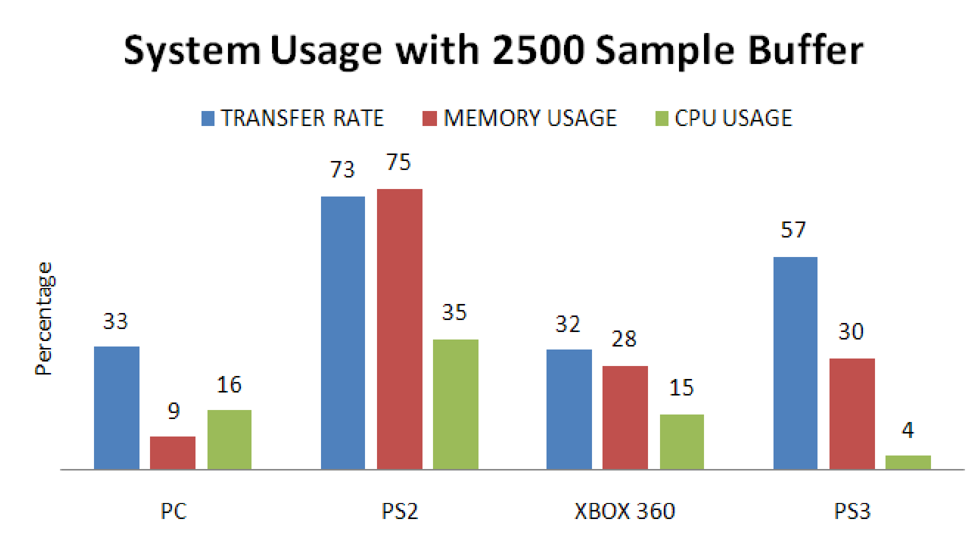

Figure 8.4: Percentage system usage with a 2500 sample audio buffer.

The main aspect that the above figures show is it that the size of the buffer has a high effect of the amount of memory allocated by the program. It can be seen, that increasing the buffer from a size of 50 samples to 2500 samples, raised the memory foot print of the system by up to 15% in the case of the Playstation 2. The resultant effect that this will have on the system is that the memory management classes will need to be further optimised. Unfortunately such advance programming ideology is passed the scope of this paper, though it should be noted that this would be an excellent area for further work.

In relation to the hypothesis, these results prove that the real time decoding works satisfactorily, and without a significant usage imprint on the two next generation consoles of the Xbox 360 and the Playstation 3. They also show that the disk transfer rates of the next generation system are not as advanced as the rest of the technology in the system. This is proved by only an 18% advantage between the Playstation 2 and 3 transfer rate results, compare to a 30% average in other areas such a CPU usage.

The second graph (figure 8.5 below) represents the amount of latency in ms, between the audio being read from the disk and then streamed. The latency can be accounted for by the use of read and stream audio buffers. Therefore it is no surprise that as the buffer sizes were increased the latency rose in a linear fashion. The resultant graph proves this stated prediction with the creation of a nearly straight line. It is also interesting to note the development console used to run the decoder, has a non-detectable amount of variance upon the latency time. Therefore it can be stated that the latency is this system in completely dependent upon the size of the audio buffers.

