


Semi-synchronous Video for Deaf Telephony with an Adapted Synchronous Codec

By

ZHENYU MA

The logo of the University of the Western Cape, featuring a stylized classical building with a pediment and columns.

**A thesis submitted for the degree of Master of Science
in the Department of Computer Science,
University of the Western Cape**

UNIVERSITY *of the*
WESTERN CAPE

Supervisor: William D. Tucker

Date: 20 February 2009

Declaration

I declare that *Semi-Synchronous Video for Deaf Telephony with an Adapted Synchronous Codec* is my own work, that it has not been submitted before any degree or examination in any other university, and that all the sources I have used or quoted have been indicated and acknowledged as complete references.

Full Name: ZHENYU MA

Date: November 2008

Signed



Glossary

3G	Third Generation (of cellular data networks)
API	Application Programming Interface
ASP	Advanced Simple Profile (of MPEG-4)
AV	Audio-Visual
AVC	Advanced Video Codec
BANG	Broadband Applications and Networks Group
Bastion	The building used by the DCCT NGO in Cape Town
B-frame	Bidirectional frame
CABAC	Context Adaptive Binary Arithmetic Coding
CAVLC	Context Adaptive Variable Length Coding
CCITT	International Telegraph and Telephones Consultative Committee
CIF	Common Intermediate Format with pixel size (324 ×288)
Codec	Compression/Decompression or Encoder/Decoder
CR	Compression Ratio
CT	Compression Time
DCCT	Deaf Community of Cape Town,—a local NGO
DCT	Discrete Cosine Transform
DeafSA	Deaf Federation of South Africa, a national NGO
DivX	Digital Video Express
DMIF	Delivery Multimedia Integration Framework
DSCQS	Double Stimulus Continuous Quality Scale
DT	Delay Time
DWT	Discrete Wavelet Transformation
EBU	European Broadcasting Union
fps	frame per second
FTP	File Transport Protocol
GPRS	General Packet Radio Service
HD	High Definition
H.261	H.261 video codec
H.262	H.262 video codec
H.263	H.263 video codec
H.264	H.264 video codec
H.323	A protocol to provide audio-visual communication sessions on any packet network
HTTP	Hyper-Text Transport Protocol
HVS	Human Visual System
ICT	Information and Communication Technology
I-frame	Intra-frame
IM	Instant Messaging
ISDN	Integrated Services Digital Network
ISO	International Organization for Standardization

TLS	Transport Layer Security
TT	Transmission Time
TTY	Teletypewriter or telephone typewriter, a telecommunication device for the Deaf
UDP	User Datagram Protocol
ViSiCAST	Visual for human Signing: Capture, Animation, Storage and Transmission
VoIP	Voice over IP
VQM	Video Quality Metric
VRS	Video Relay Service
WAN	Wide Area Network
WAP	Wireless Application Protocol
WISDOM	Wireless Information Service for Deaf people on the Move
x264	x264 video codec application implementing H.264
x264CLI	x264 Command Line Interface
x264vfw/VFW	x264 video for Windows
XviD	XviD video codec



UNIVERSITY *of the*
WESTERN CAPE

Chapter 3 Methodology	29
3.1 Problem Statement	29
3.1.1 Problems with Synchronous Video Telephony	29
3.1.2 Problems with Asynchronous Video Telephony	30
3.1.3 Latency Issues of Deaf Telephony	30
3.1.4 Quality of Signing Video	31
3.2 Research Questions	31
3.3 Research Methods	32
3.3.1 Qualitative Research Methods.....	32
3.3.2 Quantitative Research Methods.....	33
3.3.3 Software Engineering Methods	34
3.3.4 Method Integration	35
3.4 Disclosure of the Role of Member of Research Group.....	36
3.5 Ethical Considerations.....	37
3.6 Summary.....	38
Chapter 4 Experimental Design	39
4.1 Introduction to the Target Community—DCCT	39
4.2 Iterative Experimentation	40
4.3 User Observation	40
4.4 Function Tests.....	41
4.5 Adaptation Process.....	43
4.5.1 Codec Testing.....	43
4.5.2 Optimization Process	45
4.6 Summary.....	47
Chapter 5 System Design and Implementation.....	49
5.1 User Requirements Specification.....	49
5.2 Requirements Analysis.....	50
5.3 User Interface Specifications.....	51
5.4 High Level Design	55
5.4.1 Login and Presence Module.....	56
5.4.2 Compression and Transmission Module.....	58
5.4.3 Quality-latency Improvement Module.....	60
5.5 Implementation Issues.....	61
5.6 Summary.....	62
Chapter 6 Experimentation and Results	63
6.1 Iterative Process for this Study	63
6.2 Testing Preparations.....	64

List of Figures

Figure 2-1: Synchronous video server architectures	6
Figure 2-2: The Bi-level encoding	9
Figure 2-3: Asynchronous video communication	10
Figure 2-4: The Eyejot service	11
Figure 2-5: MMS architecture of communication.....	11
Figure 2-6: MobileASL project and RoI.....	14
Figure 2-7: Rationale of TESSA	15
Figure 2-8: Video relay service	16
Figure 2-9: Spatial compression example.....	20
Figure 2-10: Temporal compression example.....	21
Figure 2-11: Zigzag scanning in CAVLC.....	24
Figure 2-12: Comparison of codecs: H.262, MPEG-4 and H.264	25
Figure 3-1 : Waterfall model for the system development	35
Figure 3-2: Three-stage research.....	36
Figure 4-1: Optimization experimentation of one codec in adaption process	46
Figure 4-2: Delayed time in asynchronous video communication.....	47
Figure 5-1: The main frame of asynchronous sign language video chat.....	52
Figure 5-2: Notification message	53
Figure 5-3: Recording a video message while playing an incoming video	55
Figure 5-4: Overall structure of the modules in this project.....	56
Figure 5-5: The login and presence module.....	58
Figure 5-6: The compression and transmission module	59

List of Tables

Table 2-1: Compression Standards development timeline	20
Table 6-1: x264 parameters and their characteristics	77
Table 6-2: Summary of x264 parameters configuration result	86



UNIVERSITY *of the*
WESTERN CAPE

Chapter 1 Introduction

1.1 Technology for Deaf Communication

As Information and Communication Technology (ICT) matures, communication services must be improved to meet the needs of all types of users. For some uses, current Video over Internet Protocol (IP) brings unsatisfactory and even unrecognisable quality of video sequences. Such communication does not always meet the needs of Deaf¹ people. Asynchronous video messaging, such as EyeJot (www.eyejot.com), offers Deaf people the ability to send and receive video messages like email. Unfortunately, communicating like this incurs much delay, resulting in slow response. Even though text messaging is popular among Deaf people via cellphone or Internet, but they would prefer to use sign language for communication. Video Relay Service (VRS) attempts to help Deaf users communicate with hearing people in sign language. VRS provides synchronous video and voice services to enable those who use sign language to communicate with hearing people through a relay interpreter across the world via the Internet. However, synchronous video in VRS cannot always satisfy the communication needs of the two parties. Sometimes, the quality of synchronous video is too poor to keep up communication between Deaf users and the relay operator. Furthermore, VRS requires expensive video equipment as well as large amounts of bandwidth. Such costs are far beyond the financial capacity of many Deaf users in South Africa.

1.2 Deaf Video Communication Situations

The use of ICT to support Deaf communication occurs around the world. Many Deaf users have access to text telephony with a device like a telephone typewriter (TTY) or Teldem [18], video information delivery like TESSA [15] (detailed in Section 2.3.1), video messaging like Multimedia Messaging Service (MMS), video chat software like videoconferences and Skype or Camfrog, special purposed videophones in VRS, and video signing on mobile devices like MobilASL [10]. Because sign language is the first language for many Deaf people, they prefer to communicate with hearing people and their peers in sign language instead of text.

¹ The capital letter D indicates the cultural identity of a Deaf person who uses sign language as a first language [20].

design and evaluate a system with asynchronous video using synchronous codecs to achieve semi-synchronous sign language communication for Deaf users? This project, therefore, designs and develops software for Deaf users that enables them to communicate using high quality sign language video. To avoid the disadvantages of traditional real time video, our approach provides rapid asynchronous communication by means of a store and forward (S&F) strategy to preserve the quality of video.

Deaf people might be satisfied with the intelligibility of S&F sign language video, but might not like the increased latency introduced by this approach. The tradeoffs between latency and quality are discussed and given special focus throughout this thesis. We explored methods to ultimately improve video quality and decrease latency as far as possible to achieve semi-synchronous video telephony for Deaf users. This relies heavily on the method used to compress video files. Therefore, this thesis details the process of codec adaptation for asynchronous Deaf telephony.

1.4 Thesis Layout

Chapter 2 presents a literature review on work related to this research. Different types of video telephony are explored with regards to architectures, codecs, transport protocols, and quality measurement. Various video codecs are examined to explore the relationships between video codecs and video quality.

Chapter 3 discusses the problems with existing synchronous and asynchronous video telephony, the latency involved, and sign language video quality evaluation. The research question is presented and research methods are discussed. The methods for this research consist of qualitative, quantitative, and software engineering methods. The integration of these three methods is discussed.

Chapter 4 describes the experimental design. The chapter introduces the target community for this research, members of Deaf Community of Cape Town (DCCT), with whom the field research is performed and a number of end user experiments are conducted. Then the chapter details an iterative experimental process, consisting of data collection, performance experimentation and adaptation experimentation to identify an appropriate codec and subsequently optimize it.

Chapter 5 details the system design. We specify the requirements for this project with

Chapter 2 Literature Review

This chapter explores the architectures, codecs and transport protocols of two types of video telephony approaches and their associated quality measurement. Section 2.1 discusses a synchronous approach to video telephony. Section 2.2 addresses an asynchronous approach to video telephony. Section 2.3 describes Deaf video telephony. Section 2.4 details the codec technology behind video communication.

2.1 Synchronous Video Telephony

Recently, video communication has become popular due to ubiquitous broadband Internet. The protocol that most synchronous video software employs is either H.323 protocol (H.323) [25] or Session Initiation Protocol (SIP) [53]. Both make use of the Real-time Transport Protocol (RTP) [57] over packet-based networks through data digitization and compression. RTP Control Protocol (RTCP) [57] is used for monitoring RTP sessions. Both RTP and RTCP are used with User Datagram Protocol (UDP) [50] as the transport layer and with IP as the underlying network layer. Section 2.1.1 introduces the basic architecture of synchronous video communication. Section 2.1.2 identifies the codecs and transport protocols that a synchronous approach employs. Section 2.1.3 describes quality measurement of synchronous video.

2.1.1 Architectures of Synchronous Video Communication

Synchronous video telephony (shown in Figure 2-1) can be divided into three groups according to the roles that a mediated server plays, namely server-less, server-based, and with-server [56]. Server-less synchronous video communication does not require server mediation. For example, DaViKo (<http://www.daviko.com>) is IP-based multi-port software supporting video conferencing. It does not need a costly Multipoint Control Unit server [41], which is a device used to bridge videoconferencing connections. Therefore, DaViKo is a server-less P2P video over IP application. It is capable of penetrating a firewall and Network Address Translation (NAT) router and can accommodate up to 5 participants.

2.1.2 Codecs and Transport Protocols for Synchronous Video

The current codec standards are designed and developed for the purpose of compression and decompression for digital video. To compress and decompress one frame of a video results in little delay in an entire synchronous video communication. Thus, the codecs used in synchronous video telephony are called synchronous codecs throughout this thesis. Synchronous video telephony may employ any existing codec. For example, DaViKo employs H.264/Advanced Video Codec (H.264/AVC) video encoding and SIP-communicator uses H.263 video codec. Skype uses On2 VP-7 video compression. VP-7 technology is designed to provide superb video at very low bitrates and perform efficiently on low-power processors. A detailed discussion on codecs is in Section 2.4.

RTP and RTCP play an important role during synchronous video communication. RTP defines a standardized packet format for delivering audio and video over the Internet. RTCP provides out-of-band control information for an RTP flow. Additionally, RTCP gathers some statistical information on a media connection, such as the bytes sent, packets sent, packets lost, jitter, feedback, and round trip delay. These metrics are cues for an application to adjust the quality of service (QoS) by limiting flow or changing to a different codec. Skype appears to use its own transport protocol so that it can bridge communication between users who are behind routers, firewalls or NATs. Skype enables NAT and firewall traversal because each Skype node uses a variant of the Simple Traversal of UDP Through NATs (STUN) protocol [54] to determine the type of NAT and firewall behind. Furthermore, a Skype client randomly chooses the port number upon installation and opens a Transmission Control Protocol (TCP) and a UDP listening port. Meanwhile, it also opens TCP listening ports at port number 80, which is reserved for Hyper Text Transport Protocol (HTTP) [4], and at port number 443, which is the port for HTTP over Transport Layer Security (TLS) [52].

2.1.3 Quality Measurement in Synchronous Video

QoS of synchronous video telephony comprises video quality and latency that is incurred during video compression and real-time transmission [55]. The recommendation from the International Telecommunication Union (ITU) Telecommunication Standardization Sector (ITU-T) G.1010 [24] provides guidance on the key factors that influence QoS from the perspective of the end-user. Therefore, multimedia applications can be identified into categories based on tolerance to information loss and delay by considering user expectations.

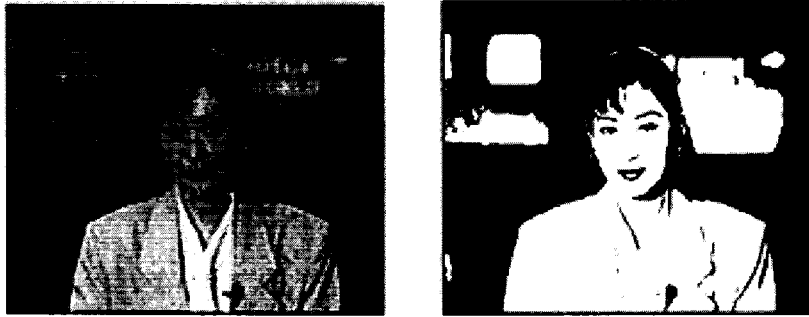


Figure 2-2: The Bi-level encoding

Bi-level video uses only two colours to concentrate the luminance (right) that was generated from a grey-scale image (left) using a simple threshold method. It was also very easy to perceive the facial expression of a person [38].

