Power Saving Strategies for Two-Way, Real-Time Video-Enabled Cellular Phones

Jessica Julie Tran

A thesis submitted in partial fulfillment
of the requirements for the degree of

Master of Science in Electrical Engineering

University of Washington

2010

Program Authorized to Offer Degree:
Department of Electrical Engineering

University of Washington
Graduate School

This is to certify that I have examined this copy of a master's thesis by

Jessica Julie Tran

and have found that it is complete and satisfactory in all respects,
and that any and all revisions required by the final
examining committee have been made.

Committee Members:

____
Eve A. Riskin

____
Richard E. Ladner

Date:____

University of Washington

Abstract

Power Saving Strategies for Two-Way, Real-Time Video-Enabled Cellular Phones

Jessica Julie Tran

Co-Chairs of the Supervisory Committee:
Professor Eve A. Riskin
Electrical Engineering

Professor Richard E. Ladner
Computer Science and Engineering

MobileASL provides low bandwidth and low complexity software video encoders to enable real-time video conversations on cell phones, therefore allowing people who are deaf to communicate in their native language, American Sign Language. This thesis presents two alternative power saving algorithms that utilize activity recognition to extend battery resources when using MobileASL. The first algorithm, called *variable spatial resolution* (VSR), adjusts the spatial resolution of transmitted videos. The second algorithm applies a technique called *variable frame rate* (VFR) and VSR to adjust both frame rate and spatial resolution of transmitted videos. My goals are to implement both power saving algorithms in the MobileASL software program; conduct a battery power study to determine the duration of the battery life while using each algorithm; and conduct a web-based user study to determine if participants could perceive changes in video quality.

The battery power study revealed that running MobileASL without any power saving algorithms consumes 99.7% of the phone's CPU and a full battery charge lasts on average 284 minutes. Implementing VFR or VSR algorithms separately extend the battery life to an average 307 and 306 minutes respectively and lowers the CPU usage to

26% and 32% respectively. Applying VFR and VSR algorithms together extend the battery life to 315 minutes and lower the CPU usage to 10%.

The experimental design of the user study was a 2 x 2 within-subjects factorial design. Major findings include discovering a significant VFR*VSR interaction, ($F_{1,15}$=5.3, p<.05), which led to determining that applying VSR reduces the extent to which participants perceive VFR to induce choppiness. Also, the application of VSR did cause participants to perceive blurry video quality,($F_{1,15}$=21.2, p<.003), and participants found the blurriness to be distracting, ($F_{1,15}$=10.1, p<.01).

The battery power study revealed that applying VFR, VSR, and both VFR and VSR all extend the battery life of a cell phone running MobileASL. Applying both VFR and VSR was found to extend the battery life the most. Therefore, the recommendation is for MobileASL to adopt the use of both VFR and VSR algorithms to extend the battery duration of the cell phone.

TABLE OF CONTENTS

LIST OF FIGURES

Figure Number Page

LIST OF TABLES

ACKNOWLEDGEMENTS

I would like thank my advisors Professor Eve A. Riskin, Professor Richard E. Ladner, and Professor Jacob O. Wobbrock for their guidance and support during my graduate studies. I would also like to thank the undergraduate students Joy Kim, Rafael Rodriguez, and Sheri Yin for their hard work and enthusiasm while creating the user study website; Tressa Johnson for her input and recording the ASL instructional videos; Rahul Vanam for suggesting the VSR idea; Neva Cherniavsky and Jaehong Chon for their assistance with the MobileASL code; and Anna Cavender for her assistance with the IRB process.

DEDICATION

To my parents Julie and Simon and my brother Justin.

"Even the largest task can be accomplished if it is done one step at a time."

CHAPTER 1: INTRODUCTION

Mobile cellular technology has become the most popular and widespread portable technology with an estimated 4.6 billion users [33]. The cell phone has greatly evolved since the first public cell phone call made in 1973. The ability to text message was not introduced until the early 1980s [29]. Today, cell phones are becoming more like small computers with the ability to access the Internet and they are more widely available and easier to use with users ranging from children as young as six to the elderly. With the increasing portability of cell phones, text messaging has become more popular than voice calls in the United States [17]. In 2008, teenagers were leading the increase in mobile text messaging, with the average teenager sending and receiving an average of 1,742 text messages per month. The typical mobile user sends and receives on average 357 text messages per month while only placing and receiving 204 mobile voice calls [17]. Mobile text messaging is superseding phone calls and influencing how future mobile phones are created.

For the Deaf and Hard of Hearing community, text messaging was available before cell phones were even created. Invented in 1964, the teletypewriter (TTY) is a typewriter connected to an acoustic coupler that sends text messages over the telephone lines in real-time [9]. This enabled members of the Deaf and Hard of Hearing community to communicate with family and friends using the available telephone network. Similarly, with the mobile phone revolution, cell phones like the T-Mobile Sidekick and Black Berry now allow for mobile text messaging.