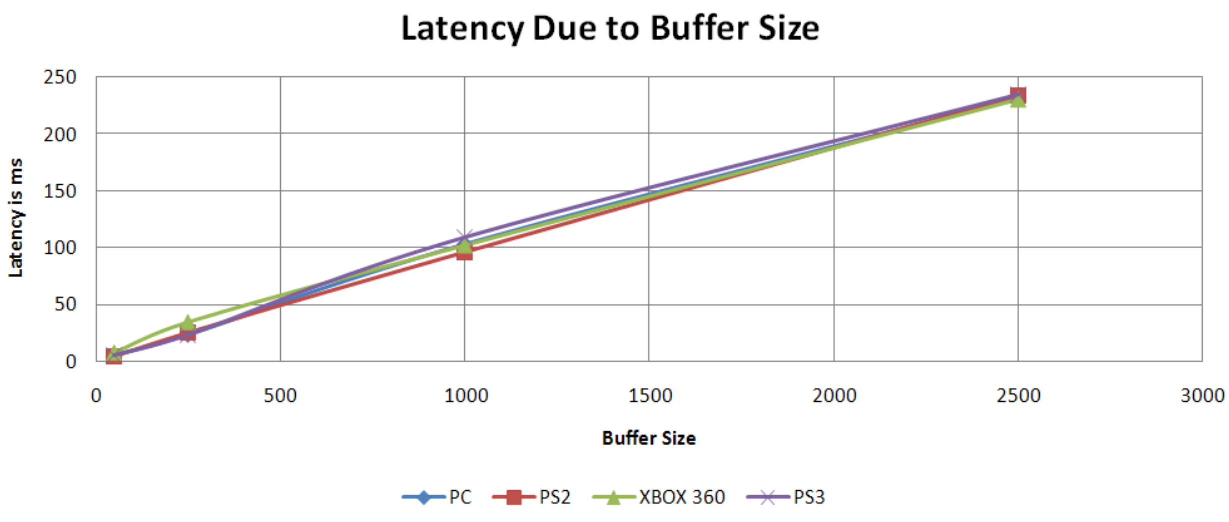

Figure 8.5: Diagram of audio latency in relation to buffer size.

8.1 Summary

The results above show that the real time decoding system is capable of being run on both current and next generation systems. The system resource percentage for the current generation console though is too high for a commercial use of this system. The latency of the audio streaming system was then tested, and it was concluded that the console used had an undetectable effect on the produced latency.

In the next section an experiment will be devised in order to test if the created latency is perceivable. This will then result in a conclusion into whether the real time KLT decoding method is subjectively suitable for use within a computer game system. It will be investigated through the use of a "test level" game interactive listening test, in which overall sound quality in judged.

9. Experimental Section

In order to judge the success of the developed system within the context of a computer game, it was important not only to test the performance of the technique, but also the incorporation of it within a game “test level”. To investigate the hypotheses a “test level” was created and experimentation within it was conducted by a group of 10 experienced listeners.

9.1 Aspects Taken From Development

In the developmental section audio was selected based upon a set a criteria. This resulted in the selection of four audio excerpts that were used for both the development stage and the “test level” experimentation. The system performance testing results also revealed that due to computational complexity the system could not efficiently run on current generation game consoles, which is taken into account in the proceeding section.

9.2 Choice of Testing Computer Game Consoles

From the developmental results it was possible to conclude that due to the amount of processing speed and random access memory required, the use of a current generation system was implausible. It was shown in the results that even though the current generation console could successfully *run* the real time decoding program, the amount of system resources left were unacceptable for commercial use. This information resulted in the decision to use two next generation consoles in the “test level” experiment, the Sony Playstation 3 and the Microsoft Xbox 360. These two systems were decided upon because of their high system performance and cutting-edge features. Both of these factors meant that the developed decoding system could be used on then commercially and left room for future developments. Also further to this the Sony Playstation 3 uses the new media type of Blu Ray x2 speed (discussed in chapter 2), while the Microsoft Xbox 360 uses an x16 speed DVD drive. This provided a good opportunity to test the next generation media type as well as the next generation systems.

9.3 The "Test Level"

The produced "test level" consisted of four square rooms; within each room one of the audio excerpts could be heard. The layout of the "test level" is shown below, in figure 9.1. The audio excerpts were un-scaled within the room so that whatever the character position within the three dimensional space, the volume of the audio remained unchanged. There was a small crossfade when moving from each room, which was read out from the two audio buffers. This was used to avoid digital clicks being produced during transition between audio, and allow the new audio buffer to be filled before streaming. The level was created used very basic texture mapping and polygon based graphics. The three dimensional model was produced in the graphical software environment Maya, and integrated into the game using Renderware Studio. The use of "grey data" was used to model the real streaming usage from disk of an average computer game. Grey data is the name given to random data that contains no in game value, its only use is to "act" like real in game data to model the data transfer rate of the media.

