Browser-based video communication for Deaf people

Yuanyuan Wang, William D. Tucker
Department of Computer Science, University of the Western Cape, Bellville, South Africa
Telephone: +(27) 21 959-2461, Fax: +(27) 21 959-3006, Email: {**2918818**, btucker}@uwc.ac.za

Abstract— This paper describes work in progress to to provide browser-based sign language communication for Deaf users. The paper introduces the communication requirements for a disadvantaged Deaf community in Cape Town, describes related work of generic open source browser-based video and video systems targeted specifically to Deaf users. We posit an experimental design to test if Adobe Flash can help provide an acceptable video system for Deaf users. We wish to learn if the transport and video quality of browser-based Adobe Flash video can provide an effective platform for sign language communication.

Index Terms— Deaf communication, browser-based video, semi-synchronous, Adobe Flash

I. INTRODUCTION

THIS paper describes how Adobe Flash can provide a viable browser-based platform on which to build a sign language communication system for Deaf users. We intend to design and test a browser-based prototype that can provide semi-synchronous video for Deaf users. Common video messaging and conferencing systems are not designed to support sign language. Problems include the frame of view, e.g. focus on the head as opposed to the entire torso required to see both the face and dynamic hand gestures. Another problem with web-based video results from low frame rate and poor resolution in systems that are actually meant to support voice communication and not the medium of sign language. This project follows on related work that adapted the x264 codec for asynchronous video in a standalone application [1]. That PC-based prototype provided users acceptable video quality for sign language communication with minimised delay. Adobe Flash has both synchronous and asynchronous capabilities. We want to design and test a browser-based prototype with Adobe Flash with a group of disadvantaged Deaf people in Cape Town, and attempt to port the prototype to a cell phone.

According to census statistics, there are roughly 4 million people with hearing impairment in South Africa [2]. Of these, 10% are profoundly Deaf, and they use South African Sign Language (SASL) as their primary means of communication. Deaf with a capital 'D' thus refers to this group of people with a unique language and culture. In South Africa, Deaf people are under-educated and under-employed due to a combination of physiological and socio-economic factors [3]. Without text and computer literacy, and unable to speak or hear, Deaf people find text communication difficult. That said, Deaf users frequently use SMS (Short Message System) with both Deaf and hearing users. However, their awareness of poor grammar and spelling in English embarrasses them and inhibits them from using text to communicate with hearing people they know are more literate than they are. Thus, Deaf people prefer to communicate proficiently in sign language.

We believe a web-based system based on Adobe Flash could address their needs. An NGO (non-governmental organisation) called Deaf Community of Cape Town (DCCT) maintains a small PC lab at the Bastion of the Deaf in Newlands, Cape Town. We want to test out a browser-based prototype with Deaf users on PCs before attempting to port the system to cell phones. The roadmap is to try out the browser-based prototype for Deaf-to-Deaf chat and a video relay service (VRS) that employs a human interpreter to translate between SASL and English to effect communication between sign language and hearing users.

The next section describes work related to currently available open source generic video systems, and then dedicated Deaf video systems. We can learn from both types of systems to learn how build a browser-based system tailored for Deaf users. Section III describes the experimental design. Section IV concludes the paper by describing the way forward.

II. RELATED WORK

Web-based video communication systems are commonly available. We are interested in open source solutions because we can examine the architecture. Tokbox supports video with Adobe Flash. Tokbox is web-based so users need not install or download speciality plug-ins in the client. Tokbox is like a web version of Skype without PSTN (public switched telephone network) breakout. Dimdim is another system based on Adobe Flash. Dimdim is open source and supports multiuser meetings. Dimdim supports video and is web-based. However, users need to install custom plug-ins to use advanced features such as desktop sharing. Users can install a Dimdim server by themselves. Vmukti is another open source system for conferencing. Vmukti is web-based and built on the .NET framework. This means a user needs to install .NET in order to use Vmukti. A limitation is that it only runs on Windows operating systems. The frame of view for all of these systems leans toward the 'floating' head to support (and not replace) audio communication, and the frame rate and resolution appear suboptimal in order to prioritise voice traffic.

Alternatively, MobileASL provides a video system specifically for Deaf users [4]. For example, MobileASL employs a region of interest (RoI) method to selectively focus on the face and moving hands within the video. MobileASL has the additional advantage of working on a cell phone to support mobile communication of American Sign Language (ASL). TESSA was a less sophisticated video tool that translated speech to text automatically, and then translated that text to sign language automatically but only for a post office scenario [5]. As a final example, Sorenson is a company that provides custom video phone, Internet and TV-based systems that support both Deaf-to-Deaf communication and VRS. VRS systems are often subsidised by governments and operators in developed regions, but none exist in South Africa.