Delay in synchronous video communication manifests itself in a number of ways, including the time taken to establish a particular service from the initial user request, the time to compress a video frame captured by one party, the time to receive specific information, and the time for playback by the other party. The latency has a direct impact on user satisfaction and includes delays in the terminal, network, and any servers. It depends on the efficiency and complexity of the video compression algorithm as well as network congestion. In LAN applications, the latency is barely identifiable except for the period of time when a communication session is initially set up. Based on RTP, synchronous video communication frequently drops some frames when the network is unable to handle the throughput in a bottlenecked WAN. Consequently, jitter is aggravated by transmission problems. In addition, bit rate affects latency as well as video quality. The use of bi-level video allows video communication at very low bit rates and gives preference to the outline features of scenes when network usage has reached bandwidth constraints. Bi-level video does not provide highest priority to the “basic colour” of an image, as in the case of conventional DCT-based compression method where Moving Picture Experts’ Group (MPEG)-1/2/4 and H.261 video codec and H.263 video codec (H.261/263) were employed (see Figure 2-2) [38]. Even with low bandwidth, bi-level video could still provide clearer shape, smoother motion, shorter initial latency and much cheaper computational cost than DCT-based methods. Hence, it was suitable for small sized devices: mobile handsets, handheld PCs and palm-sized PCs, even in low bandwidth wireless networks.

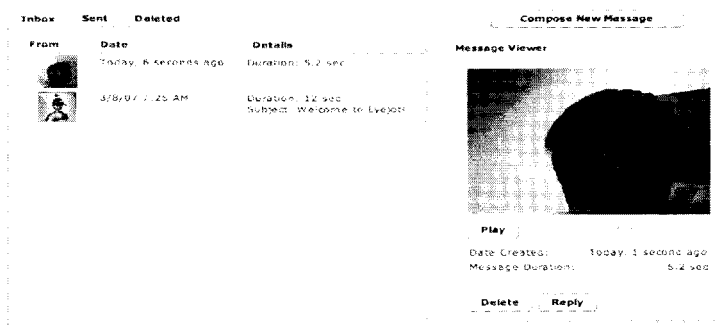


Figure 2-4: The Eyejot service

The Eyejot tool has a similar interface to an email service except that it displays video messages instead of text messages. Therefore, users could treat it like email to record videos and view incoming videos.

The server for EyeJot deals with media storage, video transmission and notification of new incoming video. Therefore, it is a client-server mode of communication. The client does not need to install any software and only requires Internet access.

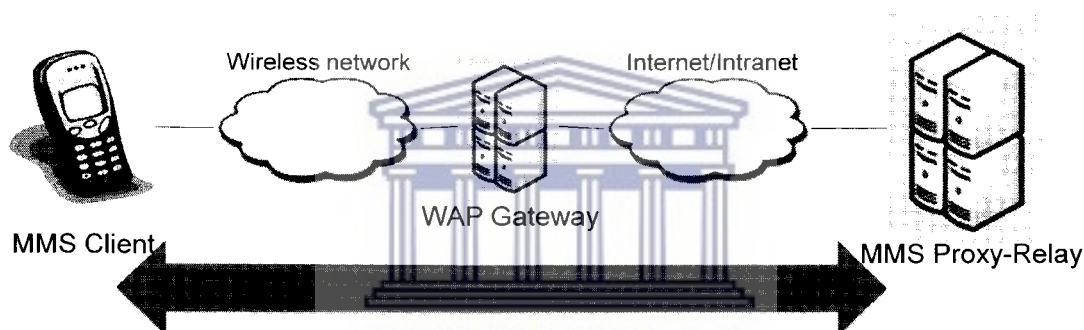


Figure 2-5: MMS architecture of communication

The MMS client interacts with MMS Proxy-relay where the MMS server connects. The operation is consistent with WAP module where MMS Proxy-relay operates as a PULL initiator or PUSH operations (<http://www.wapforum.org>).

Another type of video message is MMS. MMS, as its name implies, provides a rich set of content to subscribers in a messaging context, including clips, graphics, and video clips. With MMS, mobile devices are no longer confined to text messages. MMS is a non-real-time delivery system. MMS provides an S&F usage paradigm and is supposed to be able to interoperate with other messaging systems. A MMS client sends a multimedia message to another client through a Wireless Application Protocol (WAP) gateway and MMS Proxy-relay server, described in Figure 2-5. The MMS proxy-relay is the network entity that interacts with the user mailbox and is responsible for initiating the notification process to an MMS client. The cost of one MMS message is more expensive than that of one Short

from one application to another. Video messaging, for instance, could give rise to a much larger latency, lasting until the receiver updates their message list. Besides, synchronous QoS does not deal with subjective transmission delay that is generated by end users. Therefore, QoS for synchronous video telephony is not applicable to QoS for asynchronous video telephony.

2.3 Deaf Video Telephony

Deaf people can make use of existing synchronous and asynchronous video communication applications. However, the unnecessary voice payload will result in the degradation of sign language video quality. Text can serve as a complimentary source of communication when video is not clear enough. Therefore, Deaf video telephony is different from either synchronous or asynchronous video telephony. This section introduces some applications that are designed and developed for Deaf people in particular. Section 2.3.1 addresses the architecture of Deaf video telephony. Section 2.3.2 describes the codecs and transport for Deaf video telephony. Section 2.3.3 discusses the quality measurement for sign language videos during Deaf communication.

2.3.1 Architecture of Deaf Video Communication

The architecture of Deaf video telephony is different from both synchronous and asynchronous approaches, and is complicated by a variety of factors. Deaf people with PCs can communicate in sign language through Skype, Camfrog and other synchronous video applications mentioned in Section 2.1.1. The Mobile American Sign Language (MobileASL) project [10] marked a new era of sign language video communication by enabling Deaf users to sign with mobile devices in real time.

The proposed outcome of the MobileASL project was to maintain the intelligibility of sign language during communication. MobileASL efficiently compresses the video sequence due to stringent rate constraints and simplifies the compression algorithm enough to reduce power consumption by means of variable frame rate to distinguish signing video and “listening” video. This approach reduces both computation and bandwidth without significantly harming sign language intelligibility [13]. The MobileASL project proposed Region of Interest (RoI) [40] that was based on the experimental result of eye tracking during video perception viewed by Deaf users [45]. The eye tracker tools got information on eye movement patterns of Deaf

virtual human in front of a Deaf person (see Figure 2-7) [15].

The motivation for ViSiCAST was to improve the quality of life for European Deaf citizens by embedding sign language into Deaf people's daily lives. The goal was to widen Deaf access to services and facilities by enabling them to enjoy this kind of communication at large, such as sign language in public services, commercial transactions, entertainment, and learning and leisure opportunities (including broadcasts, interactive television, e-commerce and the WWW) [16]. Since sign language differs from one place to another, the TESSA system provided a limited set of vocabulary in a particular domain [15].

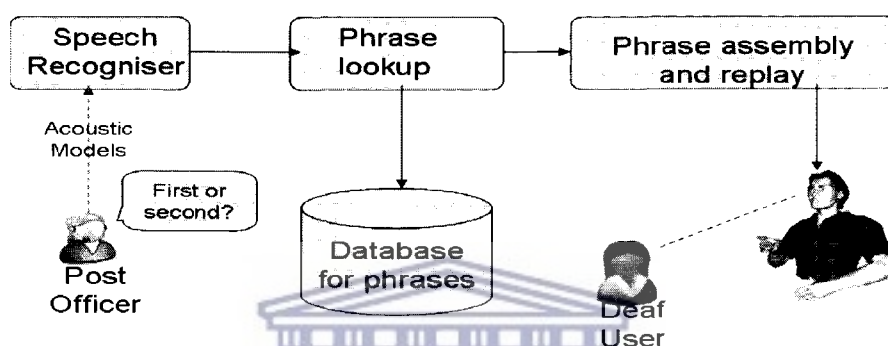


Figure 2-7: Rationale of TESSA

The core parts of TESSA are speech recognition (acoustic model), phrase lookup, assembly and relay. The speech recognition gives a list of relevant phrases according to what a Post Office clerk utters. Phrase lookup determine the closest meaning of the word or phrase from recogniser. Assembly and replay deals with reconstruct the phrase and present it by an avatar to a Deaf person [15].

A combination of synchronous and asynchronous approaches is used in VRS. VRS allows a caller to communicate in sign language with video conferencing in real time with a sign language relay interpreter. The interpreter speaks the signed message to a hearing telephone user through a Public Switched Telephone Network (PSTN) gateway. The interpreter then relays the voice message back to the caller (a Deaf person) by signing. Figure 2-8 shows the simple architecture. Some VRS services offer Voice Carry Over (the video user may speak instead of interpreter speaking), Hearing Carry Over (the video user may listen for him/herself instead of relying on interpreter), and Sign language Preference.

parties in front of the video terminal. Deaf users signed through an Integrated Services Digital Network (ISDN) videophone to a relay operator and the relay operator voiced the translation to the hearing party. TISSA stopped catering for Deaf users soon after the pilot finished [48].

2.3.2 Codecs and Transport Protocols for Deaf Video

Codecs for Deaf video telephony vary from one application to another since several approaches are employed. State-of-the-art codecs, such as MPEG-4, H.263 and H.264 are the main options for synchronous Deaf video telephony. The MobileASL project adopted x264 video codec application (x264) as its video codec to improve the quality of synchronous video as well as lowering the bit rate for transmission via the mobile network. The TESSA project, whose long-term goal was to produce a “text-to-sign synthesizer” [15], did not store video recordings, but used an avatar instead. A number of specific video codecs were developed and deployed as the video relay services were developed, such as Sorenson video codecs for Sorenson VRS.

The transport protocols for Deaf video communication are a composition of numerous synchronous and asynchronous protocols. RTP and RTCP were used for sign video communication over the IP network between a Deaf person and a relay operator in a VRS as well as VoIP between the relay operator and the hearing person through a PSTN gateway. MobileASL employed RTP as well to transmit lower bit rate frames over GPRS or 3G networks. Like asynchronous video telephony, asynchronous Deaf video communication also adopted similar methods other than RTP for transmission such as HTTP, FTP, TFTP and SFTP. There does not appear to be a transport protocol specifically designed for asynchronous sign video transmission.

2.3.3 Quality Measurement for Deaf Video

QoS for Deaf video communication emphasizes the intelligibility of a sign video more than other factors because Deaf people use a rich combination of visual communication methods such as lip reading, hand gestures, facial expressions, body movements and eye movements [30]. Since Deaf video telephony is different from both synchronous and asynchronous video telephony, its QoS should be measured differently. Consequently, with regard to the intelligibility of a signing video, some applications appear to be significantly more interested in RoI and variable frame rate [12]. MobileASL, for instance, provided Deaf users with the

on which the compression and decompression rules are based [66]. Section 2.4.1 gives a brief overview of the development of video codecs. Section 2.4.2 describes video compression schemes and techniques. Sections 2.4.3, 2.4.4, and 2.4.5 introduce the ITU-T and MPEG standards, x264, and Digital Video Express (DivX) and XviD video codec (XviD) respectively.

2.4.1 Evolution of the Video Codec

Video compression can yield high quality video with high frame resolution, high frame rate, low distortion and low cost. The development of digital video technology in the 1980s had seen the possibility of using digital video compression for a diversity of telecommunication applications, such as teleconferencing, video telephony and High Definition (HD) IP Television broadcasting [65]. Only a standard could reduce the high cost of video compression codecs and resolve critical problems of compatibility of equipment from different manufacturers. Standardization of compression algorithms for video was first initiated by the International Telegraph and Telephones Consultative Committee (CCITT) [21]. Digital transmission was of prime importance to the visual telephony and telecommunication industry.

The International Organization for Standardization (ISO) had undertaken to develop a standard for video and associated audio on digital storage media. This effort started with MPEG, a working group of ISO/IEC in charge of development of standards for coded representations of digital audio and video [22]. There are similar compression techniques used among different codecs by means of the cooperative work between standards organizations. Their universal objective is to achieve a generic compression to render a variety of video formats that are accepted by most players. Table 2-1 shows a timeline for major codecs, indicating the timeline of each standard and the development organization involved.

information that is common to the entire file or an entire sequence within the file. While it also looks for redundant information, spatial compression defines an area using coordinates instead of specifying each pixel in the area [75].

Temporal compression (shown in Figure 2-10) deals with related information between frames and discards those that are not necessary for continuity with respect to the human eye. A given background, for instance, is rarely changed, but motions such as head movements and facial area movements differ between frames. In such a case, the compression algorithm compares the first frame, known as a key frame, with the next changed frame, called a delta frame. Only changed information is kept, and a large portion of the file is therefore deleted. If the scene changes, it will tag another key frame for the new scene and continue comparing until the last frame is reached. Consequently, the file size will be comparatively reduced according to the number of key frames [75].

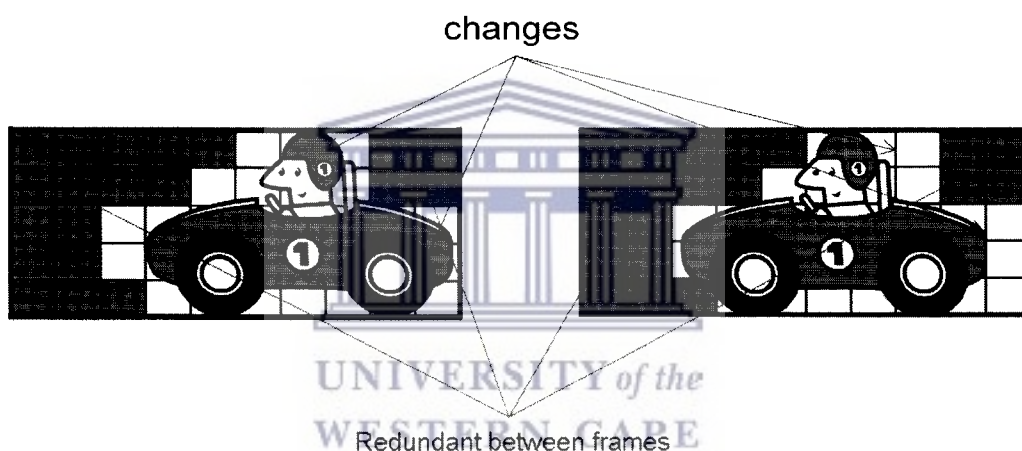


Figure 2-10: Temporal compression example

Temporal compression deals with discarding the redundant information and keeps track of changes between frames. The changed information is for reconstruction when decompression is conducted. This technique is mostly used for moving pictures and it can greatly reduce the file size.

In addition, there are two main transformations used by video compression techniques as well as image compression: DCT [1] and Discrete Wavelet Transformation (DWT) [9]. DCT is a lossy transformation. DCT samples an image at regular intervals after which it analyses the frequency components. DCT then discards those frequencies that do not affect the image. DCT is the basis of many standards. The core matrix of DCT is, for instance, an $M \times M$ matrix [51],

testing—describing procedures for determining the characteristics of coded bit streams and the decoding process.