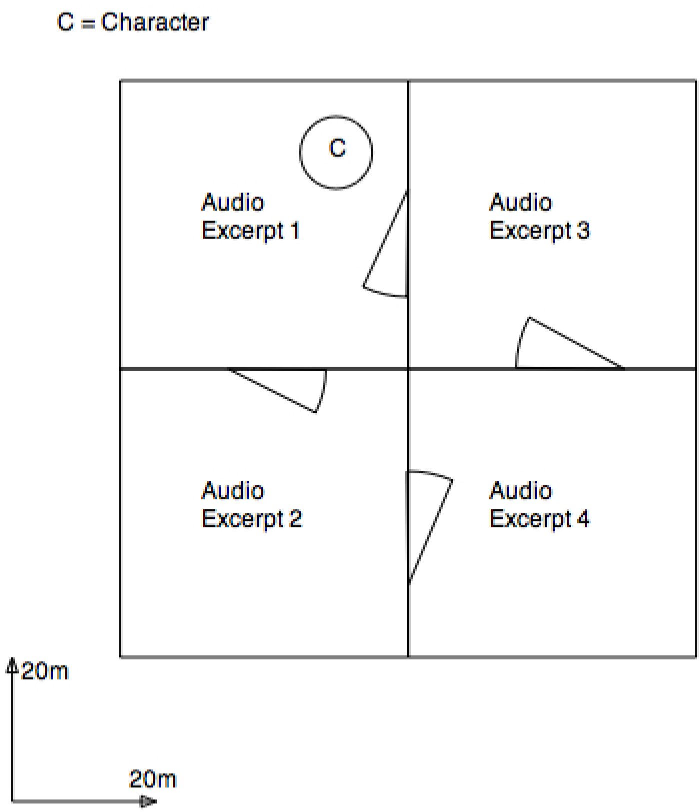

Figure 9.1: The "test level" layout.

9.4 What to Listen For?

The test that conducted required the listens to judge the overall sound quality of the produced audio. This was specified to the listener, as the overall quality of the audio heard including frequency, distortion or spatial aspects. In order to create a reference, the listeners were played the four audio excerpts before the encoded audio began. They were then told to reference each as full marks in each category. In both of the two qualities the listener was ask to mark the quality out of 10. In overall sound quality 10 represented a sonically perfect sound, while 0 an unlistenable representation of the sound. The instructions that the listeners were given are shown in appendix A.

9.5 The Listening Environment

All of the conducted listening tests took place in a studio quality room with a noise rating curve of NR 25. The playback speakers where placed at strict positions to insure the listening position specified by ITU-R BS. 775. The listeners where then sat in front of a monitor and given the game console controller so that they could control the three dimensional character on screen. This is shown graphically in figure 9.2. The five loudspeakers used were aligned so that the same relative SPL was produced by each speaker at the listening position. This calibration was done with the use of pink noise, a portable sound pressure level meter, and a high quality reference microphone.

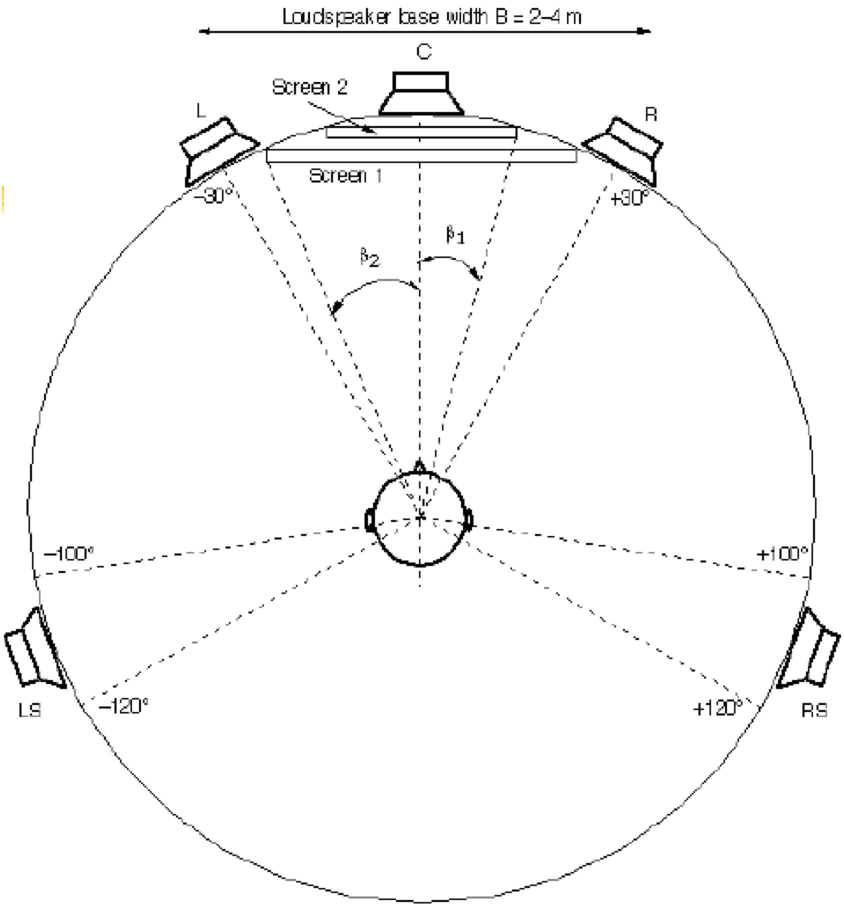

Figure 9.2: ITU-R BS.775 Standard Listening Room [ITU 1993]

9.6 Conducting the Experiment

Before the test began, listeners were presented a printed set of instructions, given time to read them through thoroughly (appendix A), and asked if they had any questions regarding the test. After this they were shown through to the listening room and introduced to the very simple movement controls of the "test level" game. They were then allowed a few minutes to familiarise themselves with the layout of the level. During the familiarising session audio output of the console was set to mute. Finally the "test level" was reset and the listening test initialised.