In the United States, text messages are conveyed in English. However, American Sign Language (ASL) is spoken natively in the Deaf community in the United States and English-speaking Canada [19, 22], and therefore English is considered their second language. Like other languages, ASL is a genuine language with a distinct grammar and syntax similar to spoken languages. Conversations spoken in ASL are similar to spoken language in that multiple people may "hold the floor" at once [14]. In addition, ASL contains back-channel feedback [15] where the listener acknowledges understanding, similarly to how a hearing person says "uh-huh." Therefore, ASL conversation can be divided into two parts: signing and not signing because ASL conversation involves turn-taking (times when one person is signing while the other is not). A former graduate student who worked on MobileASL, Dr. Neva Cherniavsky, created a method called activity recognition (discussed in Chapter 2) which identifies signing and not signing parts of a conversation [11]. Activity recognition is also used in the implementation of the alternative power saving algorithms.

ASL is different from spoken languages because information is conveyed through the use of different hand movements and facial expressions. Different messages can be conveyed through facial expressions such as a simple head shake, the position of their eyebrows, and the use of the mouth. People receiving signs primarily focus their attention on the mouth rather than the hands of an ASL signer [6, 24, 27]. Therefore, ASL and other sign languages are considered visual languages [16] because information is not communicated with the use of sounds.

Text messaging is not the optimal means of communication, especially for native ASL speakers, because it is slow. Text messaging can only convey 5-25 words per minute (wps) while signed and spoken languages convey 120-200 wpm [20]. Personal computer-based video communication technologies, such as Skype and different instant messenger software, allow for real-time video, but at the expense of the mobility of the user. It is also possible for people who are Deaf to communicate with hearing people through the use of video phones and video relay services (VRS). Figure 1 demonstrates how VRS works.

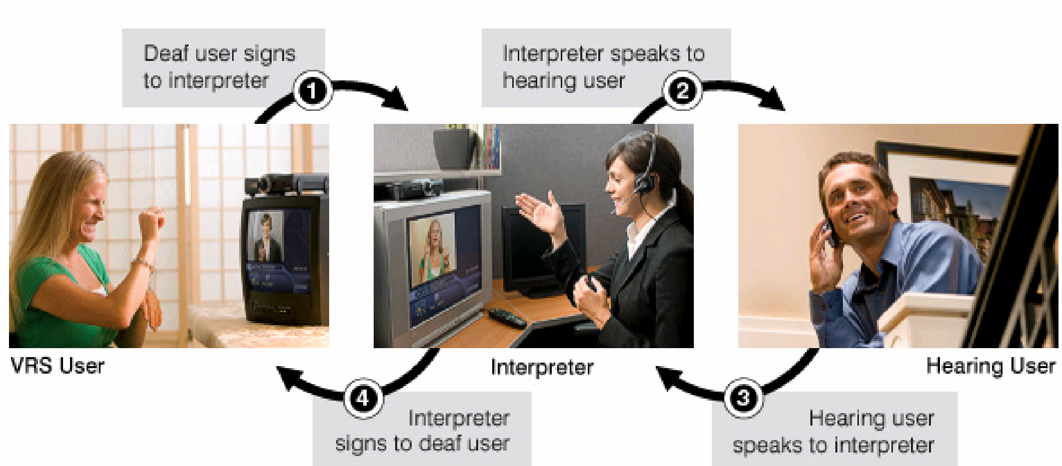


Figure 1: Example of how VRS works [1].

To use VRS, the user who is Deaf signs through a video phone to an interpreter, who translates the ASL message into English and speaks the message on a regular telephone to the hearing user. Since 2002, all VRS services are federally subsidized nationwide in the United States.

Currently, the United States cellular network does not support real-time video conversation; therefore only stationary means of communicating in ASL are available. Eventually, the capabilities of the mobile phone will allow effective means of mobile communication for users who are Deaf, besides just text messaging.

1.1 MobileASL

In a collaborative project with the University of Washington and Cornell University, the NSF-funded MobileASL project is addressing the need for alternative mobile communication for Deaf users. The MobileASL project has developed a software video codec which allows very low bit rate coding (under 30 kilobits/second) to transfer real-time video on the current cellular network in the United States, therefore allowing users who are Deaf to be mobile while using video phones. The codec designed is based on the open source x264 implementation of the H.264 standard [7, 25] which allows for the use of any readily available x264 decoder. A user-interface has also been created to allow users to access and use the MobileASL codec.

The HTC TyTN II cellular phone running Windows Mobile 6.1 operating system was selected as the device to run the MobileASL codec because it has a front facing camera and screen which allow the user to see the screen while the camera is capturing the user. This phone has the capability to prop itself up on a table at an angle during conversations (see Figure 2).

Figure 2: HTC TyTN II cellular phone.

The current version of MobileASL has a captured video size of 96x96 pixels. The video is sent to the x264 encoder before being transferred over a wireless or 3G cellular network. During the decoding process, the video is enlarged to QCIF (176x144 pixels) and shown in the MobileASL application. The video capture size was purposely set to be smaller than the display size to transmit 10-12 frames per second (fps) and increase intelligibility of conversations. Sending smaller video frames results in less data to process which allows for the higher frame rate. An older implementation of MobileASL had a video capture and display size of QCIF; however the frame rate was lower (7-8 fps). The previous power saving study, described in Chapter 2, was conducted using the QCIF capture and display video size with the 7-8 fps transmission rate. The research I present in my thesis uses the current implementation of MobileASL.