MPEG-2, also known as H.262 video codec (H.262), addresses a well-established set of encoding and decoding procedures for digital audio and video, formalized as a standard (ISO/IEC 13818) [22]. The significant enhancement it has over MPEG-1 is its ability to efficiently compress interlaced video. MPEG-2 defines a “transport” system rather than just a codec and also defines a bit stream that directly addresses and intervenes in a complicated physical process that allows sounds and images to move further and faster in media networks. MPEG-2 also reorganized relations within and between images and sounds [36].

MPEG-4 provides a wide framework for the joint description, compression, storage, and algorithm of Synthetic/Natural Hybrid Coding [42]. It defines improved compression algorithms for audio and video signals, and efficient object-based representation of audio-video scenes [17]. MPEG-4 plays an important role in multimedia applications over IP-based networks. It comprises all types of media: video, audio, text, and graphic, and it introduces the concept of fully object-based representation with scalability support for each object. MPEG-4 also supports end user interactivity. The MPEG-4 system standard utilises a session protocol called Delivery Multimedia Integration Framework (DMIF) [38] that is conceptually similar to FTP. The return value for DMIF is a pointer indicating the place where a data stream can be obtained rather than a pointer to the data itself. This technology is applicable to mobile and PSTN systems, and it supports videophones, video mail, electronic newspapers and other low bit rate situations.

H.261 was published by the ITU in 1990 and was designed for data rates that were multiples of 64Kbit/s, and was sometimes called $p \times 64\text{Kbit/s}$ (where p is in the range 1-30). These data rates were suitable for ISDN lines for which this video codec was designed. Therefore, H.261 is ideal for two-way communication over ISDN and is based on DCT (see Section 2.4.2) using intraframe and interframe compression [28]. H.263 was a provisional ITU-T standard, and was designed for low bit rate communication. The coding algorithm of H.263 is similar to that of H.261, except for some improvements and changes on performance and error recovery. The differences between H.261 and H.263 coding algorithms include the pixel precision used for motion compensation [29] and the supported resolutions. H.261 uses full pixel precision and a loop filter, whereas H.263 uses half pixel precision. H.261 first

improves the quality of encoding as well as the efficiency by means of zigzag scanning with run-length encoding (see Figure 2-11).

It applies inter-picture prediction, spatial prediction from the edges of neighbouring blocks for intra coding, and lossless Macro Block (MB) coding for precise prediction [11]. As seen in Figure 2-12, H.264 appears to achieve superior clarity of video quality. Numerous synchronous video communication applications have been deployed with H.264 embedded in order to improve real time video quality [31].



Figure 2-12: Comparison of codecs: H.262, MPEG-4 and H.264

Codecs employed in the above pictures are H.262 (left), MPEG-4 (middle), and H.264 (right) [11]. The frame with H.264 compression appears the clearest.

2.4.4 x264 Encoder

x264 is an open source implementation of H.264 standard. x264 is used by many popular software packages such as ffdshow, ffmpeg and MEncoder. According to a recent study, x264 showed better quality than several commercial H.264/AVC encoders [11]. Other results proved that the x264 codec yielded significantly better subjective quality than other widespread codecs such as DivX, XviD and Windows Media Video [70]. x264's high performance is ascribed to its flexibility in rate control, Motion Estimation (ME), MB mode decision quantisation and frame type decision algorithms [63]. Unfortunately, x264 is missing some of the features that H.264 has i.e. switching Intra-frame (I-frame) and switching Prediction frame (P-frame) slices, flexible MB ordering, arbitrary slice ordering, redundant slices, and data partitioning [34].

2.4.5 DivX and XviD

Modern video codecs require flexibility, efficiency and robustness. Both DivX and XviD, based on the MPEG-4 standard, meet these demands. They originated from the OpenDivX project, and subsequently broke into two branches when DivX became commercial software.

flexible reference number, CAVLC and CABAC. x264 implements the H.264 standard and is open source. Therefore, it is used by many players including this research. The next chapter discusses the methodology of this research into how to provide Deaf people with a semi-synchronous sign language video service.



UNIVERSITY *of the*
WESTERN CAPE

Chapter 3 Methodology

Chapter 2 discussed work related to synchronous, asynchronous and Deaf video telephony and their corresponding technologies and QoS. A significant issue facing synchronous Deaf video telephony, discussed in Section 2.1.3, was the quality of video in real time communication. Section 2.2 noted that there were no suitable methods to measure the QoS for asynchronous video communication and that asynchronous latency was a problem. Furthermore, popular synchronous and asynchronous video communication tools do not necessarily fulfil requirements for sign language video communication. This chapter discusses the problems facing Deaf video communication and explores approaches that could provide better communication services for Deaf people. Section 3.1 details the gaps found in both synchronous and asynchronous approaches to video telephony for Deaf users, and emphasises the particular requirements for sign language video quality. Section 3.2 presents the research questions. Section 3.3 proposes methods that can help to answer these questions. Section 3.4 presents the ethical issues for this endeavour.

3.1 Problem Statement

A synchronous approach to video communication for Deaf people hinders sign language comprehension due to the poor quality of video. This approach appears to have good performance in LAN scenarios, but not in WANs. An asynchronous approach is also not ideal due to the delays incurred. Section 3.1.1 discusses the problems in the use of synchronous Deaf video telephony. Section 3.1.2 shows the problems in the use of asynchronous Deaf video telephony. Section 3.1.3 discusses the latency issues in Deaf video telephony. Finally, Section 3.1.4 considers the quality of sign language video.

3.1.1 Problems with Synchronous Video Telephony

Synchronous video applications are not specifically designed for Deaf people. Synchronous communication can generate video with distorted and inconsistent quality during communication due to packet loss that occurs while transmitting over unstable networks. This is especially not conducive to sign language communication. Deaf people cannot comprehend sign language videos due to the variable latency or frequent disconnection resulting in the halt of the conversation [66]. There is, therefore, an urgent need to optimize

continues for a longer period. This thesis proposes to use asynchronous video communication for Deaf telephony so that the quality of the sign language video is retained regardless of network bandwidth. Latency within an asynchronous approach increases with recording delay. Compression time is also increased and is necessary in asynchronous video communication since an entire video file has to be compressed as opposed to one single frame at a time for the synchronous approach. Therefore, this project focuses on the reduction of latency for asynchronous-based video communication to perform what we call semi-synchronous video communication.

3.1.4 Quality of Signing Video

Sign language video communication depends greatly on subtle movements of a signer, e.g. small movements of hand gestures or the eyes. These movements subtly change the meaning via context. Facial expression, gazing direction of the eyes, direction of the eyebrows, lip movements and hand gestures as well as shoulder shrugging help to complete the meaning in sign language. Therefore, sign language video requires intelligibility for these features. In addition, there is no way to conduct a service evaluation for asynchronous video telephony because QoS standards for evaluation of synchronous video communication are not applicable to asynchronous video communication, nor for sign language video, as mentioned in Sections 2.2.3 and 2.3.3. Therefore, we also propose an alternative approach to QoS in order to judge the quality of asynchronous video communication and to evaluate the usability of the system for Deaf people.

3.2 Research Questions

This thesis concentrates on answering one main research question. The research question is explicitly stated as follows:

How can we design and evaluate a system with asynchronous video by adapting synchronous codecs to achieve semi-synchronous sign language communication for Deaf users?

This research question entails two primary objectives: how to build up asynchronous video software to allow Deaf users to use sign language video communication, and how to make this asynchronous communication suitable to achieve semi-synchronous communication with more intelligible video quality and less latency. It follows that there are several subquestions:

complex situations [58]. However, qualitative research methods give us the opportunity to highlight many angles of people-centred situations.

In qualitative research, the main data gathering tools are interview, discussion and user observation. Qualitative research is often characterized by a spiral structure where each phase is based on previous stages. Alternatively, we can describe such an approach as an on-going dialogue between the participants and the researchers through which we continuously improve mutual understanding by means of investigation, data collection and analysis. Interviews improve such mutual understanding and establish a good relationship between the participants and the researchers. User observation give the researchers opportunities to compare different results from different subjects and help to achieve better measurement. The data analysis methods employed during qualitative research direct the researchers to interpret the data from the perspective of the participants in the investigated situation. In other words, interviews and user observation explore an understanding of the boundaries of the researched phenomenon [58]. The qualitative data obtained are integrated into technical system design by means of the total immersion method [6]. The total immersion method simply exposes the members of the development community to their users and has a profound effect on design. The total immersion method indicates no clear boundaries between the participant group and the research group.

This research is based on user observation and investigation of the real world. It emphasizes discovering the patterns and workings—values and behaviours of a particular group or community. It is important to be aware of the interaction and interlink between anthropology and mass communication. Therefore, to observe and discover the Deaf users' values and behaviours improves our understanding of real users.

3.3.2 Quantitative Research Methods

The use of quantitative methods in this project will be to search for the minimized delay time during video communication and identify the factors that improve video quality for sign language communication. There are some specific methods to evaluate the changes of latency such as total objective delay time and overall compressed video quality analysis and assessment. Total objective delay data will be gathered during prototype testing when Deaf participants use our system. The delay data is categorized into three groups: record/play delay, compression delay and transmission delay. During video communication, all the relevant

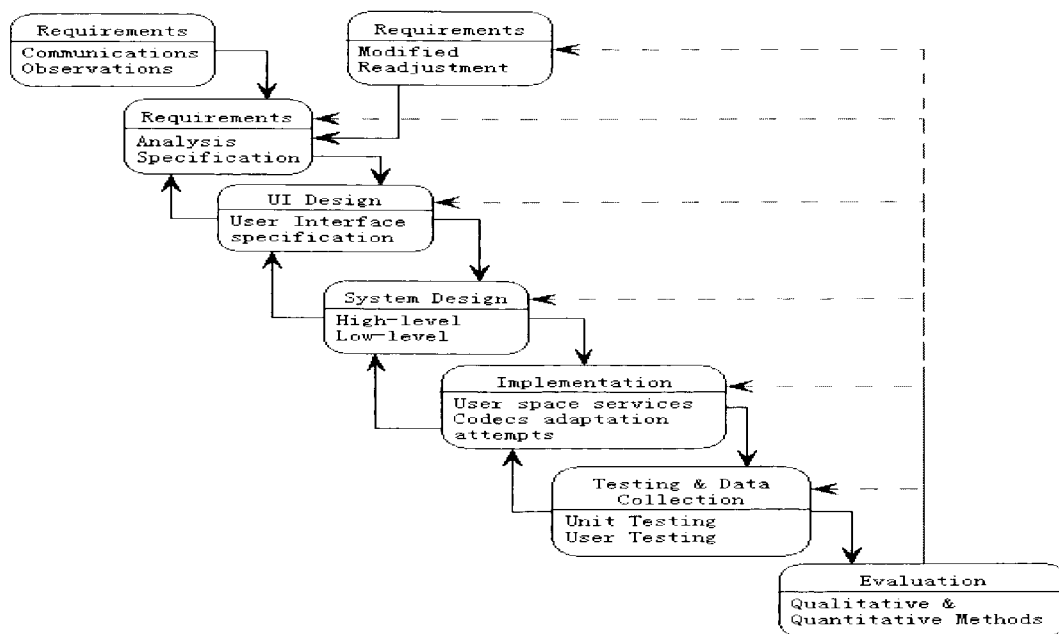


Figure 3-1 : Waterfall model for the system development

The waterfall model in this project is a variant model with cyclical development as well as scenario-driven iterative design. The advantage of this is the gradual achievement of the target by means of user participation and user-driven design and development.

3.3.4 Method Integration

The technical system development methodology in this thesis was a prototyping approach situated within iterative qualitative research that took place with the help of quantitative data collection and analysis, forming an iterative three-stage approach (shown in Figure 3-2). An iteration cycle comprised the qualitative methods for user requirements capture and analysis, exploratory prototyping design and development based on the analysis from the previous stage, and quantitative methods for objective data collection and a subsequent qualitative evaluation from the user side. The outcome of each iteration process became an input of next part of the process and then started the next iteration.

The first stage of this research was also a combination of qualitative and quantitative methods with a focus group—a group of participants representing the whole community and culture [35]. The research began by using a qualitative tool, an open-ended questionnaire pertaining to user requirements that was analyzed using content analysis techniques and that enabled us to identify relevant variables and information related to experiments. Content analysis was a data analysis method used to determine the presence of certain words or concepts within a context or sets of contexts [58]. The second stage of this research

lecturers, supervisors and postgraduate students at the University of the Western Cape whose aim is to design, develop and evaluate IP based, multi-modal semi-synchronous communications to bridge the digital divide in South Africa. The author is a member of that research group. Only the author is responsible for the work reported in this thesis.

3.5 Ethical Considerations

Since this research employed a user-centred approach with Deaf people, we had to consider whether our research was likely to cause physical or emotional harm to participants, such as violating their rights to privacy by posing sensitive questions or by gaining access to records that contained personal data. Therefore, an informed consent sheet (see Appendix A & B) was required before the study and interviews began in order to help users understand what would be done with them. However, the explanation of this project was supposed to be brief and clear so that they could easily understand. On the other hand, we had to learn about the culture of the Deaf community to ensure that norms and culture were respected to establish a good relationship with the participants.

Thus, we explained the entire process of the research to each participant with the assistance of an interpreter who was recommended by Deaf Federation of South Africa (DeafSA) (www.deafsa.co.za). All SASL interpreters were bound by a code of ethical practice that incorporated the necessity for confidentiality and the censure of discussing information gained during interpretation sessions [20]. Each participant was made fully aware of the risks and benefits of the evaluation and had fully understood the information sheet (see Appendix A) where an introduction of our motivation, methods, experiments and evaluation for this research were described. A sign language video of the contents of the informed consent form was recorded by an interpreter. The interpreter also translated the contents of the consent form into SASL if any of the potential participants had difficulty accessing the written content. After that, each person was asked to sign a consent form (see Appendix B) on agreement. In order to respect the autonomy of each participant, voluntary participation was emphasized.

The identities of the participants in structured interviews were protected by the researcher by means of storing video files in a password protected PC during the period of transcription and analysis. A randomly ascribed number identified individuals in transcriptions. All recorded files were destroyed as soon as the analysis was complete.

Chapter 4 Experimental Design

The experimental design concentrates on the approach presented in Chapter 3, and describes how to answer the research questions from Section 3.2. This chapter details how to collect the data, how to analyse the data, and how to use the experimental data to investigate the research questions. Section 4.1 introduces the target community with whom most of the experiments were conducted. Section 4.2 discusses the iterative process described in Section 3.3. Section 4.3 describes data collection. Section 4.4 discusses performance experiments for asynchronous Deaf video communication. Section 4.5 describes the adaptation process for a synchronous codec to make it suitable for asynchronous sign language communication.