9.7 Summary

Experimental procedures were instigated, in order to investigate if the system developed above could be incorporated into a computer game streaming system. A formal listening test was set up in which ten experience listeners were required to judge the overall sound of the streamed decoded audio. The listeners were first played the original 5 channel audio excerpts to create a reference for the given marks. The four audio excerpts were kept consistent between the developmental and experimental stages.

In the section 10 the results of the above experimentation will be shown and analysed, in order to create a conclusion towards the hypotheses stated in section 6.

10. Experimental Results and Analysis

The results below represent an investigation into the subjective suitability of the developed real time decoder, in the application of computer games. A small range of marks was used, because it was decided the listening would be simple to partake in. The listeners used were picked for their knowledge and experience of computer games and not audio. This particular method of testing was chosen because a subjective view of the audio in the context of the computer game was needed to be assessed.

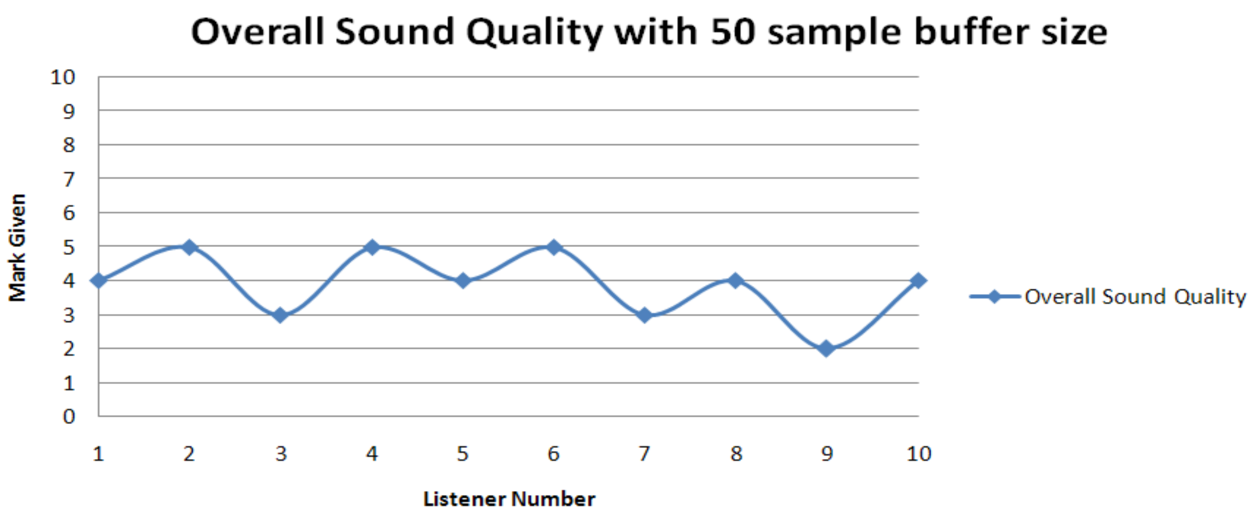

Figure 10.1: Overall Sound Quality of Produced Audio using a 50 sample buffer size.