The objective of MobileASL is to provide real time video communication using off-the-shelf mobile phones. However, we still face challenges in overcoming limited bandwidth, low processing speed, and limited battery life on the HTC TyTN II cell phone.

● **Limited bandwidth:** In the United States, most mobile phone networks use 2G (second-generation wireless telephone technology) networks that were mainly built to support voice services and slow data transmission. The 2G networks can support a bandwidth of around 30-50 kbps [28] using General Packet Radio Services (GPRS). Japan and Europe, however, have 3G networks that are already providing mobile sign language communication. For instance, in Sweden mobile sign language communication is available because their networks transfer video at a bit rate of 64 kbps, which is considerably higher than MobileASL transmitting video at 30 kbps. Video transmitted over the cellular network places heavy loads on the system, and 3G networks can allow more streaming video. Unfortunately, 3G is currently only available in select locations in the United States as Figure 3 displays.

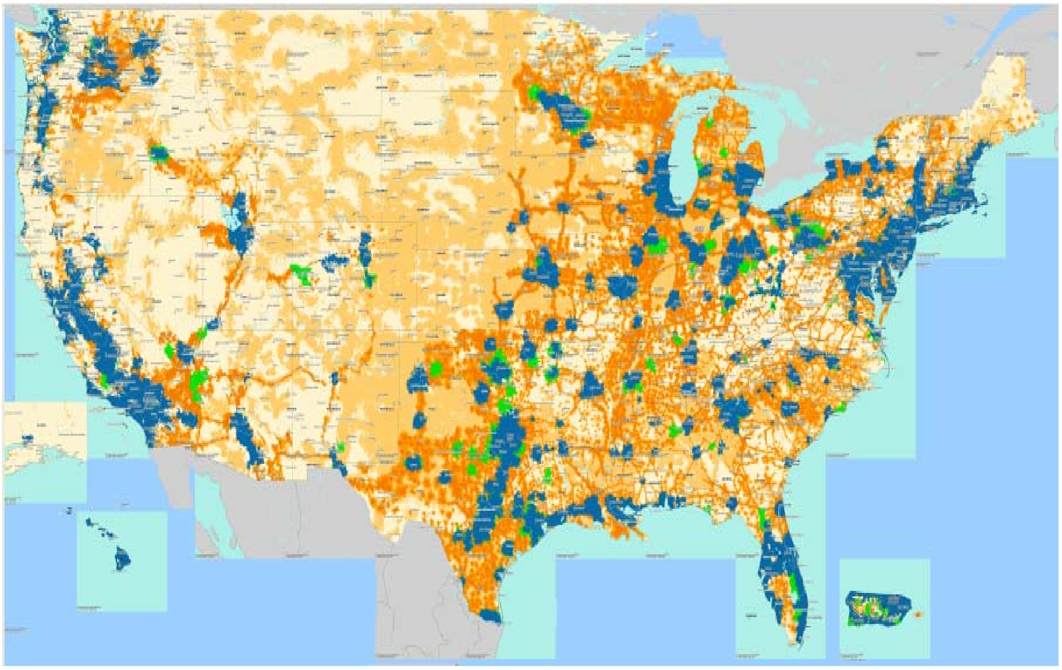


Figure 3: AT&T's coverage of the United States, September 2009. Blue and green is 3G; dark and light orange are EDGE and GPRS. The rest is 2G or no coverage [34].

● **Low processing speed:** Cell phones capable of running operating systems like Windows Mobile have very limited processing power. The HTC TyTN II cell phone's processor has a clock speed of 400MHz while the processor of a desktop computer has a clock speed of 2.66 GHz or higher. The demand placed on a cell phone to transmit video in real time consumes the battery resources in a short period of time.

● **Limited battery life:** The HTC TyTN II cell phones use lithium-ion batteries which degrade over time [10]. Furthermore, running the MobileASL application has high power consumption rates that drastically deplete a full battery charge from 40 hours to 2.5 hours [13]. A big contribution to the battery drain is the cell phone's touch screen which consumes 80% of the phones energy alone [18]. Addressing the limitations of battery life is difficult because while larger bandwidth and advancements in processor speed will be available in the future, battery life is not keeping up with Moore's law as demonstrated in Figure 4.