III. Experimental Design

Adobe Flash is a common, yet proprietary, way for users to exchange video and audio data over the Internet. Many browsers support Adobe Flash with plug-ins. There are two ways to transmit media data between a server and client using Adobe Flash. Firstly, video media can be transferred asynchronously as a download with HyperText Transfer Protocol (HTTP). This method practically guarantees a high standard of video quality, dependent on the host machine's processing capability, at the expense of incurred delay. Secondly, video can be streamed with the Real Time Media Protocol (RTMP) where bandwidth availability determines the quality and speed of the video playback. Streaming can be real-time, but can also continuously start and stop thus providing a mixture of real-time and asynchronous transfer, making it semi-synchronous at best. Video quality can also be artificially degraded to improve streaming speeds. We want to test both forms of delivery with actual Deaf users to see how the delivery methods address the needs of sign language communication.

The overall system design is shown in Fig. 1. There are two primary components within the system: server and client. We will use the Red5 Flash server to get and broadcast the media files. A MySQL database will store user information. The Graphic User Interface (GUI) and some view controllers are included in the client within a browser. We will use the OpenLazlo open source software to show the pages, and record and playback the media files.

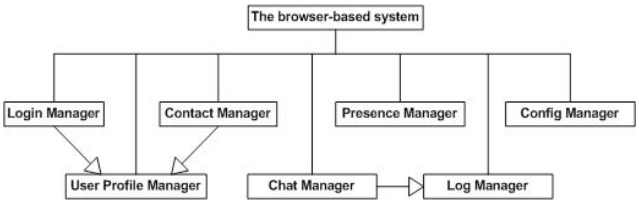

Fig. 1 The main components of the proposed prototype are shown here. The log managers are invisible to the user and run in the background automatically. Users may directly interact with the other components.

When users open the system's web page on the server, they immediately see the login GUI connected to the login manager. Once logged in, a user can use the GUI to change configuration parameters managed by the config manager, add and remove contacts in the contact list managed by the contact manager, and modify user information managed by the user profile manager. Presence information is automatically updated by the presence manager. There are four presence options: online, offline, away and busy. The chat manager handles synchronous text chat and semi-synchronous or asynchronous video chat. The chat manager only comes into play when a chat starts. It handles video with RTMP or HTTP depending on configuration. The log manager records chat activity for subsequent analysis. It logs the time and the status of each chat, and saves the information in log files in a MySQL database.

We have done a simple demo and shown it to a staff in DCCT. He gave us very useful advice about the size of video screen. Almost all the web cameras installed in PC put their focus on the faces. However, sign language users need to communicate with others by looking their eyes, signing at the same time. They have to put the web camera distantly to use it, or sign before their faces. This problem shows that the existing systems have a limitation, and we want to solve the problem in this research. A possible solution is that using a digital camera instead of web camera. Digital camera often has a wide angle lens.

After implementing the system, we will collect both qualitative and quantitative data when testing out the system with actual Deaf users at the Bastion. The qualitative data will be collected with semi-structured interviews, focus group and overall user observation. We will instrument the software to automatically collect quantitative system performance and usage data. Tools such as a scatter diagram and histogram will be used to analyse it. The goal is to triangulate qualitative and quantitative data along the lines of [1].

IV. Conclusion and future work

In conclusion, we want to build and test out a browser-based sign language communication system with Adobe Flash. We want to examine the performance and quality of such a system in two modes, asynchronous and semi-synchronous, with actual Deaf users. We intend to build all system components with open source software. We hope to use such a tool to provide web-based sign language Deaf-to-Deaf communication on a PC, on a cell phone and eventually use it with a VRS service.

ACKNOWLEDGMENTS

The authors thank the staff and members of DCCT for their participation in the project. We also thank Telkom, Cisco and THRIP for financial support via the Telkom Centre of Excellence (CoE) programme.

References

[1] Z.Y. Ma ZY and W.D. Tucker, Adapting x264 to asynchronous video telephony for the Deaf, *Proc. South African Telecommunications Networks and Applications Conference, (SATNAC 2008)*, Wild Coast Sun, Eastern Cape, South Africa, 2008, 127-132.

[2] STATSSA. Primary tables South Africa: Census '96 and 2001 compared: *Statistics South Africa*, 2004.

[3] M. Glaser and D. Aarons, A Deaf adult literacy collective, *Stellenbosch Papers in Linguistics*, 2002, 34, 1-18.

[4] A. Cavender, R.E. Ladner, and E.A. Riskin, “MobileASL: Intelligibility of sign language video as constrained by mobile phone technology, *Proc. 8th international ACM SIGACCESS conference on computers and accessibility*, New York, NY, USA, 2006, 71-78.

[5] S. Cox, M. Lincoln, J. Tryggvason, M. Nakisa, M. Wells, M. Tutt, and S. Abbott, TESSA, a system to aid

communication with deaf people, *Proc. 5th international ACM conference on assistive technologies*, New York, NY, USA, 2002, pp. 205-212.

Yuanyuan. Wang is currently a Masters student at the University of the Western Cape (UWC) with the Bridging Applications and Networks Group (BANG).

William D. Tucker is a senior lecturer in computer science at UWC and leads BANG research there.