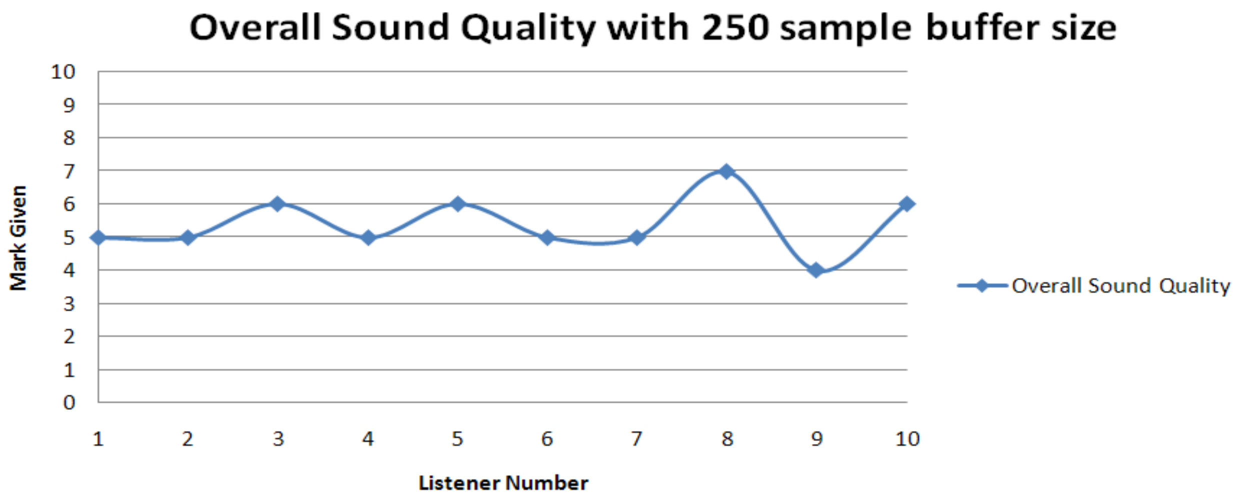

Figure 10.2: Overall Sound Quality of Produced Audio using a 250 sample buffer size.

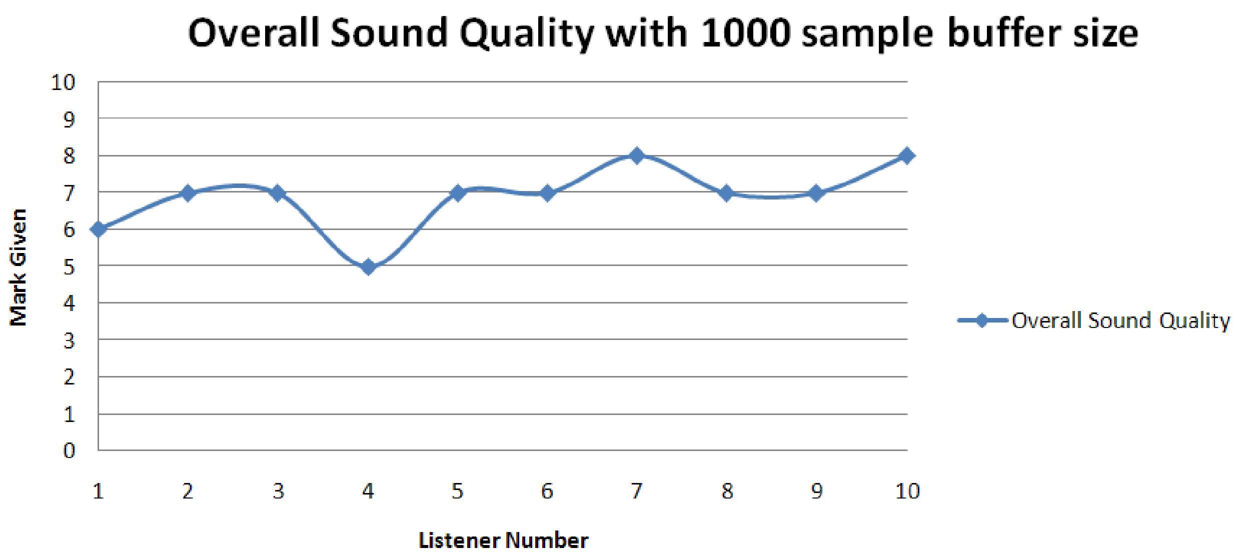

Figure 10.3: Overall Sound Quality of Produced Audio using a 1000 sample buffer size.