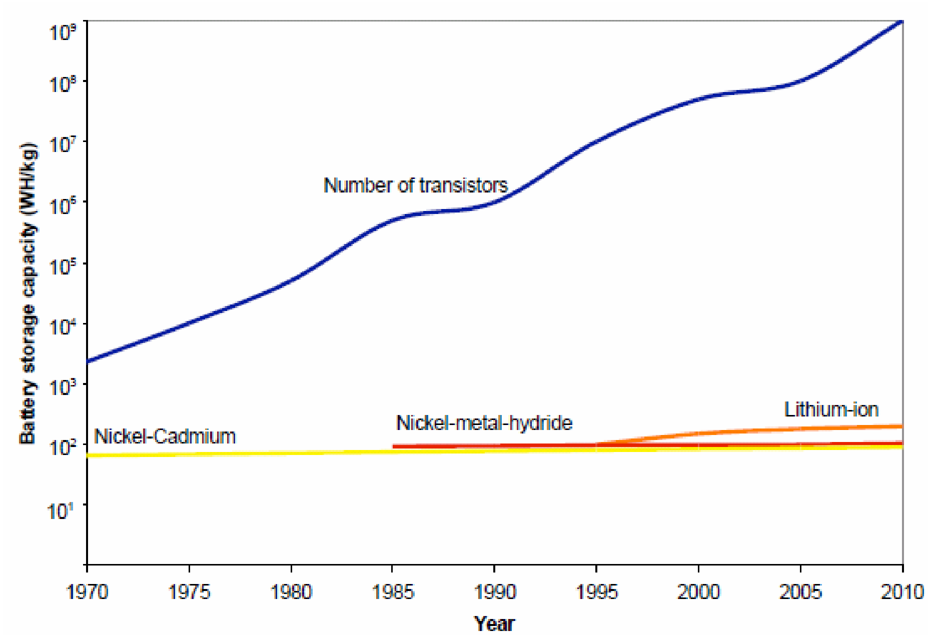


Figure 4: Battery storage capacity (measured in watt hours per kilogram) versus number of transistors, on a log scale [21].

Previous research conducted by Dr. Cherniavsky investigated a method to extend battery resources which is called *variable frame rate* (VFR) [12]. The VFR algorithm has been incorporated into the MobileASL platform as one method to save battery resources. VFR manipulates the temporal resolution of the transmitted video to save computational resources. Activity recognition is used to determine when a person is signing or not signing. (Chapter 2 will discuss how frames are identified as signing or not signing in MobileASL.) When a person is signing, video is transmitted across the cellular network at 10-12 fps. When the person is not signing, the frame rate is reduced to 1 fps, which produces a choppy video quality. Figure 5 demonstrates how video frames are reduced during the not signing portions of a conversation.

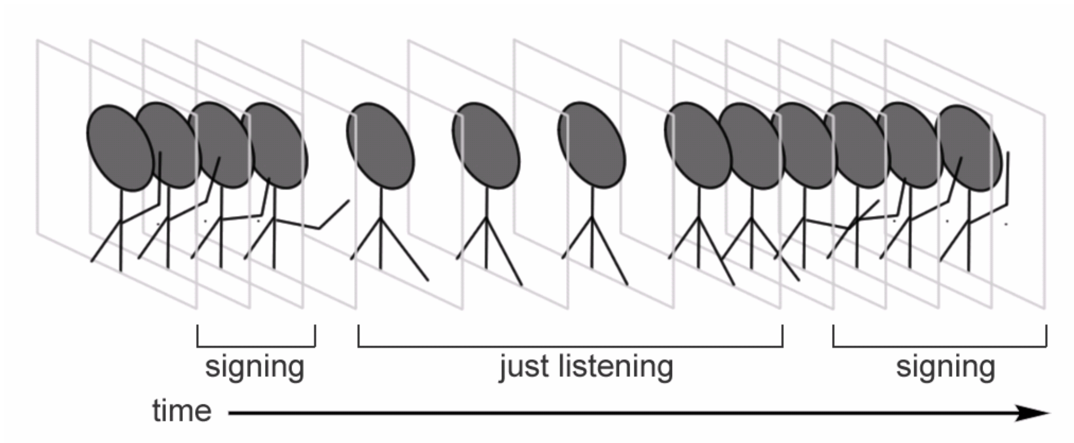


Figure 5: Example of variable frame rate implementation. From left to right, the frame rate decreases when the signer is not signing (just listening), resulting in “choppy” video quality from [12].

Chapter 2 will detail how the VFR algorithm extends the battery life by reducing the number of frames sent to the encoder during the just listening portions of a conversation. Even though the implementation of VFR was found to save battery power, participants from a user study investigating the human perception of VFR commented how they were unsure if the choppy video quality was due to the VFR algorithm or because of a bad wireless internet connection. The feedback and results from the user study investigating the VFR algorithm led to my investigation of two alternative algorithms to extend the battery life of the cell phone.

1.2 Contributions

My thesis presents two alternative power saving algorithms that prolong battery resources when using MobileASL. The first algorithm adjusts the spatial resolution of not signing video and the second algorithm adjusts both frame rate and spatial resolution of not signing video. My goals are to implement both power saving algorithms in the MobileASL software program; analyze the battery duration of each algorithm; and conduct a user study to determine if MobileASL users are willing to sacrifice video quality to prolong the battery life of the phone.

The first alternative power saving algorithm, called *variable spatial resolution (VSR)*, manipulates the spatial resolution of not signing (just listening) portions of a conversation to use less battery power while maintaining intelligibility. The VSR algorithm downsamples not signing frames to 1/4 of the original size before encoding. This method provides a constant stream of video while reducing the amount of data that needs to be encoded and processed. Figure 6 demonstrates the implementation of VSR for just listening portions of a conversation.