4.1 Introduction to the Target Community—DCCT

This thesis focuses on a real-world environment with real problems rather than an in-lab experiment. The target community for this project was a group of disadvantaged South African Deaf people characterised by Deaf cultural pride, illiteracy, physiological impairment and underemployment [19]. Therefore, this research aimed at understanding the social space of Deaf people in their environment. We conducted academic research to solve a real problem for this community.

We have been associated with the target community, DCCT, for almost five years. DCCT was founded in 1987 and is a non-governmental welfare organization. DCCT attends to the needs of Deaf people in the Western Cape. DCCT's historical function was to serve the black and coloured Deaf community whose needs were neglected in the past by national bodies serving Deaf people in South Africa. This organization has an active Deaf membership of over 1000 members. The Bastion is the name of the building used by the DCCT, and it acts as a cultural and educational centre. DCCT provides skills training and social services for unemployed Deaf people in order to improve their lives. Every week there are multiple English literacy classes to bridge the gap between Deaf and hearing people in terms of their education background. A small computer lab at the Bastion provides ICT services to Deaf users free of charge.

We visited DCCT each week, communicating with them via an interpreter to discuss research issues regarding improvement and progress made. We also helped Deaf lab managers fix network and PC errors. We helped them maintain their network and update

User observation gave us an opportunity to understand the organization (community) deeply and thoroughly, as well as the broad context in which Deaf people worked [44]. As we involved ourselves with the target community, the experience enriched our thoughts and equipped us with an informed view of the target environment so that we could design a better and more appropriate application. Thus, this method required us to spend long periods with Deaf people in the field.

This research obtained experimental results continuously, analysing users' feedback with each iterative session. We did this on a weekly basis, conducting experimental tests and data collection. Every visit strengthened the relationship between the Deaf participants and us. The close contact and continuous interviews helped form the user requirements of the system design. At the beginning of this research, we spent much time in their working and living environment, observing the way they accomplished their tasks and talking with them to learn about their daily lives. Having observed their behaviour, we finalized the user requirements for this research. During the user observation, questions were also asked of the users to provide clarity on their work and lives. Based on user observation, notes were taken in each meeting and discussion. The notes helped record the updated requirements and document reflection for the next iterative process. The data collected from user observation were triangulated with subjective and objective results described in Sections 4.4 and 4.5 to provide a broader understanding of both the social and technical aspects of the research question.

4.4 Function Tests

Function tests verified that the system provided asynchronous video communication between Deaf people by using sign language. We checked the system functionality of login and presence service, video file storage and transmission as well as other user functionality on capturing, playing and overall usability.

In order to allow the login process to work on both LAN and WAN, we designed and developed two different types of login to tackle different networks. Both LAN and WAN versions were designed on P2P communication with the help of a login server. The login server registered users and notified the clients with corresponding packets by which the connection information for communication was encapsulated. Therefore, both versions were with-server asynchronous applications (as explained in Section 2.1.1). The login server

decrease the incurred delay. Recorded information and usage did not violate the privacy of the participants according to the ethical considerations discussed in Section 3.4.

The unit tests were conducted during function tests. The compression process, for instance, was tested by different codecs for video quality. The compression unit test was an individual part of this whole system to verify the video compression techniques. The system provided an interface for different codec plug-ins and each candidate codec generated compressed video files that were evaluated by both objective and subjective methods. Similarly, there were several other unit tests such as login, video capture and transmit, and notification test. Most of the data from unit tests were recorded to a log file for analysis and messages were displayed on a user interface to verify the success or failure of a given unit test.

4.5 Adaptation Process

This research was meant to determine the codec that was the most suitable for asynchronous video communication, and optimize the tradeoffs between the quality of video and latency to achieve semi-synchronous sign video chat. The appropriate codec was chosen by means of subjective assessment and objective metrics measurement. Refinement and reconfiguration was then performed on the appropriate codec to evaluate its eligibility for asynchronous adaptation. Therefore, the experimentation comprised two phases, namely, codec testing to identify the most appropriate codec, and optimization testing that was dedicated towards configuration of the codec parameters so as to be able to apply it to asynchronous sign language video communication. Section 4.5.1 describes the details of the codec testing effort design. Section 4.5.2 introduces the optimization testing design.

4.5.1 Codec Testing

The attempt to choose the best codec for asynchronous purposes proved highly challenging. Each existing codec has a unique set of advantages in terms of either compression ratio or compression time that depend on the complexity and efficiency of its compression algorithms. H.264, DivX, and XviD have good reputations for their video quality and compression algorithms that were discussed in Sections 2.4.4 and 2.4.5 respectively. We compared these three codecs through a series of subjective and objective assessments to identify the most suitable one for use in asynchronous sign language communication.

For subjective assessment, MOS on the quality was determined by means of questionnaires,

where \bar{x} , \bar{y} , σ_x , σ_y , and σ_{xy} are respectively the estimates of the mean of x , mean of y , the variance of x , the variance of y and the covariance of x and y . C_1 and C_2 are constants that stabilize the division with weak denominator. The value of SSIM is between -1 and 1 and has the best value of 1 when $x_i = y_i$ for all values of i , which means the compressed video has the exact same structure, luminance and chrominance contrast as the original one [74].

Another objective measurement for perceived video quality is VQM, developed by the Institute for Telecommunication Science [49]. VQM measures the perceptual effects of video impairments including blurring, jerky/unnatural motion, global noise, block distortion and colour distortion, and combines those factors into a single metric [79]. It takes the reference video and the compressed video as input and is computed by means of calibration, quality features extraction and a quality parameters calculation [79]. The calibration estimates the spatial and temporal shift as well as the contrast and brightness offset of the processed video sequence with respect to the original video sequence. Quality features extraction extracts a set of quality features that characterizes perceptual changes in the spatial, temporal and chrominance properties from spatial-temporal sub-blocks of video streams using a mathematical function. The quality parameters calculation computes a set of quality parameters that show perceptual changes in video quality in comparison to those extracted from the original video. Consequently, VQM is computed using a linear combination of parameters calculated from a quality parameters calculation method.

Fortunately, the Video Compression Group from MSU has deployed a battery of complete assessments on objective measurement as well as a subjective perception-testing suite for video [70]. We made use of MSU's automated suite to evaluate the quality of compressed sign language videos. Compression ratios and delay times were also compared between the candidate codecs to reveal the tradeoffs between the quality and latency. The results from both subjective assessments and the objective evaluation determined the codec to be integrated into our system for Deaf users.

4.5.2 Optimization Process

The optimization process was based on the results obtained from the experiment described in Section 4.5.1, and was dedicated towards optimizing the codec algorithm through modifications on parameter configurations to find the best combination of parameters that provided best quality at the lowest latency. Figure 4-1 illustrates the optimization process on

the interval of time after they received a new video file, and before they played that file.

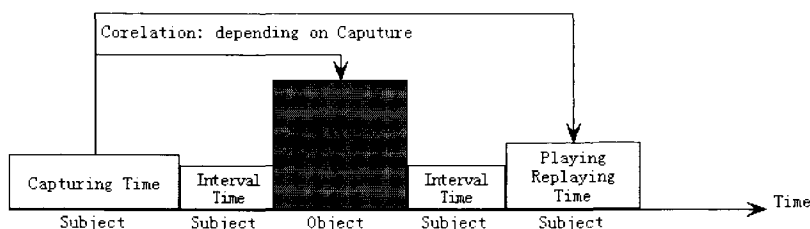


Figure 4-2: Delayed time in asynchronous video communication

Delay in asynchronous video communication comprised two aspects: objective delay and subjective delay. The components of objective delay are compression time and transmission time. Both are controlled by the compression complexity and the efficiency of algorithm. The subjective delay, on the other hand, is dependent on end users. Such delays are correlated interactively. Video capture time, for instance, affects the compression and transmission time due to the file size of the captured video, and consequently affects the playback and replay time.

As shown in Figure 4-2, objective delays, compression and transmission times, were technical delays that could be reduced with good design and development. Subjective delay, however, formed most of the delay and was difficult to tackle. Video capture time affected compression and transmission times, as well as playback and replay times. One way to reduce capture time was to ask Deaf participants not to record a long video message. Accordingly, the system provided a notification scheme that gave a short message to remind Deaf participants not to record long messages. In order to avoid interrupting Deaf people, the notification message was displayed in a message box instead of in a window popup, as detailed in Section 5.3. The interval time could not be removed because of the necessity of a period of thinking time for the Deaf participant to perform the next step. Therefore, the notification service could shorten the interval time. Once the video capture time was reduced, the compression time, the transmission time and the playing time would all also be reduced.

4.6 Summary

The experimental design focused on the feedback from Deaf participants of DCCT. It was an iterative process that combined user observation, where user requirements were gathered and the feedback of our system were collected; function tests, where the asynchronous sign language video communication was conducted positively and actively; and an adaptation process, where the most appropriate synchronous codec was chosen to suit asynchronous video communication. The adaptation process comprised two phases: a codec test, where the

Chapter 5 System Design and Implementation

We developed software prototypes to carry out the experimental design. The prototypes addressed the issues of sign language video quality and latency. In this chapter, Section 5.1 specifies the user requirements. Section 5.2 gives an analysis of these requirements. Section 5.3 describes the user interface specification of the software. Section 5.4 describes a high-level design and provides an outline of the system modules. Lastly, Section 5.5 indicates some implementation considerations.

5.1 User Requirements Specification

The user specification comprises two types of requirements. One is for Deaf users and the other is for the research. Deaf users' requirements concern the sign language communication aspect and the research's requirements concentrates on collecting data on latency and the performance of video codecs.

A previous system built for DCCT members was called the SoftBridge for Instant Message Bridging Application (SIMBA) [64]. SIMBA, built in 2004 by another BANG student, was a semi-synchronous voice relay system and bridged communication between a hearing user and a Deaf user. However, SIMBA failed to achieve take-up because Deaf users preferred to use sign language instead of text and even SMS. SIMBA was not tested for a specific communication task and provided complicated setup procedures that made it difficult for Deaf users to use on a PC. Furthermore, Deaf users could only use it at the Bastion.

We found that Deaf people complained about the quality of video with a number of Internet-based tools or the cost of communication that synchronous video telephony provided. The Deaf participants also complained about the loss of video quality when enlarging the video display window of Camfrog, and they preferred the small sized video screen that the free version of Camfrog provided instead. On the other hand, when they were waiting for a video message response from Eyejot, Deaf participants were mostly doing nothing, but surfing the Internet. It seemed that the latency in asynchronous communication made them tired of waiting for a response.

From user observations and regular interviews, we found that Deaf users preferred to communicate in sign language rather than typing text messages on computers or mobile devices. They wanted a clean and simple user interface instead of complicated multiple

system could be available for Deaf users everywhere.

Deaf users should not be concerned with the recording, compression and transmission processes. They want to be far away from how these processes work. They would like to see a quick response after they click the buttons. Therefore, the recording, compression and transmission processes are technically hidden from the user interface except for showing response messages to Deaf users so that they can know what is going on and what is coming up next. All videos, both recorded and compressed, need backups with identification tags for subsequent quality analysis and all events related to time consumption must be written to a log file for latency calculations. The system requires capture, compression, transmission and playback unit tests, as described in Section 4.4, to examine the functionality of communication.

Deaf users are most concerned with the video quality of signed communication. Section 3.1.4 described the characteristics of sign language video and indicated that the intelligibility of video was the most significant factor in Deaf telephony communication. The system should provide several codecs used for video compression so that Deaf users can determine which one is the best for asynchronous sign language video communication. Eventually, the system will adopt one codec considered as the best from both users' perceptions and technical aspects, and optimizes it in order to maximize video quality and minimize latency. The latency issue is also a significant factor to satisfy Deaf users. However, the subjective delay that is brought out by Deaf users can be slightly reduced by notifying the user with an informative message or warning sign in order to shorten the delay during the communication. The objective latency, on the other hand, is optimized to minimal through a rich set of evaluations. The quality-latency tradeoffs are optimized to serve the asynchronous communication.

5.3 User Interface Specifications

DCCT members have been disadvantaged in their educational background and ICT knowledge. Consequently, this system was designed for simple access to avoid complicated steps to get it running. The system had to be easy to understand for Deaf users without a large amount of ICT training. Since an asynchronous video communication introduces large latency, Deaf users might be faced with long waits for responses. Therefore, the system

online contact person list. Figure 5-1 depicts the basic screen, its buttons and the auxiliary messages to make the interface reasonable easy to use. In addition, the notification functionality helps Deaf participants with their communication. The *LogIn* button, for instance, is the only active button if a user has not logged in yet. After a user logs into this system, it becomes inactive. Meanwhile, *logoff* button become active, indicating that the user is now online and is available to others. User name, automatically given by default, may be changed by users themselves. Furthermore, it is a unique tag to identify oneself to others online.

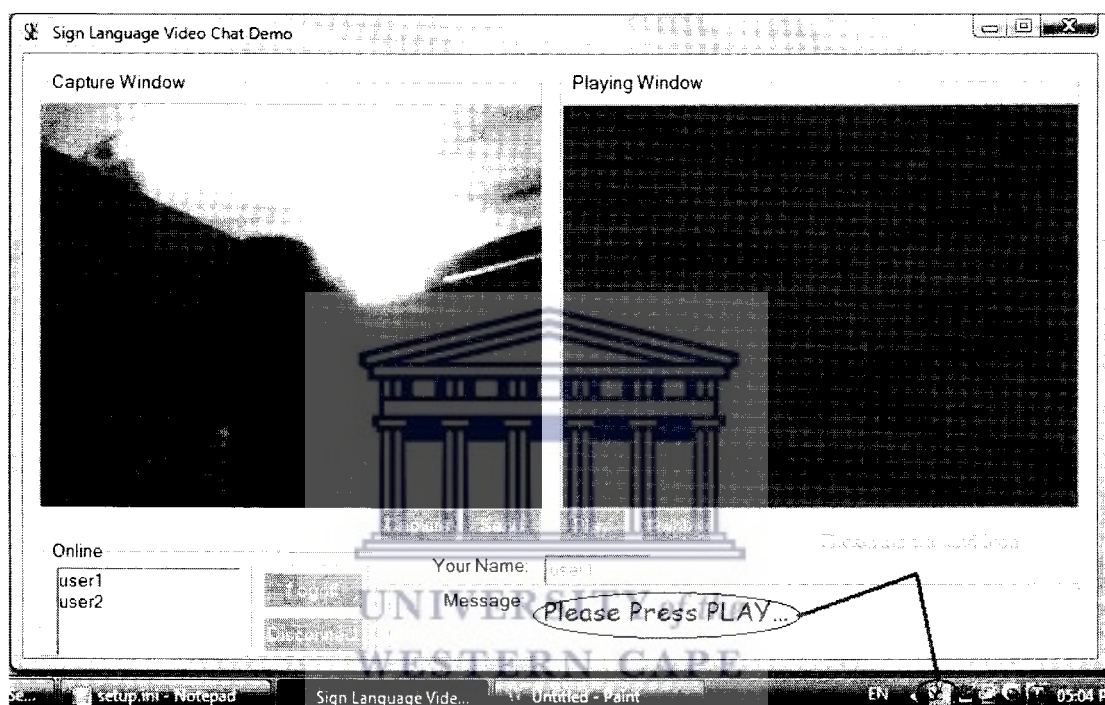


Figure 5-2: Notification message

The notification service informs Deaf users of what is going. Among those messages, the new incoming video message is the most significant because the delay for responding to this message may postpone communication between the two parties and impede the communication from going further. The system utilizes flickering icons and messages to notify the user. The system avoids using any popup window to keep the interface simple.