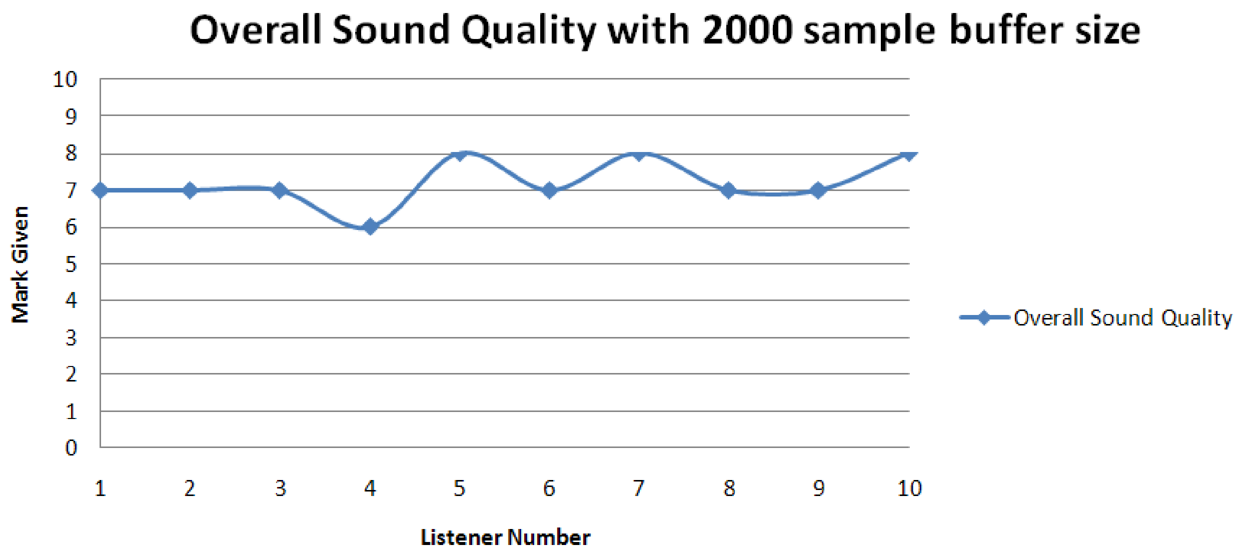

Figure 10.3: Overall Sound Quality of Produced Audio using a 2000 sample buffer size.

It is clear to see from the above results that the perceived quality of the audio rises as the buffer get larger. This is due to the large amount of cross over between the "snapshots" of audio, therefore creating left distortion. The results could further be improved by the use of a larger group of listeners. It would still though, even with a larger group, be important to continue the use of listeners with experience in player computer games. This is because they have the most ability to judge the audio within the context.

11. Discussion and Conclusion

The development of next generation computer game consoles, together with the wide use and availability of 5.1 surround sound systems in the home, has created the need for a high quality encoding method. This method needs to further the current surround implementations of 3:1 stereophony, and provide the high envelopmental qualities required. In this project a complex real time decoder was hypothesised. It is hoped that this newly developed system of encoding and decoding, could provide a real time solution for streaming computer game surround audio.

11.1 Discussion of Results

The overall pattern of the results suggested that the real time program developed would be suitable for use as a real time decoder, this meet with the first of the stated hypotheses in section 6. The hypothesis was not meet fully though due to the program using a large amount of system resources which, meant that even though it could be run on a current generation console, it had no commercial value. Results from the experimental section proved that the created decoder produced a subjectively pleasing sound upon play back when used with a high buffer setting. The problem with having the high buffer setting was that latency up to 245ms was created which is known to be noticeable. Overall the second hypothesis was met, but only for applications where the original audio data was not synchronised with the visual, such as ambience recordings.

11.2 Suggestions for Further Research and Development

In the context of multichannel audio reduction the Karhunen-Lòeve Transform is still relatively new, cutting-edge technology. Therefore it is clear to see that additional research is needed within many areas of the method. The most immediate area that would highly benefit from additional research, is the perceptual effect of the temporal-adaptive KLT. Further research in this area could produce a more efficient perceptually optimised version of the real time system created here. If this system was developed further in the future it could be incorporated into High Definition Television or internet surround sound encoding systems. The KLT audio data reduction system easily lends itself to multichannel audio broadcasting, due to its complex encoder and relatively simple decoder.

References

Altizer, Roger, 2007: "PLAYSTATION 3 (PS3) Specifications and Details" http://playstation.about.com/od/ps3/a/PS3SpecsDetails_3.htm, accessed 05/02/07.

Bennett, Hugh, 2006: "The Authoritative Blu-ray Disc (BD) FAQ" http://www.emedialive.com/Articles/ReadArticle.aspx?ArticleID=11392, accessed 05/01/07.

Blauert, Jens, 1997: *Spatial Hearing* (MIT Press).

Blu Ray Founders, 2004: "White Paper Blu-ray Disc Format", *Blu Ray Founders* (August).

Bosi, Marina, 2000: "Multichannel Audio Coding and Its Applications in DAB and DVB", *Presented at IEEE 5th International Conference* (August).

C++ Library for Audio and Music, 2007: "CLAM Main Website" http://clam.iua.upf.edu/, accessed 05/01/07.

Dressler, Roger, 1998: "Dolby Surround Pro Logic II Decoder – Principles of Operation", *Dolby Laboratories*.

Dressler, Roger, 1997: "Dolby Surround Pro Logic Decoder – Principles of Operation", *Dolby Laboratories*.

Everitt, Brain and Dunn, Graham, 2001: *Applied Multivariate Data Analysis* (Oxford: Oxford University Press).