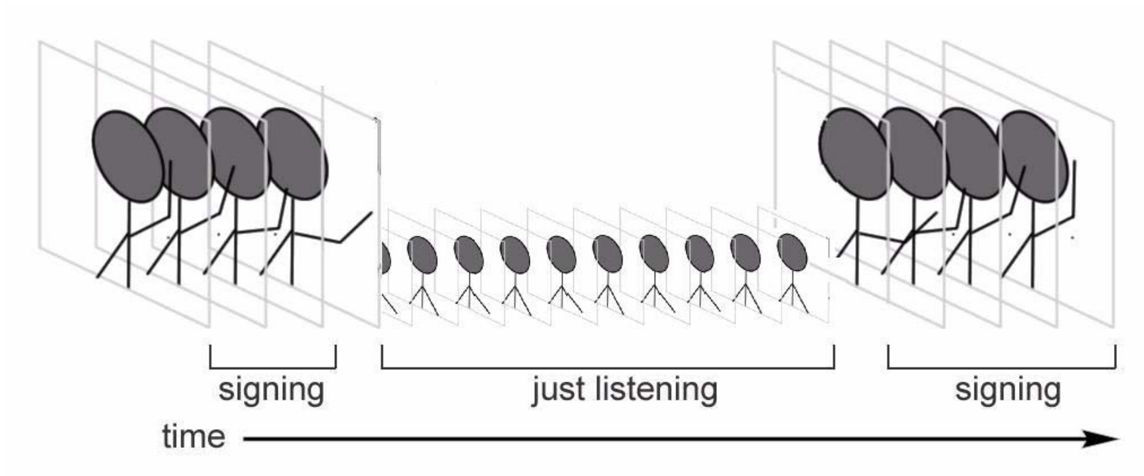


Figure 6: Example of variable spatial resolution implementation. From left to right: the not signing (just listening) frames down sample to 1/4 the original frame size, resulting in "blurry" video quality.

As Figure 6 shows, the VSR algorithm maintains the video transmission rate of 10-12 fps, but downsamples the not signing video frames to one-fourth of the original size, which produces a blurry video quality.

The second alternative power saving method is the combination of VFR and VSR. Intuitively, combining the two methods should produce further power savings. Sending less data per frame and fewer frames per second would be computationally less intensive than the algorithms individually. Figure 7 displays how the just listening frames will be processed.

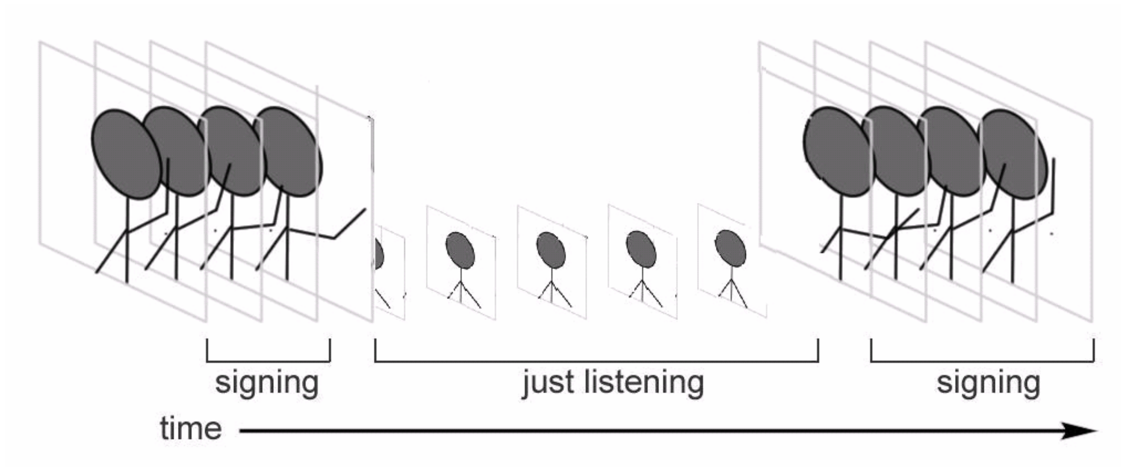


Figure 7: From left to right: the implementation of VFR and VSR during the just listening portions of a conversation. The resulting output is a combination of blurry and choppy video.

To prove that the two alternative power saving algorithms indeed prolong battery resources, I implemented both power saving algorithms and demonstrated that they save system resources while producing intelligible ASL conversations. I intend to also demonstrate that potential MobileASL users are willing to sacrifice video quality to gain longer battery life.

1.2.1 Power Saving Study

To determine if each encoding algorithm is producing significant changes in battery duration, I quantified the results by measuring (1) battery duration, (2) current drain, and (3) CPU use of the cell phone in a battery power study of each algorithm as well as during the use of no algorithms which I call "default." Chapter 3 will detail how the power saving study was conducted. The HTC TyTN II phone uses a lithium-ion battery that can hold up to 1350 mAh of charge [18] which can vary between phones. Therefore, the results recorded from four new HTC TyTN II phones were averaged and reported in the power savings study. Measuring and comparing the battery drain, current consumption, and CPU usage will give better insight of how the phone works to better understand where the battery resources are allocated.