Once a user logs in, the user name appears in the *Your Name* text box (see Figure 5-1), and the *LogIn* button turns into a *LogOff* button. The *Connect* button is activated to allow a user to contact other users who are displayed in the *Online* list box. There are three choices to initiate the connection process. The first is simply to double click a user's name in the *Online* list. The second is to single click the user's name in the *Online* list, and then click the

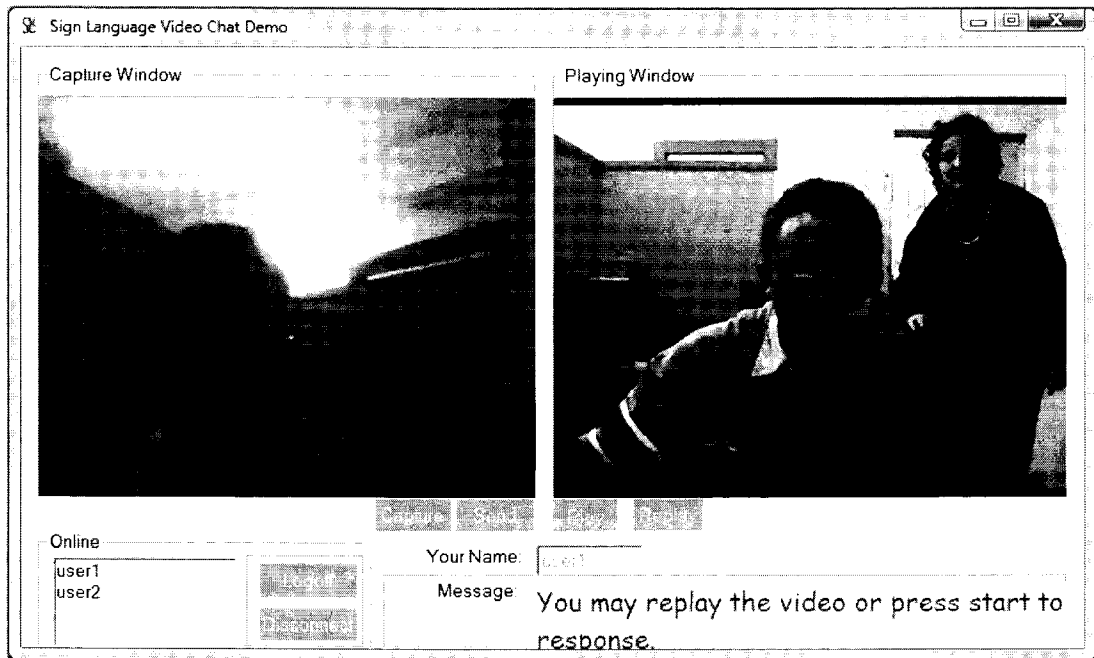


Figure 5-3: Recording a video message while playing an incoming video

The system allows a user to record a video message (see the left panel), while he/she is playing an incoming video (see the right panel). Employing S&F method, the system records a video message to a local place and then transmits it. An incoming video message is also stored locally. Therefore, the recording and playing processes are independent.

5.4 High Level Design

An early prototype of asynchronous video telephony for Deaf users was developed in Java Media Framework with a limited number of video formats [43]. It was designed for P2P in LAN to be able to accommodate two persons to communicate. The system for this project supports the latest codecs in order to provide better quality with less delay. This project's system design involves the following primary modules: a login and presence module, a video compression and transmission module, and a quality improvement module. Figure 5-4 illustrates all of the modules and their relationships. Section 5.4.1 describes details of the login and presence module. Section 5.4.2 addresses the compression and transmission module. Section 5.4.3 discusses the quality-latency improvement module.

a presence service to distribute users' online/offline status to others by means of broadcasting status information. The login server provides a virtual structure that stores all the usernames and IP addresses for online users.

Once the login server receives a login request, it constructs a map structure that maps the username to the requester's IP address and subsequently sends the IP address back to the requester. This is hidden from the user; the requester does not need to know its own IP address. Having received an IP address from the login server, the requester side then writes this IP address into a local file, *setup.ini*, where all information concerning the presence server's address, FTP's information and preset paths for video files are defined. This simplifies the user configuration process. Furthermore, this helps the system penetrate firewalls network and traverse NAT structures in some complicated topological networks as long as the IP address for presence server is public². This is different from STUN technology used in Skype, discussed in Section 2.1.2. The presence server then broadcasts the entire map structure to each online client node. After that, the login server waits for a new client. Therefore, it is not involved in any other processes other than the login and presence module and is independent of the main application.

The login process is more complicated if a client is behind a router or NAT. If the two communication parties are both within a LAN, for instance, the connection is set up directly between both sides. The login server does not necessarily have a public IP address, because it is in the same network with clients. The purpose of the existence of the login service here was to allow a communication connection request to be forwarded to the correct client by looking up the username and IP address mapping structure. Figure 5-5 illustrates the design of the login service with presence functionality. The only role that the login service plays is to provide a registration for clients, allowing them to be available to other clients. Each time a new user logs in or an online user logs off, the login server updates its virtual structure and consequently the presence functionality that the login service offers distributes online status by broadcasting each online node. The termination of the login server signals the destruction of that map structure.

² With the help of the dynamic domain name system, the IP address of our presence server machine that sits in our LAN appears public by means of the correct configuration for a router with port forwarding.

similarities in terms of compression effectiveness and algorithm complexity.

The latest codecs, such as DivX, XviD and x264, all based on the MPEG-4 standard, were selected as candidate codecs and were plugged into the system for the adaptation process. The codec with the most advantages would eventually be taken as the compression tool for this system and would be consequently carried on to the next step of the adaption process: optimizing that codec and integrating it into our system.

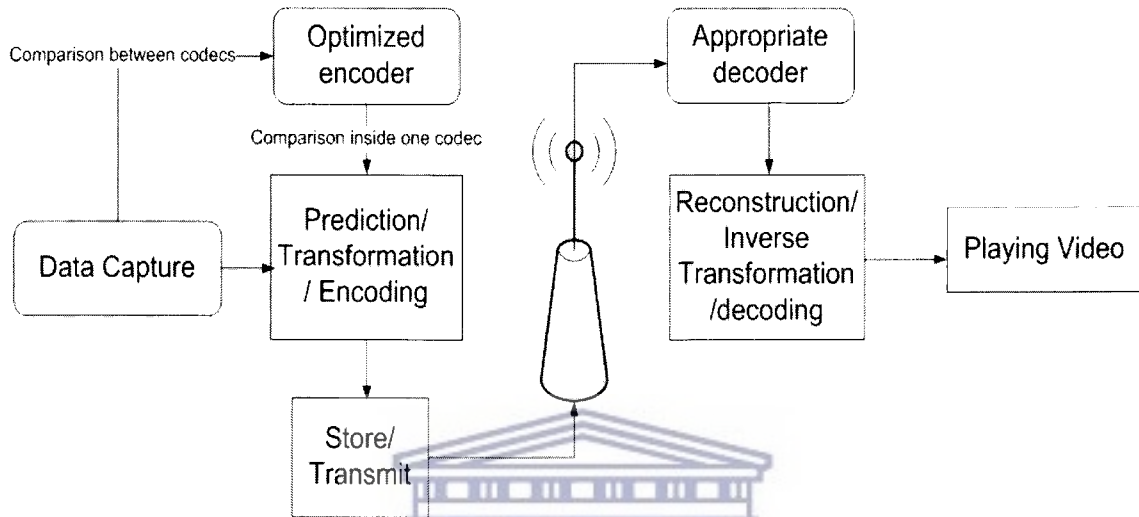


Figure 5-6: The compression and transmission module

The compression and transmission module comprises adaptation, capture, compression, transmission, and playback processes. Among these processes, the adaptation process, the most significant one, is used to compare different codecs to find one best suited to asynchronous video communication in sign language. It also compares different configuration parameters of a given codec to determine quality-latency tradeoffs to achieve higher video quality and less asynchronous delay. The capture process deals with capturing a video source and writing it into files. The compression process makes use of the codec to compress a video file. The transmission process fulfils video file transfer. The playback process is simply the playback of a received video file.

Figure 5-6 is a flowchart of the compression and transmission module. This module provided interfaces so that different codecs could be plugged in and different configurations of a given codec could be optimized inside the system. The transmission method adopted FTP that had been shown to offer better performance than TFTP and SFTP in our previous project [43]. FTP transmission allowed a video file to be transmitted via any Internet topology. In addition, FTP transmission had both active and passive modes so that the packets could penetrate a firewall or NAT. The implementation of the video file transmission was not related to the login server because this application was pure P2P.

The quality-latency improvement module has two responsibilities: the codec testing and the optimization experimentation, explained in Sections 4.5.1 and 4.5.2, respectively. The first deals with choosing a better codec for asynchronous use in order to provide an intelligible video file after compression, evaluated by means of objective metric and subjective MOS. The second, based on the results of the first, concentrates on that specific codec by modifying its compression algorithm and altering configuration parameters in order that the modified codec caters for sign language communication. The tradeoffs between quality and latency are shown in Figure 5-7.

5.5 Implementation Issues

In order to establish equity between all candidate codecs, the comparison process in the codec testing was performed under uniform conditions for all codecs. The machine used for each codec had the same CPU speed, memory, and web camera as that of all other codecs. In order to accommodate the candidate codecs, the system needed programmable interfaces to handle switching between codecs. Since the Windows Media SDK and the DirectX SDK were used, this system ensured that all codecs could be detected by the system after the relevant codecs were installed under Windows XP. In addition, the working platform for this system, as well as our experimental environment, was Microsoft Visual Studio in the Windows operating system. The system exchanged information with the fundamental functionalities provided by the Win32 Application Programming Interface (API) and MFC APIs. Therefore, the system was not a cross-platform system and only worked in MS Windows operating system.

Since the H.264 is not open source at the time of this writing, the experimentation had to make use of x264 instead. x264 appears to have almost the same functionality as H.264 except that it is a miniature version of H.264, and it is open source, as explained in Section 2.4.4. Fortunately, the decoder for x264 was very popular because most video players accept x264 on the fly.

For playback and replay processes, this system uses ffdshow as a general decoder of DivX, XviD and x264 videos. Therefore, this system did not take account of the decoding process used in the playback and the focus of this research was completely on the encoding process. To create a completely new codec was neither necessary nor meaningful, and would take a

Chapter 6 Experimentation and Results

The iterative method of experimentation detailed in Section 4.2 guided the development of a series of asynchronous sign language communication prototypes. We experimented with several video codecs in the lab and with Deaf users. The results of each experiment led to the next, as we aimed to improve video quality of the sign language videos while minimising latency. This chapter discusses the testing procedure according to the research methodology described in Section 3.3, and the data collection methods that followed. Then, we discuss the results obtained from both laboratory and user tests. Section 6.1 describes the iterative process. Section 6.2 discusses the preparations that led to the testing phase. Section 6.3 describes the details of the codec testing and the results obtained. Section 6.4 discusses the details of the optimization experimentation and the corresponding results.

6.1 Iterative Process for this Study

We applied an iterative process for this study that included both qualitative and quantitative research methods to accompany the standard waterfall software engineering process (see Figure 6-1).

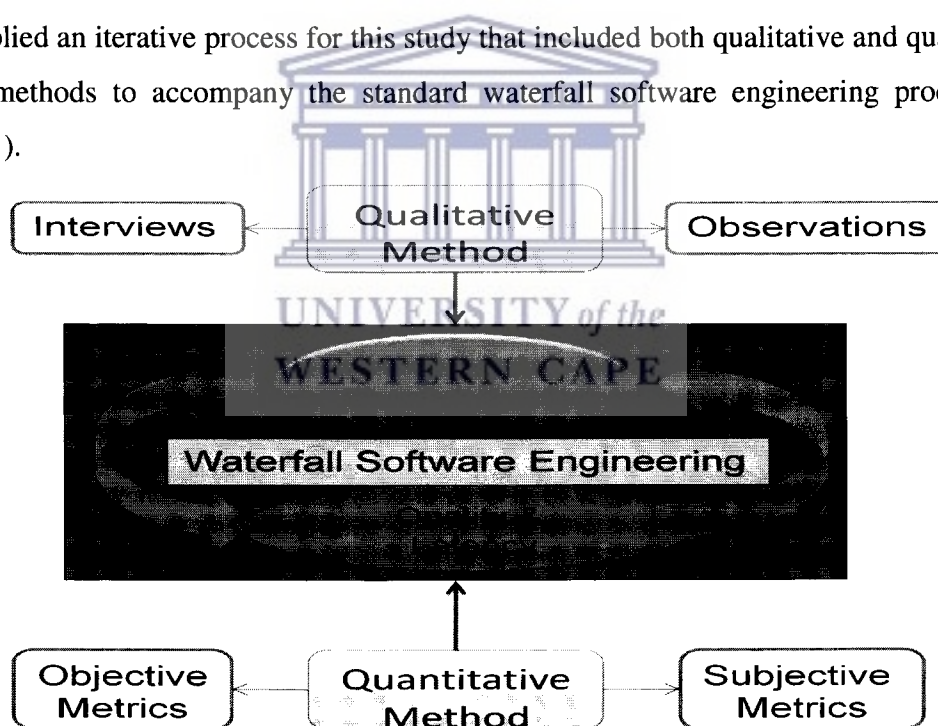


Figure 6-1: Iterative process

The overall iterative process consists of qualitative and quantitative methods that accompany a traditional waterfall software engineering method. The user requirements determine the project goals—video quality and latency to satisfy the Deaf user. Iterative the qualitative and quantitative methods inform the application of the waterfall process to make it more responsive to user needs.