Field, Andy, 2005: *Discovering Statistics Using SPSS* (London: SAGE Publications).

Gerzon, Michael, 1974: "Surround Sound Psychoacoustics", *Wireless World*, Vol. 80, pp.483-486.

Griesinger, David, 1990: "Recording and Reproducing the Space of Audio", *Presented at AES 8th International Conference: The Sound of Audio* (April).

Griesinger, David, 1997: "Spatial Impression and Envelopment in Small Room", *Presented at 103rd AES Convention, New York* (September)

Griesinger, David, 1998: "Spaciousness and localisation in listening rooms and their effects on the recording techniques", *AES,* Vol. 37, pp. 255-268.

Henning, Lars, 2006: "The Perceptual Importance of the Karhunen-Lòeve Transform in Multichannel Audio", *Final Year Tonmeister Technical Project* (March).

Holman, Tomlinson, 2001: "The History and Future of Surround Sound", *Presented at Surround Sound Professional* (March).

Howard, David and Angus, J.A.S, 1996: *Acoustics and Psychoacoustics* (Oxford: Focal Press).

ITU-R, 1993: "Recommendation BS.775: Multi-channel stereophonic sound system with or without accompanying picture", *International Telecommunications Union*.

Jiao, Yu, 2006: "Hierarchical Bandwidth Limitation of Surround Sound - A Psychoacoustical Study", *PHD Transfer Script*.

MatLab, 2007: "MatLab Main Website"
http://www.mathworks.com/, accessed 05/01/07.

Microsoft, 2006: "Xbox 360 Specifications"
http://www.xbox.com/en-AU/support/xbox360/manuals/xbox360specs.htm, accessed 01/02/07.

Morris, Andy, 2007: "Data Streaming Use In Computer Games", *Electronic Arts Uni* (August).

Mued, L et al, 2003: "The Effects Of Lip Synchronization", *Presented at IEEE VIE 2003 International Conference* (July).

Nakayama, Takeshi et al, 1971 : "Subjective assessment of multichannel reproduction", *AES,* Vol. 19, pp. 744-751.

Open Source Audio Library Project, 2007: "OSALP"
http://osalp.sourceforge.net/, accessed 05/01/07.

Pearson, Karl, 1901: "Principle Components Analysis", *The London, Edinburgh and Dublin Philosophical Magazine and Journal*, Vol. 6, No. 2, pp 559-572.

Russell, Steve, 1960: *Spacewar!* (Massachusetts Institute of Technology).

Rumsey, Francis, 2007: Audio Engineering 3, 2, January 2007.

Rumsey, Francis, 2005: *Spatial Audio* (Oxford: Focal Press).

Smith, Lindsay, 2002: "A tutorial on Principal Components Analysis" http://csnet.otago.ac.nz/cosc453/student_tutorials/principal_components.pdf, accessed 01/01/07.

Theile, Günther, 1991: "On the naturalness of two-channel stereo sound", *AES*, Vol. 39, pp. 761-767.

Turk, Matthew and Pentland, Alex, 1991: "Face Recognition Using Eigenfaces", *Computer Vision and Pattern Recognition Presented at IEEE Computer Society Conference* (June).

Woszczyk, Wieslaw, 1993: "Quality Assessment of Multichannel Sound Recordings", *Presented at AES 12th International Conference* (May).

Yang, Dai et al, 2003: "High-Fidelity Multichannel Audio Coding with Karhunen-Lòeve Transform", *IEEE Transactions On Speech and Audio Processing*, Vol. 11, No. 4, pp. 365-380.

Zieliński, Slawomir et al, 2003: "Computer Games and Multichannel Audio Quality - The Effect of Division of Attention between Auditory and Visual Modalities", *Presented at AES 24th International Conference* (May).

Appendix A: Listening Test Instructions

Computer Games Surround Sound Listening Test.

Thank you very much for being kind enough to do help me with my listening test. Please read the instructions below and feel free to ask any relevant question.

INSTRUCTIONS

When you enter the listening room you will be soon a screen and given a games console controller. Then controller will be used to move the character on screen between the four rooms. In each of the four room is a ambience of music source, the source is at a fixed volume within the room, so will not change intensity while you are in the room.

Please take this time now to familiarise yourself with the level layout.

Now you will be played the four audio excerpts that are present within the level, these excerpts will act of references and you should allocate then full marks in the testing below. All the excerpts you hear during the “test level” will be compared to these sounds.

There are two marks that I want you to award the audio heard within the “test level”, these are Overall General Quality, and Envelopmental Quality. They are both rated from 0 to 10 and a recorded on the separate scoring sheet.