1.2.2 Power Savings User Study

The implementation of the three power saving algorithms is only beneficial if they are utilized by MobileASL users. Chapter 4 details the web-based user study conducted to investigate the human perception of video quality resulting from the use of the different power saving algorithms.

CHAPTER 2: RELATED AND PREVIOUS WORK

2.1 Related Work

A unique aspect of the MobileASL project is that we are creating a technology that takes into consideration how ASL speakers communicate with one another, specifically members of the Deaf and Hard of Hearing community. Conversations spoken in ASL are similar to spoken language in that multiple people may "hold the floor" at once [14]. In addition, ASL contains back-channel feedback [15] where the listener acknowledges understanding, similarly to how a hearing person says "OK" or "uh-huh." Since users of MobileASL may "sign over one another," we investigate if modifying the spatial or temporal resolution of the video when one person is not signing negatively affects intelligibility.

Sign language recognition is a related research topic where the objective is to translate sign language into English text. There are many active research topics detailing the most recent and state-of-the-art applications [23, 24]; however, the goal of MobileASL does not include interpretation or translation of ASL. Instead, we focus on increasing mobile communication accessibility for ASL speakers. We intend to transmit live streaming video using the mobile phone processor while conserving battery power consumption.

Previous work on the human perception of video content showed that motion is an important factor that influences how Deaf people interpret sign language [24]. Peripheral low-resolution vision is a key component in the perception of motion. Muir and Richardson explored how Deaf people view sign language in video and the application of this to design video communication systems [24]. Their findings concluded that a Deaf viewer focus is placed on the facial region of a signer in order to pick up the small detailed movements in the signer's facial expression and lip shapes. This region is of interest because it conveys important sign language information to the receiver. My work would like to explore the acceptable degradation of video quality before intelligibility of ASL is compromised. ASL conversation can be divided into two parts: signing and not signing because ASL conversation involves turn-taking (times when one person is signing while the other is not). I apply an encoding algorithm during the not signing, i.e. "just listening," portions of a conversation to lower spatial resolution since there will be less motion by the listener.

Another related research topic deals with the automatic activity analysis of video and is active in the computer vision community. Conversational sign language video is not widely studied, but there are related problems that can be applied in the MobileASL project. Shot change detection [14] determines when a video changes scenes in order to automatically parse and extract key frames. However, since shot change detection is usually not done in real-time analysis, most existing algorithms analyze the entire video at once. Finally, the videos used for analysis have large difference between scenes, while our videos only have minor changes between the signing and not signing portions of video. Using the concept of shot change detection, Dr. Cherniavsky developed the MobileASL activity recognition scheme that distinguishes between signing and not signing portions of video [11].

2.2 Previous Work

2.2.1 Activity Recognition

Dr. Cherniavsky's activity recognition scheme is based on calculating the sum of absolute differences of the luminance component of consecutive frames to identify signing and not signing frames [12]:

$$d(k) = \sum_{i,j \in I(k)} | I_k(i,j) - I_{k-1}(i,j) |$$

Frames that are classified as signing may contain a lot of activity such as fast movement in the hands or face, resulting in large pixel differences. Frames that are similar will have small pixel difference due to less movement by the user. Therefore, less movement will result in frames classified as not signing. If the difference between each frame was above a certain threshold, then the frame was classified as signing; otherwise, not-signing. The threshold value was determined by implementing the sum of absolute difference on multiple training videos [12]. This method was found to be sensitive to extraneous motion in the background; however, previous research conducted on our baseline differencing method discovered that this worked well with correct recognition rates averaging at 84.6% [12]. Further, the method is fast enough to work in real-time on the cell phone. Figure 8 demonstrates a general overview of activity analysis implemented in MobileASL.

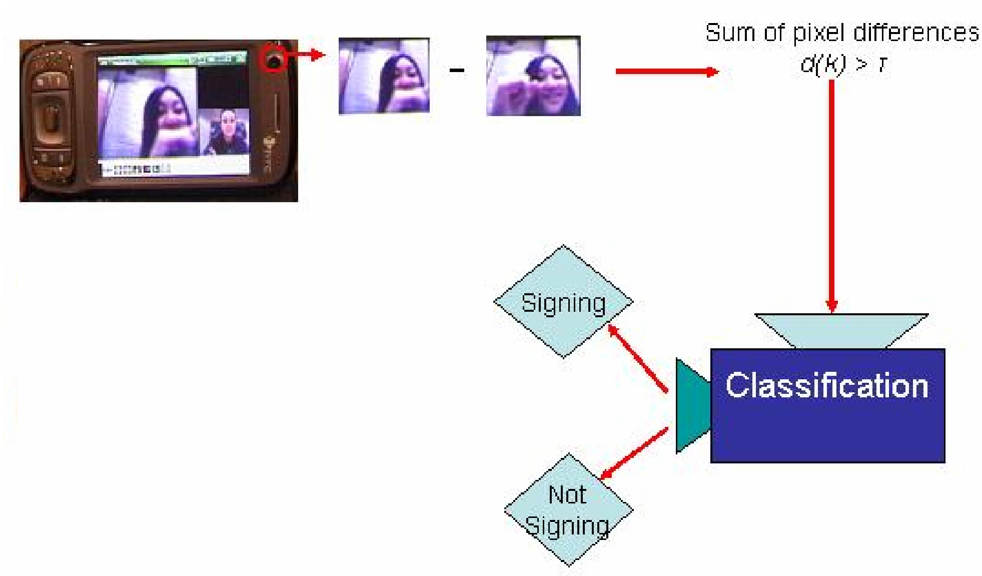


Figure 8: Overview of activity analysis from [12].