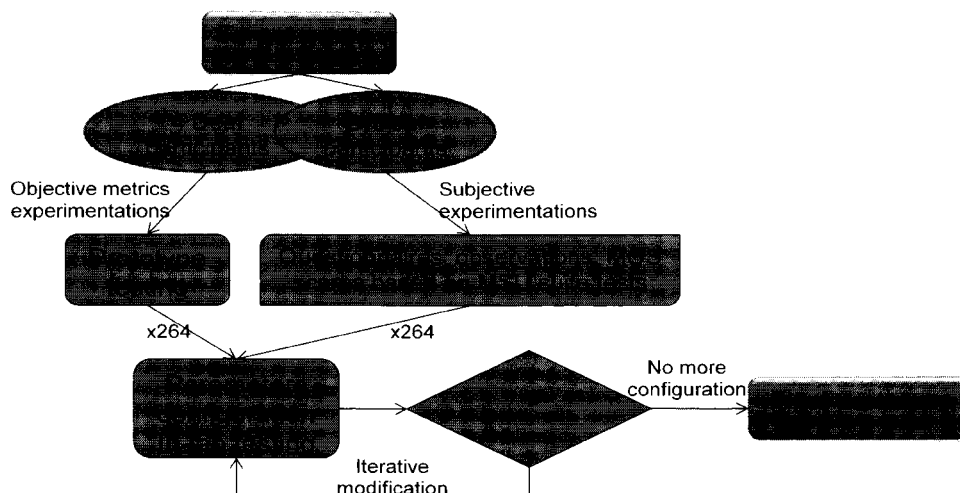


Figure 6-2: Testing process overview

The testing consisted of qualitative and quantitative methods on data collection and analysis. The codec selection testing was conducted in the field with Deaf participants and parameters configuration testing was conducted in the lab.

6.2.1 Target Group Selection

A significant consideration in designing a study is to select a subject sample that represents the population to whom the findings will eventually apply. The selected sample size is important with regard to applying results to the larger population. Selected participants need to reflect the relevant characteristics of the whole population. Therefore, the sample of participants for this project was chosen from the staff of DCCT, its social workers, English literacy class students, and some other Deaf people who attended the computer lab regularly. We also involved a SASL interpreter for translation.

Deaf participants were informed about our research and told what we were doing there before they became the participants of our project. An interpreter interpreted our ideas and intentions to them in sign language. Many of the participants had gaps in computer literacy. Therefore, we gave them some basic training sessions on computer literacy. This was often done informally alongside one of the English literacy courses at DCCT, described in Section 4.1. A computer lab in the Bastion has been available for DCCT members for several years. There were three Deaf lab assistants helping Deaf users on a daily basis. They would also solve some computer problems when our researchers were not present. As an increasing number of Deaf people used the computers, this project had a larger group of Deaf people from which to obtain volunteer participants. Volunteers signed the consent form (see Appendix B) and all information pertaining to details of the participants and records of data

tracing and backups of video files.

The time that users had to participate in the study was limited. To the research group, the once-a-week visit to the target community was limited as well. The greater the number of participants involved, the better the results would be. We allocated specific time slots for user testing on a given prototype, with each slot running for a specific period of time. For Deaf participants, perceptual tasks and performance testing appeared to be demanding. They would become fatigued or lose interest due to the workload. We therefore endeavoured to limit the test time to no more than 10 minutes for each participant. Necessary pauses and breaks in-between testing helped reduce the participants' fatigue levels. Suggestions and comments were welcomed during the pause and break time.

We tried to avoid unexpected problems or incorrect outcomes due to mistakes made by the participants because of misunderstanding the testing environment or procedure. A number of practice trials were conducted with the target sample to make sure that participants were familiar with the testing environment in the presence of an interpreter. Participants were asked to respond as they deemed appropriate with absolutely no right or wrong responses. Furthermore, every trial test recorded lists of responses or recommendations from the participants. These recommendations were helpful because the practice trials were not actual tests, and made for better testing.

6.3 Codec Testing Effort

The codec testing effort aimed at finding the most appropriate codec for asynchronous sign language communication. The goal was to provide high quality video evaluated with both objective metrics assessments in laboratory testing and the MOS values given by Deaf participants in the field. Laboratory testing employed standard objective metrics to evaluate the quality of videos. Perceptual testing was based mostly on the subjective opinions of Deaf participants because they were more knowledgeable in sign language than the developers were. They were asked to rate the intelligibility of sign language video. Most codecs were used in real-time video communication or for media storage. We adapted them for asynchronous Deaf video communication in a semi-synchronous environment. An iterative series of tests determined the codec best suited to asynchronous communication. The testing process integrated qualitative, quantitative and software engineering methods, described in

attitude towards video quality testing was encouraging. Participants thoroughly understood the tasks and readily recognized the small differences in video quality between different compression types. It became obvious that some of the participants had understood the differences between synchronous and asynchronous video communication when they tried our prototype. Those Deaf participants who had prior experience with synchronous video communication applications noticed the video quality was improved in our prototypes.

Participants were invited to join a group discussion after each session that involved the viewing of videos and test running the prototype. During these discussions, participants expressed their opinions and gave suggestions on all or some of the sign language video segments. In most cases, videos seemed to be quite similar. They could only tell the abrupt changes among the videos.

The development and implementation of the project was cyclical. Group discussions were held frequently at the beginning of each development cycle. Group discussions produced innovative ideas and insights that led to system improvements, such as enlarging the video display size, minimizing the number of mouse clicks, optimizing button sizes, adjusting the position, font colour and size of message boxes, as well as providing certain stimuli to notify users of what was happening. As expected, very few complaints about video quality were received since the use of the asynchronous approach had led to increased video quality. However, a greater number of complaints about the incurred latency were noted. The interactive discussion between Deaf people and us clearly improved our understanding of their requirements and simultaneously made them understand our goals.

6.3.3 Questionnaires

Participants were asked to fill in a simple written questionnaire after watching four sets of video clips. The questions in this questionnaire (see Appendix C) concerned the quality of the different video files compressed differently. A written questionnaire was given to a participant after the interpreter had explained it, directing the participant to answer each question. The first two questions queried the general attitude of the participant towards sign language video, and then the following two questions dealt with quality of the videos that were compressed differently. The name of codec name was not revealed to the participant, nor any explanation given in this regard in order to avoid confusion on the part of the participant. Furthermore, participants were not informed which video was the original and

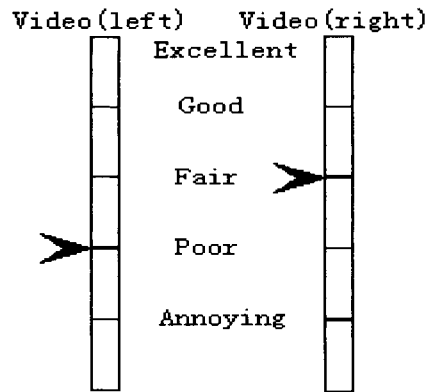


Figure 6-4: DSCQS measurement

DSCQS displays two individual scales after a viewer finishes viewing a pair of video clips. The viewer then gives scale marks for each video according to the levels shown: annoying, poor, fair, good, and excellent. Each level has a corresponding mark for MOS calculation.

SAMVIQ is a more complicated method that allows a viewer to play any video clip from the test set and to give it a mark. The video clips are played one after another, not in pairs, and each of them must be played at least once. After finishing watching all of the video clips, the viewer will see a list of marks for each video clip. This list of marks is then automatically written to a reference file with the compressor's name. In our case, there were four video files with the original video included. A Deaf viewer watched each of these files randomly, one after another, and gave a mark to each video. A report showed a list of marks for each video after Deaf viewers watched all of the videos.

6.3.5 MOS

An average subjective measure of a video sequence is named MOS, as detailed in Section 2.1.3. This mark was obtained by averaging the subjective scores of the participants. The formula is given as follows:

where v is the number of video segments ($v = 4$ in our test case) for which MOS is calculated. S is the sample number of the participants ($S = 21$ in our test case). $Mark_{i,v}$ is the mark given by the i^{th} Deaf person to the v^{th} video segment.

To estimate the probability of Deaf viewers being able to distinguish between two video clips that were compressed in different codecs, the z -test [69] was used to calculate for each pair of codecs. Let x and y be any two clips, then

video file; and S_c is the size of the compressed video file) and total objective delay time that comprised compression time and transmission time. With the help of a built-in automatic data collection tool, log files were created to record all of the relevant information as described in Section 6.4.2 for each communication session.

Figure 6-6 illustrates the compression ratio comparison between the three codecs. It can be seen from the pie chart that x264 gave a significantly larger compression ratio that indicates that the difference of S_o and S_c was very large. Consequently, S_c was small enough to bring down transmission time as well.

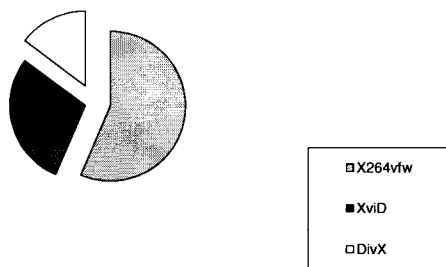


Figure 6-6: Compression ratio comparison

With the production of a smaller file size after compression, x264, implementing the H.264 standard (see Section 2.4.4), gave a better compression ratio than both DivX and XviD. x264 Video For Windows (x264VFW or x264vfw) is the API for Windows that is described in Section 6.4.1.

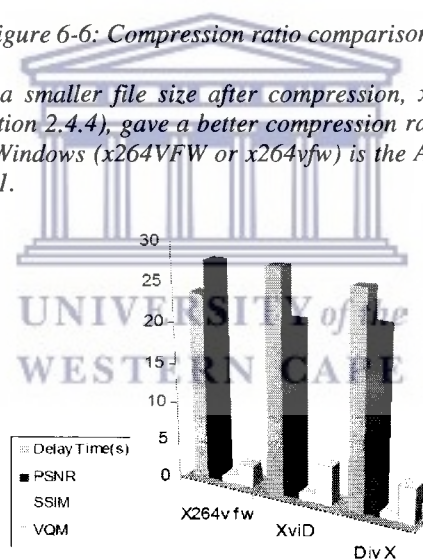


Figure 6-7: Objective evaluation metrics among DivX, XviD and x264vfw

From comparison of objective evaluation metrics, x264vfw seems to decrease the objective delay time as well as yield higher quality of video with respect to the values in PSNR and SSIM. Low delay time means low latency during communication. High values of PSNR, SSIM, and VQM mean good quality of video.

A larger compression ratio meant more time spent compressing frames, consuming more CPU, and consequently generating more delay during compression. In order to obtain balanced results, other metrics should also be taken into account. We therefore examined the

vector prediction in Section 6.4.9. Section 6.4.10 identifies some other issues that affected the results.

6.4.1 Integration of x264vfw

x264 is a codec based on the H.264 compression standard. There are two versions of the API, namely: x264CLI (x264 Command Line Interface) and x264vfw. x264CLI is command-line based and requires third party libraries when compiled. x264vfw, an API for Windows, depends only on the external library libx264.lib that is generated by compiling x264 source code. We used the x264vfw API as a codec plug-in to accomplish compression. The relationship between x264, x264vfw and our system is illustrated in Figure 6-8.

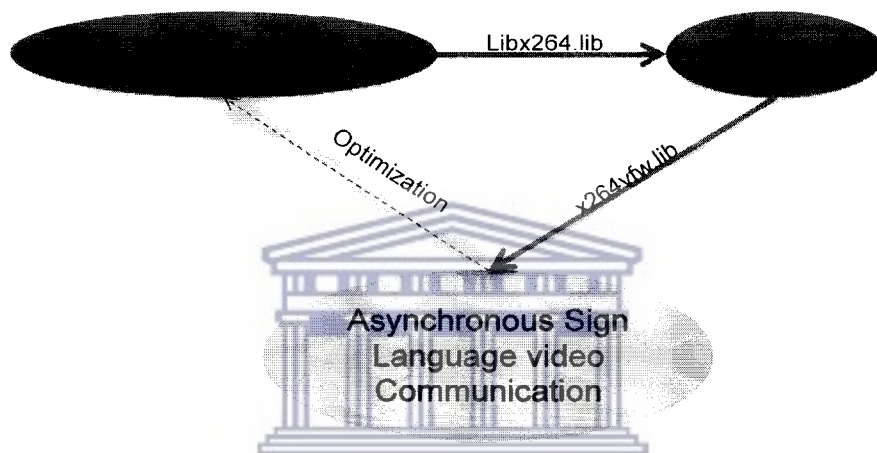


Figure 6-8: Dependencies for x264 adaptation

Our system depends on x264vfw.dll that is created by compiling the x264vfw API. x264vfw relies on libx264.lib that is generated by compiling the x264 source code. The optimization in the adaptation process takes place in the x264 source code.

The optimization took place within the x264 source code. Recompile resulted in a corresponding modification on the libx264.lib library each time, and x264vfw changed accordingly. This was the initial step of integration of x264 into our system and was then followed by the configuration changes in pursuit of the reduction of delay without degradation in video quality. Our sign language asynchronous video communication tool evaluated the adaptation each time in the last step of each quality-latency monitoring cycle described in Section 6.4.3.

6.4.2 Event and Time Tracing

Throughout the testing procedure, an event and time tracing function was added to our

These quality-latency monitoring tests were conducted by modifying x264 parameters whilst monitoring changes in latency and quality. Latency monitoring was obtained from event and time consumption tracing (see Section 6.4.2).

Quality monitoring evaluation utilized the MSU video quality measurements metrics: PSNR, SSIM index and the VQM values (see Section 6.3.6). The system collected the data on CR, compression time (CT), transmission time (TT), and total (objective) delay time (DT) (see Section 6.3.6). The comparison tests looked for changes in percentages of all of these metrics, and all configuration modifications pursued only one goal: improve quality and reduce latency in each cyclical test. Furthermore, all experimental modifications in the adaptation processes were done in the lab so that the output would not add extra factors that could affect latency subjectively.

Parameters	Characteristics
Integer Motion Estimation	dia: diamond search with radius 1 hex: hexagonal search with radius 2 umh: uneven multi-hexagon search Chroma: enabled or disabled
Reference Frame	up to 16 reference frames for motion compensation
Noise Reduction	Levels from 0 to 5 or more
Entropy coding	CAVLC: luminance and chrominance residual encoding CABAC: dynamically chooses probability module for encoding, depending on current content and previous encoded content
B-frame	multiple B-frames with adaptive or non-adaptive decisions
direct Motion Vector (MV) prediction modes	spatial, temporal and auto
Chroma	Enabled or disabled
In-the-loop deblocking filtering	Enabled or disabled

Table 6-1: x264 parameters and their characteristics

This table lists the most important terms and specifications with which x264 is configured. Certain combinations of parameters were able to perform well in real time, but were not necessarily suitable for asynchronous video communication.