Now you have heard the original excerpts, seen the level and read the scoring instructions, I will come in to listening room and reset the level so the test can begin.

Thank you again for the help and taking part.

Remember you have just been KLT’ed!

Appendix B: MatLab Code for an Offline Temporal-Adaptive KLT Multichannel Encoding Algorithm.

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
%   Matlab Implementation of Temperally adaptive KLT processing
%
%   Born: 11-2006 Yu Jiao
%
%   Adapted by Chris Green on 10/02/07
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

clear;    %Clear matlab workspace

% Parameters
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Wave filename and location set up
filename = 'A';
inpath       = 'AUDIO/';
outref       = 'AUDIO/';

% KLT process parameters

Adaptive          = 1 ;             % 1: Temprally adaptive
                                    % 0: Applying to whole audio excerpt
retchan           = 2;              % Number of retained eigenchannels


% Time domain window  parameters

winspan       = 50;                 % Window size [samples], THE VALUE THAT WAS CHANGED
                                    % TO MATCH AUDIO BUFFER

winshift      = winspan/2;          % Window hop size [samples]
wintype       = 1;                  % 0: Hann (squared sin window) analysis window
                                    % 1: Hann (squared sin window) analysis window
framesize     = winshift;           % Frame size [samples]



% Read audio file
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

audioin = [inpath filename '.wav'];
disp(['Processing audio file: ' audioin]);

[input,sfreq,BITS]=wavread(audioin,48000);  % Read file, store sampling frequency and bitrate of audio.


in = input';   % Rotate matrix 90 degress.
in(4,:)=[];    % Remove LFE channel.


[nchan nsamples]=size(in);  % Create information of audio


[nchan nsamples]=size(in);  % Remove DC offset of audio
out= zeros(size(in));
```

```
% Initialization
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


% Prevent some impossible parameter combinations

if retchan > nchan
  disp('retained eigenchannel number exceeds the original channel number');
  return;
  elseif retchan == nchan
  disp('retained eigenchannel number equals to the original channel number. No compression will be achieved!!!');
end

if wintype > 1
  disp('window type is not properly assigned');
  return;
end


% Initiate transform window
%**********************************************8


if wintype == 1
  if mod(winspan,winshift) == 0
    a = sqrt(2*winshift/winspan);
    x = 0:(winspan-1);
    win = (a*sin(x.*(pi/winspan))).^2;
    plot(win);
    %win = [zeros(1,winfrontzeros) win zeros(1,winfrontzeros)];
  else
      disp('window overlap is not good.');
  return;
  end
end

removal=zeros(nchan);
removal(1:retchan,1:retchan)=eye(retchan);   % Eigen channel removal.


if Adaptive ==1 %adaptive
    if mod(nsamples,framesize)==0
        nframes      = floor(nsamples/framesize);
    else
        nframes      = floor(nsamples/framesize) + 1;
    end
```

```
% KLT processing
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


i           = 1;%sample counter
framecount  = 1;%frame counter


disp('KLT processing...');



% Process frame after frame
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

percent=0;

while i + winspan <= nsamples,

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

    for j=1:nchan

  tempin(:,j) = win.*in(j,i:(i+winspan-1)); % Window'ing
  end

  [v d]=CovKLT(tempin);


  eigvalue=v;
  eigenvector= d;
  y = tempin* d;
  z=y*removal;

 buf=y;



        out(:, i:(i+winspan-1)) = out(:, i:(i+winspan-1)) + buf;

        ivout(:, i:(i+winspan-1)) = ivout(:, i:(i+winspan-1)) + d;
```

```
%display the processed percentage

if  percent < floor(framecount/nframes*100) & mod(percent,10) == 0
    disp(['  Processed ' int2str((percent+10)) '% of audio']);
  end
 percent =  floor(framecount/nframes*100);


 framecount=framecount+1;

 i = i + winshift;
end


else   % process the whole audio

tempin = in';


  [v d]=CovKLT(tempin);


  eigvalue=v;
  eigenvector= d;
  y = tempin* d;
  z=(y*removal);
  out=y;

end

iving = size(ivout);

fidA = fopen('AUDIO/A.ikl', 'w'); % Open file stream for binary writing

countA = fwrite(fidA, iving, 'int'); % Wirte amount of inverse information to file

countA = fwrite(fidA, ivout, 'float32');  % Write KLT inverse information to file.

wavwrite(out,sfreq,BITS,'AUDIO/EncodedA.wav'); % Write encoded audio data to a file

fclose('all'); % Close all files
```