2.2.2 Variable Frame Rate Algorithm

Another research topic of interest is the effects of frame rate on sign language instruction [28]. Johnson and Caird discovered that 1 and 5 frames per second (fps) were enough for beginners to learn ASL from ten videos, each containing one sign. Sperling et al. also found considerable reduction in comprehension of ASL when frame rates where reduced from 10 to 5 fps, slight reduction in comprehension from 15 to 10 fps, and an insignificant difference in comprehension from 30 to 15 fps [30].

Previous research conducted by Dr. Cherniavsky on the MobileASL project investigated four different frame rate combinations of video to determine the users' intelligibility of video content. The videos used contain "conversationally-paced" signing by people who are Deaf that are fluent in ASL, which result in many quickly produced signs. For the signing portions of a video the frame rate was set at 10 fps, since previous studies indicated that this value was adequate for sign language intelligibility. For the not signing portion, Dr. Cherniavsky studied 0, 1, 5, and 10 fps. The 0 fps corresponds to showing one frame for the duration of a not signing segment (a freeze-frame effect). Figure 5 from Chapter 1 demonstrated how video frames are reduced during the just listening portions of a conversation. Reducing the number of frames sent to the encoder during the just listening portions of a conversation revealed that there is a significant power savings due to less transmission of data, processor cycles, and power consumption [11]. Figure 9 demonstrates the reduction of the average processor cycles when encoding video at 10 fps, 5fps, and 1fps.

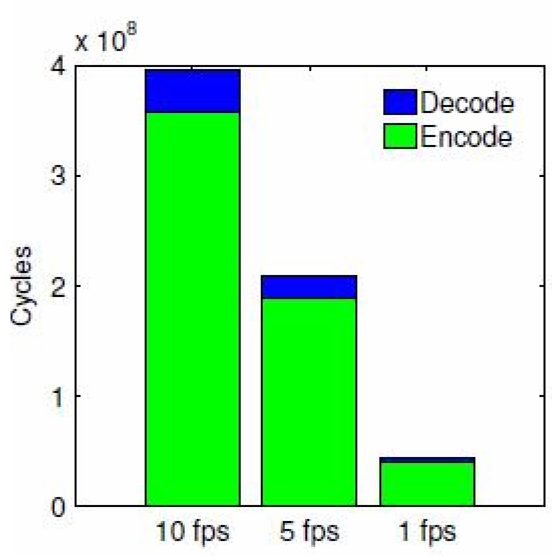


Figure 9: Average processor cycles for 10 fps, 5 fps, and 1 fps. From [11].

Since the smallest number of processor cycles is needed for encoding video at 1 fps, 1 fps was selected as the reduced frame rate speed when transmitting not-signing frames. This method is called *variable frame rate (VFR).* Figure 10 demonstrates how the identification of signing and not-signing frames during activity recognition impacts the frame rate of videos transmitted.

.

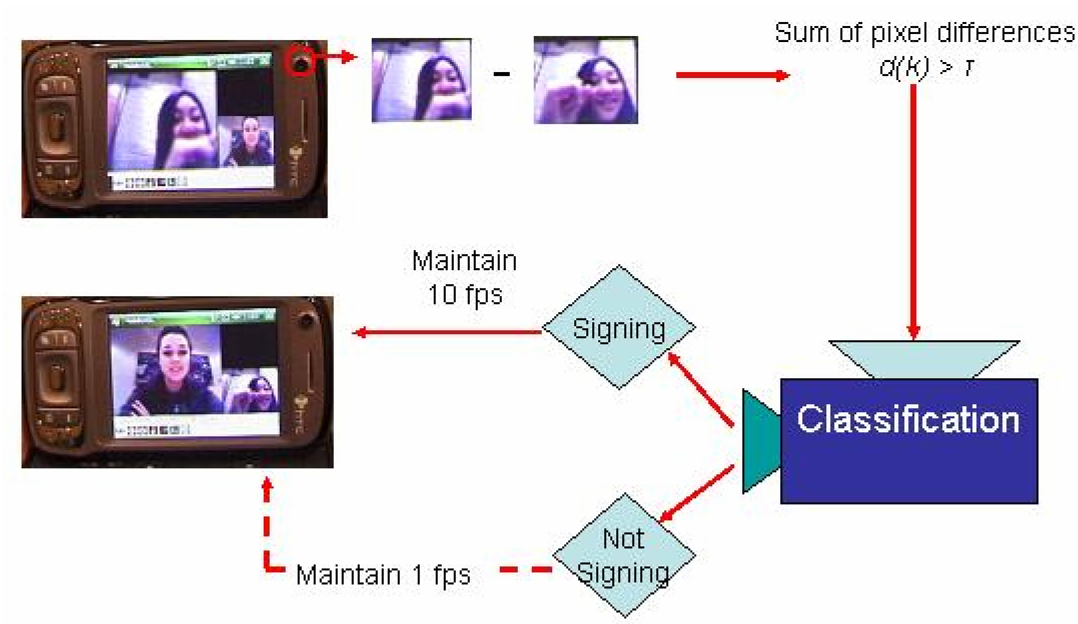


Figure 10: Activity recognition used to identify signing/not signing frames to influence the video frame rate from [11].