Figure 6-9 depicts the cyclical process of parameter configuration. Whenever a specific parameter was modified for a particular purpose, e.g. parameter ME for motion estimation methods, the system rebuilt the source code of x264 and then x264vfw to keep the codec

MV. In the ME method, finding the MV is called the search method and this method affects the encoding efficiency significantly. Examples of integer pixel based motion estimation search methods are dia, hex, umh, esa, and tesa. The dia method is a diamond search with radius 1 and is widely considered as a fast method [81]. The hex method is hexagonal search with radius 2. The umh method is uneven multi-hexagon search. The esa method is exhaustive search and the tesa method is Hadamard exhaustive search [81]. The last two methods are time consuming, and, therefore, only the first three methods were used in comparisons. Figure 6-10 shows the comparison of the three different search methods (dia, umh and hex).

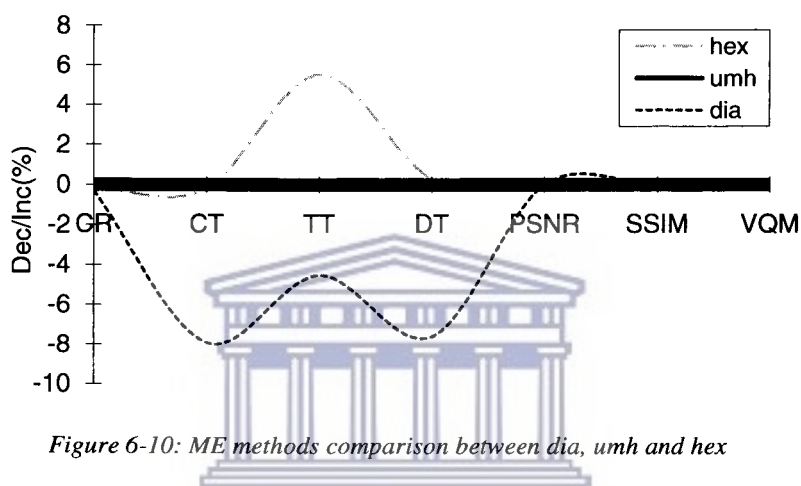


Figure 6-10: ME methods comparison between dia, umh and hex

The x-axis is the comparison contents while the y-axis is the percentage of increment (greater than 0) or decrement (smaller than 0). The rest of comparisons were treated with all of these measures. The relationships between the axes depict the changes of latency and quality. umh, the default parameter used in x264, is a reference in our case. It is shown on the x-axis line. The dia method achieved reductions in latency and even a bit of increase in PSNR over the other two. The hex method increased transmission delay time heavily, obtaining the same quality as that in umh method.

The dia search method made x264 more efficient and saved time during compression with a moderate speed increase (7.6086%), and resulted in a reduction of delay time. The quality of compressed video also slightly improved, with a bit of an increase in PSNR over the other two and with VQM increasing by 0.12247%.

6.4.5 Reference Frames Testing

The reference frame test sought an optimal number of reference frames. Section 2.4.3 noted that the number of reference frames was typically one, or in the case of conventional B-frame, two. x264 allows up to 16 reference frames in the compression process. More

process. In this test, however, it did not make a remarkable impression on quality and changes were hardly noticeable. Thus, this test concentrated on the delay time and the changes of objective metrics on quality. It can be seen in Figure 6-12 that when NR is 3, the delay time was slightly reduced to a total of 0.1001% and the objective metrics for video quality only improved slightly with VQM increased by 0.169296%.

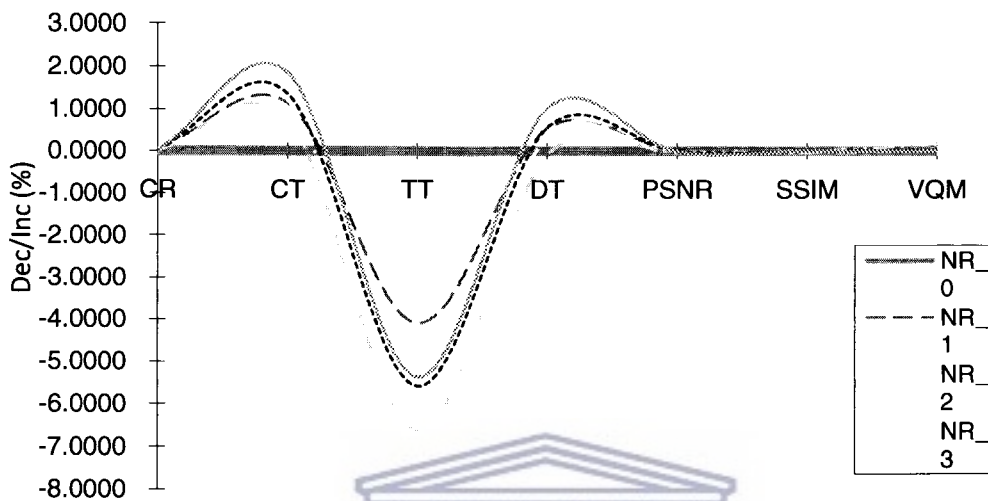


Figure 6-12: Noise reduction levels comparison

In the graph above, NR_0 is taken as a reference shown on the x-axis line. The noise reduction process aims at improving the video quality from the perceptual view aspect, and therefore, it introduces a complex algorithm to remove noise that appears in unnoticeable MBs. The higher the level of noise reduction, the more time consuming the compression process will be. NR level of three was the best one, with a huge reduction in transmission time and a slight improvement in quality.

6.4.7 Entropy Coding Testing

CAVLC replaced the previous Universal Variable Length Coding in earlier versions of H.264 that emphasized residue entropy coding. The concept of CABAC is to represent an input character stream by using a number between 0 and 1. Additionally, CABAC allocates a bit to the entire input stream instead of each character of that input stream. Its complexity appears in the process of possibility estimation and updating, and it dynamically matches one of the possibility models according to the current encoding context and even the previous encoded context.

In the entropy testing, these two methods were employed by the system respectively to compare the outcome. It was interesting to note that using CAVLC instead of CABAC not

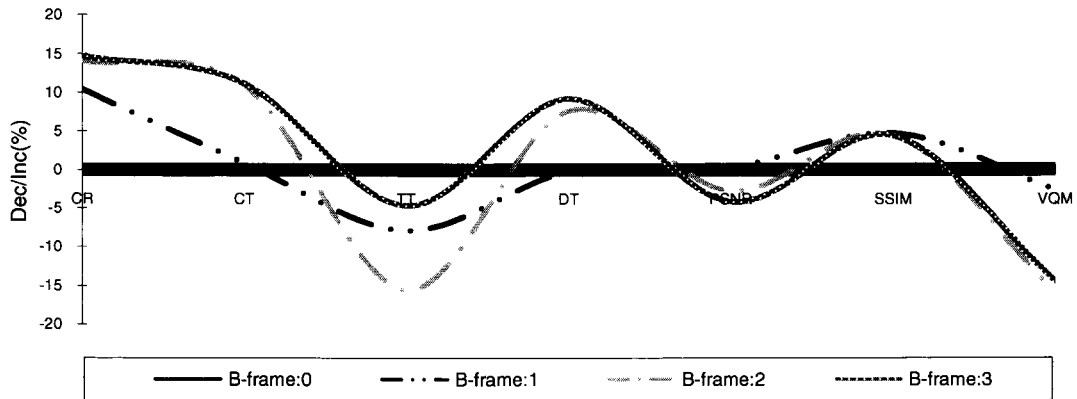


Figure 6-14: Comparison on B-frame numbers: none, 1, 2 and 3.

In the graph above, no B-frame is a reference shown on the x-axis line. The compression time increased as more B-frames were involved in the encoding process. Although the transmission time seemed to be reduced, the total delay time increased. The tradeoffs of quality and latency were balanced when only one B-frame was used without great impact on quality of sign language video.

6.4.9 MV Prediction Modes Testing

The MV compresses video by storing changes to an image from one frame to the next. The process is a bi-dimensional pointer that communicates to the decoder how much left or right and up or down the predictive macroblock is located from the position of the macroblock in the reference frame or field. If any error occurs in MV prediction, the decoder will not decode the corresponding frames that might affect the following decoding process. Thus, the video cannot be played due to a decoding corruption. Whether or not the MV search is determined to be efficient totally depends on Mean Absolute Error (MAE) [78]:

$$MAE_{(i,j)} = \frac{1}{N^2} \sum_{k=0}^{N-1} \sum_{l=0}^{N-1} |C(x+k, y+l) - R(x+i+k, y+j+l)|,$$

where $C(x+k, y+l)$ represents the pixels in the MB with upper left corner (x, y) in the target frame; $R(x+i+k, y+j+l)$ represents the pixels in the MB with upper left corner $(x+i, y+j)$ in the reference frame; and N^2 is the area of the current MB. Thus, the aim of the MV search is to find a vector $\overline{V_{(u,v)}}$ such that $MAE_{(i,j)}$ is the minimal value. This test involved three modes of MV, namely, spatial, temporal and auto. Spatial MV calculates the motion vector within a single frame while temporal MV calculates the motion vector between frames. The auto mode for MV calculation could be either of the two or both in a particular context, such as a quick hand gesture. In this test, the default MV was calculated from relative spatial offsets.

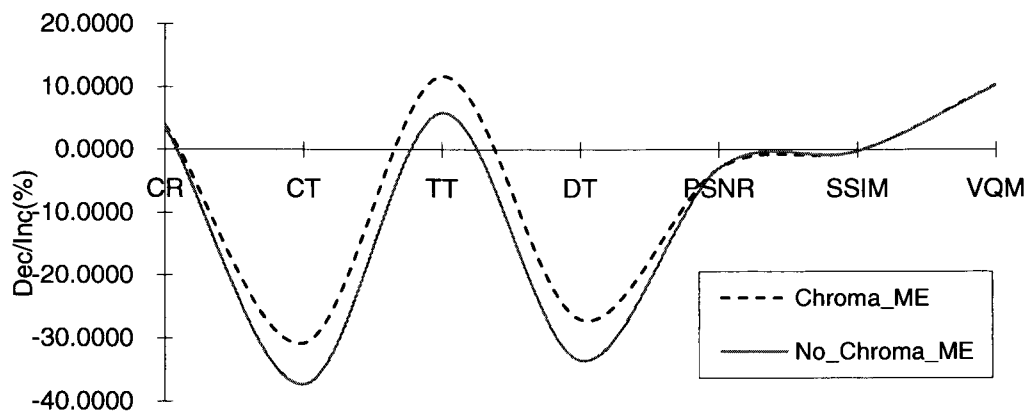


Figure 6-16: Chrominance in ME

The characteristics of chrominance in the ME test determined the complexity of the compression algorithm and certainly increased the compression time. However, disabling the chrominance did not cause losses in the colour of video at all. It did make the colour contrast lower than average so that the bit rate reduced and the compression algorithm reduced in complexity causing lower time consumption.

An in-the-loop deblocking filter prevented the blocking of artefacts incurred from spatial motion vector prediction that are common to other DCT-based image compression techniques, and makes the frames smooth by reducing the pixels with noises and by blurring the interlaced edges where MBs meet. However, in our tests, we found that the compression speed penalty had a significant impact on latency. In addition, it required the same deblocking techniques on the decoder side. Therefore, disabling the in-the-loop deblocking filter suited our asynchronous video communication solution.

6.4.11 Summary

Data gathered from the experiments was either subjective or objective. Subjective data was collected by means of interviews, focus groups questionnaires, and user observations in conjunction with qualitative MSU methods, and combined into MOS scores. Objective data was collected with the help of MSU objective metric tools together with built-in time and event tracing. The entire experimental process consisted of two main phases: the first to find the best codec and the second to optimize it through parameter configuration. In the codec choice phase, we found that x264 was the best of the three state of the art codecs, DivX, XviD and x26. In the optimization phase, we found the best combination of configuring parameters of x264 by balancing tradeoffs between quality and latency in our case. These values are shown in Table 6-2.

Chapter 7 Conclusion and Future Work

This chapter summarizes the thesis by recapping the research question, reviewing our approach to solve it, and discussing the results obtained from the experiments to choose and adapt a synchronous codec into an asynchronous video communication tool for Deaf people. This chapter also discusses future work for the inquiry.

7.1 Conclusion

This research was motivated by previous work done with a group of Deaf people who represented disadvantaged South African Deaf people. They were interested in video communication by means of sign language. However, affordable and accessible synchronous video communication options did not provide intelligible sign language video for them. Asynchronous video communication, on the other hand, was cumbersome and entailed long latency. Deaf video telephony differed from both synchronous and asynchronous video telephony for hearing users because sign language video required high quality to recognize subtle movements of the hands and facial expressions. The research question was therefore: *How can we design a system with asynchronous, as opposed to synchronous video, by adapting synchronous codecs to achieve semi-synchronous communication for Deaf users?* Section 3.2 mentioned that this research question had two perspectives. The subjective perspective involved the development and deployment of an asynchronous video communication tool, and an objective perspective explicitly addressed quantitative considerations to establish that such a system brought semi-synchronous service with high sign language video quality and minimal latency. This research effort produced an asynchronous approach, with a current synchronous codec that was adapted and optimized for our system. We built a semi-synchronous video communication application to provide high video quality and minimal latency services. The rest of this section summarises the entire research process. Figure 7-1 shows the overview of this project. Section 7.1.1 summarises social and technical perspectives. Section 7.1.2 discusses subjective and objective perspectives. Section 7.1.3 concludes with the results we achieved and lessons we learnt in this regard.

7.1.2 Subjective and Objective Perspectives

From the experiences related to us by the Deaf participants, we found that they wanted to use sign language during communication. Thus, the quality of sign language video was a significant concern in our research. The outcome from initial subjective data collection pointed to an asynchronous approach in order to provide sign language video quality. The subjective perspectives expressed in the MOS results helped us determine which codec was better for asynchronous video communication with respect to quality and latency considerations. In the case where the results obtained from subjective experimentation could not clearly distinguish between video files compressed by different codecs, objective video quality measurement metrics helped. The subjective evaluation reported in Section 6.3.5 for three codecs: DivX, XviD and x264, correlated with the objective evaluation described in Section 6.3.6.

Objective perspectives gave solid results on quality evaluation of sign language video by measuring metrics that predicted the intelligibility of video. Those metrics included PSNR, SSIM and VQM. They were based on both the spatial and temporal structure of video. A traditional quality metric, PSNR, described the basic signal to noise ratio to highlight differences between an original and an encoded video, implying how close the encoded video was to the original one. Besides PSNR, we also used the other two objective metrics, SSIM and VQM, to help judge the quality of sign language videos. Additionally, the objective perspective measured latency incurred from the S&F strategy. Objective latencies were measured automatically by our software and were recorded into files, serving as optimization data references.

Subjective and objective perspectives were mutually interactive and necessarily complementary. The outcomes obtained from both subjective and objective quality evaluations were compared to triangulate results. The subjective perspective included opinions of Deaf people and their attitudes towards the quality of compressed videos. This contributed to an improvement of our system according to the participants' feedback. When watching the video clips that were compressed differently, for instance, they made their own judgments on the quality of those sign language videos, and distinguished the slight changes in each sign language video, although they were not aware of the codec used. Objective latency was not measured through subjective methods, but through objective methods instead.

delay is different because it depends completely on the user. One way to reduce it is to optimize the notification service to urge Deaf users to save time during the capture and playback of videos. Notification optimization could follow each step made by the users and indicate the status of success or failure of operations in order to save time. We could also add Bluetooth notification via a user's cell phone to vibrate in order to notify user of an event when they are away from the PC. In addition, ICT literacy training for the Deaf users are significant important so as to enable them to be equipped with new technology. Therefore, the Deaf users could make use of our new devices and systems in our further research.

Codec optimization is under development. A new standard draft H.265 has been developed by the ITU- Study Group 16 (ITU-SG16) and is expected to be finalized in 2009-2010. The goals of H.265 focus on simplicity and "back to basics" approach; coding efficiency (ITU-SG16 says that the efficiency in H265 is twice than that in H.264); computational efficiency; loss/error robustness; and network friendliness. In principle, they consider encoder as well as decoder computational efficiency to be worthy of consideration [62].

Transmission optimization could be done in two ways. One is to reduce the size of the video file and the other involves network considerations. The file size is controlled by the capture process. The capture process could divide captured videos into segments and encode them separately, and then transmit them individually. Meantime, a remote side could reconstruct them and hopefully minimise jitter.

7.2.2 Future User Interfaces

The user interface of the asynchronous video service could be made more advanced and intelligent. For example, Deaf users could sign to control the operations instead of using the mouse and keyboard. Both types of interaction could be captured by a web camera. Usability testing could be conducted in the future to evaluate the semi-synchronous service for Deaf people with its user interface.

7.2.3 Cross-platform Support

This system was built in Microsoft Visual Studio 2005 and was only deployed with the Windows operating system. A cross-platform version of this system could be achieved with open source libraries that are supported by various types of operating systems, such as Qt and wxWidgets.

More and more Deaf users are cell phone subscribers and show great interest in cell phone

References

- [1] Ahmed, N.; Natarajan, T. and Rao, K.R. (1974). Discrete Cosine Transform. *IEEE Trans. Computers*, C-23, pp. 90-93.
- [2] Alan, M.D. (September, 1992). Operational Prototyping: A new Development Approach. *IEEE Software*, p71.
- [3] Bangham, J. (2000). Speech and Language processing for disabled and elderly people. *IEEE Seminar*.
- [4] Baset, S.A. and Schulzrinne, H. (April, 2006). An Analysis of the Skype Peer-to-peer Internet Telephony Protocol. *IEEE International Conference on Computer Communications*, Barcelona, Spain. pp. 1-11.
- [5] Bauer B. and Kraiss K. F. (2001). Toward a 3rd Generation Mobile Telecommunication for Deaf people. *10th Aachen Symposium on Signal Theory*, Aachen, Germany. pp. 101-106.
- [6] Beyer, H. and Holtzblatt, K. (1995). Apprenticing with the customer. *Communications of the ACM*, 38 (5), pp. 45-53.
- [7] Bi, Houjie. (2005). The new generation video compression encoding standard-H.264/AVC. ISBN: 7-115-13064-7, Beijing, China: China Post & Telecom Press.
- [8] Bulow, M.; Burger, J.; Hagge, L. and Panto, S. (March 24-28, 2003). UML and Exploratory Prototyping for Developing Interactive Applications. *The International Conference on Computing in High Energy Physics (CHEP) 2003*, San Diego.
- [9] Calderbank, R. C.; Daubechies, I.; Sweldens, W. and Yeo, B. L. (1998). Wavelet Transforms that Map Integers to Integers. *Applied and Computational Harmonic Analysis (ACHA)*, 5 (3), pp. 332-369.
- [10] Cavender, A.; Ladner, R. and Riskin, E. (October 23-25, 2006). MobileASL: Intelligibility of Sign Language Video as Constrained by Mobile Phone Technology. *Proceeding of ASSETS 2006: The Sixth International ACM SIGACCESS conference on Computers and Accessibility*, Portland, OR. pp. 71-78.
- [11] Chen, J.; Kao, C. and Lin, Y. (2006). Introduction to H.264 Advanced Video Coding. *Proceeding of the 2006 conference on Asia South Pacific design automation*. ACM Press, New York, NY, USA. pp. 736-741.
- [12] Cherniavsky, A.; Cavender, A. C.; Ladner, R. E. and Riskin, E. A. (October 15-17, 2007). Variable Frame Rate for Low Power Mobile Sign Language Communication. *Proceedings of the 9th international ACM SIGACCESS conference on Computers and Accessibility (ASSETS'07)*, Tempe, Arizona, USA. pp. 163-170.

- [27] ITU-T P.910. (1999). Subjective video quality assessment methods for multimedia applications - Series P: Telephone Transmission Quality, Telephone Installations, Local Line, Networks - Audiovisual quality in multimedia service.
- [28] ITU-T Recommendation H.261. Video Codec for Audiovisual Services at px64 kbit/s.
- [29] ITU-T Recommendation H.263. Video coding for low bit rate communicatio.
- [30] ITU-T SG16. (1998). Draft Application profile: Sign language and lip reading real time conversation usage of low bit rate video communication. Geneva.
- [31] Ivanov, Y. V. and Bleakley, C. J. (September 23-28, 2007). Dynamic Complexity Scaling for Real-Time H.264/AVC Video Encoding. *MM'07*, pp. 962-970.
- [32] James, C. L. and Reischel, K. M. (2001). Text input for mobile devices: comparing model prediction to actual performance. *CHI'01: Proceedings of the SIGCHI conference on Human factors in computing systems*, pp. 365-371.
- [33] Johanson, M. (May 1-5, 2001). An RTP to HTTP Video Gateway. *Proceeding of the 10th international conference on World Wide Web*, Hong Kong, Hong Kong. pp. 499-503.
- [34] Joint Video Team (JVT) of ISO/IEC MPEG and ITU-T VCEG. (2003). Draft ITU-T Recommendation and Final Draft International Standard of Joint Video Specification (ITU-T Rec. H.264/ISO/IEC 14496-10 AVC), Pattaya, Thailand.
- [35] Kitzinger, J. (July 29, 1995). Qualitative Research: Introducing focus groups. *BMJ*, pp. 299-302.
- [36] Kolarov, K. and Lynch, W. (1999). Very low cost video wavelet codec. *SPIE Conference on Applications of Digital Image Processing*, Vol.3808.
- [37] Lewis, A. S. and Knowles, G. (April, 1992). Image Compression Using the 2-D Wavelet Transform. *IEEE Trans. IP*, 1 (2), pp. 244-250.
- [38] Li, J.; Chen, G.; Xu, J.; Wang, Y.; Zhou, H.; Yu, K.; Ng, K. and Shum, H. G. (September 30 - October 5, 2001). Bi-level Video: Video Communication at Very Low Bit Rates. *Proceeding of MM'01*, Ottawa, Canada: ACM.
- [39] Lian, C.; Huang, Y.; Fang, H.; Chang, Y. and Chen, L. (2005). JPEG, MPEG-4, and H.264 Codec IP Development. *Proceeding of the Design, Automation and Test in Europe Conference and Exhibition (DATE'05)*, IEEE.
- [40] Liu, L. and Fan, G. (February, 2003). A New JPEG2000 region-of-interest Image Coding: Most significant Bitplanes Shift. *Signal Processing Letters, IEEE*, 10 (2), pp. 35-38.
- [41] Long, B. and Baecher, R. A Taxonomy of Internet Communication Tools - Dynamic Graphics Project. University of Toronto, Department of Computer Science and Knowledge Media Design Institute, Toronto, Canada.
- [42] Lu. G. (1996). *Communication and Computing for Distributed Multimedia Systems*. Norwood, MA, USA: Artech House.

- [57] Schulzrinne, H.; Scsner, S.; Frederick, R. and Jacobson, V. (July, 2003). RTP: A transport protocol for real-time applications. STD 64, REC 3550.
- [58] Seaman, C. B. (July/August, 1999). Qualitative Methods in Empirical Studies of Software Engineering. *IEEE Trans. on Software Engineering*, 25 (4), pp. 557-572.
- [59] Shen, J.; Yan, R.; Pei, S. and Song, S. (November 2-8, 2003). Interactive Multimedia Messaging Service Platform. *Proceedings of the 11th ACM International Conference on Multimedia (MM'03)*, Berkeley, CA, USA. pp. 464-465.
- [60] Shintani, M.; Ohara, T.; Ichihara, H and Inove, T. (2005) A Huffman-based coding with efficient test application. *Proceedings of the 2005 conference on Asia Pacific design automation*. Shanghai, China.
- [61] Smith, S. W. (2001). *The Scientist and Engineer's Guide to Digital Signal Processing*. California Technical Publishing.
- [62] Sullivan, G. J. (April, 2005). Meeting report for 27th Video Coding Experts Group(VCEG) Meeting. ITU-telecommunications Standardization Sector, Study Group 16.
- [63] Sullivan, G. J.; Topwala, P. and Luthra, A. (August, 2004). The H.264/AVC advanced video coding standard: Overview and introduction to the fidelity arrange extensions. *SPIE conference of Digital Image Processing*.
- [64] Sun, T. And Tucker, W. D. (September, 2004). A SoftBridge with Carrier Grade Reliability Using JAIN SLEE. *Proceeding of Southern Africa Telecommunications Networks & Applications Conference (SATNAC 2004)*, Stellenbosch, South Africa. pp. II-251-252.
- [65] Tang, Q. and Jin, J. S. (2003). Compressed video transmission over digital networks: analysis and design. *ACM international Conference Proceeding Series, the 2002 Pan-Sydney workshop on Visualisation*, Sydney, Australia. pp. 97-100.
- [66] Tashiro, Y.; Yashima, Y. and Fuju, H. (January, 2006). NTT's Technologies for Next-Generation Video Services. *ACM Computers in Entertainment*, 4 (1), p. Article 5A.
- [67] Tucker, W. D.; Glaser, M. and Lewis, J. (September 2003). SoftBridge in Action: The First Deaf Telephony Pilot. *Proceeding of Souh African Telecommunications and Networking Application Conference(SATNAC 2003)*, George, South Africa. pp. II-293-294.
- [68] van der Merwe, K. (2005). Want access to government services? With TISSA you can! *SAPS, 2006*.
- [69] Vatolin, D.; Petrov, O.; Parshin, A. and Titarenko, A. (December, 2005). MPEG-4 AVC/H264 video codec comparison. *Moscow, Russia: Graphics and Media Lab of Computer Science Department, Moscow State university*.

Appendix A: Information Sheet

Who are we?

We are Computer Science researchers from the University of the Western Cape and the University of Cape Town. The team members are Zhenyu Ma, Wilson Wu, and Bill Tucker. Meryl Glaser is our advisor.

What do we want to do?

We want to improve communications between Deaf and hearing people. The aim is to provide a Deaf relay service to the telephone system using computers on the Internet. Our system, Video Relay Service, allows Deaf and hearing people to communicate with help of relay operator. The Deaf user sits at a computer (PC) at Deaf Community of Cape Town's (DCCT) premises. We will provide the PCs and web cameras. The Deaf user communicates with signing video. A Deaf user's records what he or she signed via web cameras and stores in local machine, then the recorded file is transmitted to a qualified sign language interpreter and the interpreter speaks to a hearing from an earphone. The hearing user response with voice to that interpreter and the interpreter record her or his signing and transmits to that Deaf user. Our research is primarily the development of the communication software. We will work with Deaf and hearing people to design and change the software over time. Our research runs in cycles where we introduce improvements, provide training and talk to the users to make improvements for the next cycle.

Over the past several years, we have attempted to build this service several times. We have learned that there are two main issues concerning the use of such a system: computer literacy of Deaf people and proper operation of this service. We want to deal with these issues, because both issues introduce a lot of delay into the conversation. For example, the hearing user will have to wait for the Deaf person to record a sign video and for the transmission to a relay operator and translation from sign language to voice by the relay operator as well. Likewise, the Deaf user will have to wait while the operator translates the speech into sign language, records a video, and as well transmits the video file. We want to learn how to best deal with these kinds of delay in the conversation.

Why are we doing this?

We have already developed some software in the laboratory, but the reason we are doing this project is that we are interested in how we can use technology to help communication for the Deaf community. We are doing this as part of our research studies. We want to know things like, the system is useful, how we can make it better and how many times you use it. We will write about our experience of doing this work to help others who want to do similar work in South Africa, and even the rest of the world. We also want to show the Department of Communications the kinds of things that can be used to improve communications for the Deaf.

Who funds this project and how will it continue when we are done?

This research is funded by the Centres of Excellence at both UWC and UCT until the end of 2008. Please note that this is not a donation, nor is a commercial product. We are interested in learning how to use technology to help the Deaf communicate with the hearing over long distances. If Asynchronous Video Relay Service proves useful, we would like to convince the Department of Communications to support the project. We hope that the community will work with us to do this!

If you agree to join this project, I will ask you to sign a consent form, but you can leave the project at any time without any penalty to you at all. Participation is your free choice. You will be asked to use the system and allow yourself to be interviewed about the system at regular intervals when we visit your site.

Appendix C: Questionnaire

Part I: General Questions:

1. What is the most difficult issue you have encountered during the real time video communication? Please choose some appropriate answer(s) according to your situation:
 - A. Video quality issue
 - B. Complexity of running communication software
 - C. Internet access issue
 - D. Frequent connection drops during communication
2. As far as non-real time video communication methods, such as MMS, Video Messaging, etc., are concerned, what do you think of this approach?
 - A. Improved video quality
 - B. More meaningful of Video contents
 - C. More latency during communication
 - D. Difficulty in performing video communication

Part II: Video Codecs Comparisons:

1. What do you think the BLURRING of the videos you have just viewed?

	V1	V2	V3	V4
Excellent	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Good	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Fair	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Bad	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Annoying	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

2. What do you think the UNDERSTANDING of the videos you have just viewed?

	V1	V2	V3	V4
Excellent	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Good	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Fair	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Bad	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Annoying	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Note: V1-V4 is compressed by H.264, by XviD, by DivX and by MPEG4 respectively.