To determine if battery consumption was reduced when using the VFR implementation, a simulated sign language conversation was conducted to monitor power usage for half an hour on two phones. The first experiment measured the battery drain when VFR was implemented on both phones and the second experiment measured the battery drain when the default (no algorithm) was applied. The default setting of MobileASL maintains a transfer of QCIF (176x144) video frames at 10-12 fps during all portions of a conversation (signing and not-signing). The simulated conversation consisted of high motion for one minute (to simulate signing), no motion for the next minute (simulating not-signing), and so on. Figure 11 shows the battery durations of the mobile phone for the VFR and default implementations.

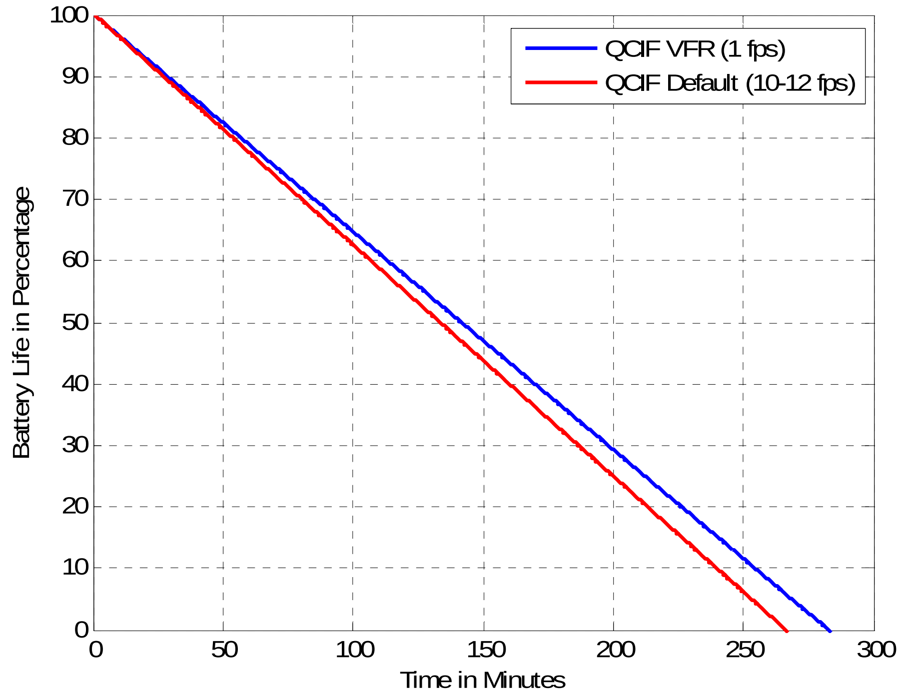


Figure 11: Battery life of cell phone for applied VFR and default implementations of MobileASL.

As Figure 11 demonstrates, there is a noticeable extension of battery life when VFR is implemented. Regression analysis shows that the rate of battery depletion is linear. Therefore the measurements were extrapolated and we found that the battery duration for the default and VFR settings are 266 and 283 minutes respectively. A preliminary user study in [11] was conducted to determine if varying the frame rate negatively affects conversation intelligibility. Participants commented that the perceived choppiness of the video resulted in uncertainty if the video call was over or if there was a problem with the connection with the cellular network. Therefore, this leads to investigating alternative approaches to saving battery power, which are discussed in chapter 3.

CHAPTER 3: PHONE IMPLEMENTATION

The successful investigation of altering the temporal resolution (VFR) to prolong battery life (as discussed in chapter 2) encourages the investigation of two alternative power saving algorithms: variable spatial resolution (VSR) and the application of both VFR and VSR. Responses from participants in the user study testing the VFR algorithm expressed that during the times when a person was not signing, they were unsure as to whether or not the video stream had frozen or the conversation had disconnected. Therefore the application of the VSR power saving method is investigated to determine whether or not it prolongs the battery life of one full battery charge. I also investigate the combination of VFR and VSR. In the following sections, I begin with a brief background on the YUV 420 format that is used by the HTC TyTNII cell phone's camera. Then, I discuss the phone implementation of VSR and the combination of VSR and VFR. Finally, I discuss the results of a power study which determined which power saving algorithm (VFR, VSR, and combination of VFR and VSR) conserves the most battery life.

3.1 YUV 420 Format

A video frame consists of pixels that represent a 2-dimensional image. In order for a frame to be transmitted over a cellular network, it must be transmitted in packets over a cellular network, and decoded by the recipient. The HTC TyTNII cell phone camera captures video frames in YUV 420 format. Figure 12 is an example of a 4x4 pixel image and its YUV 420 representation.

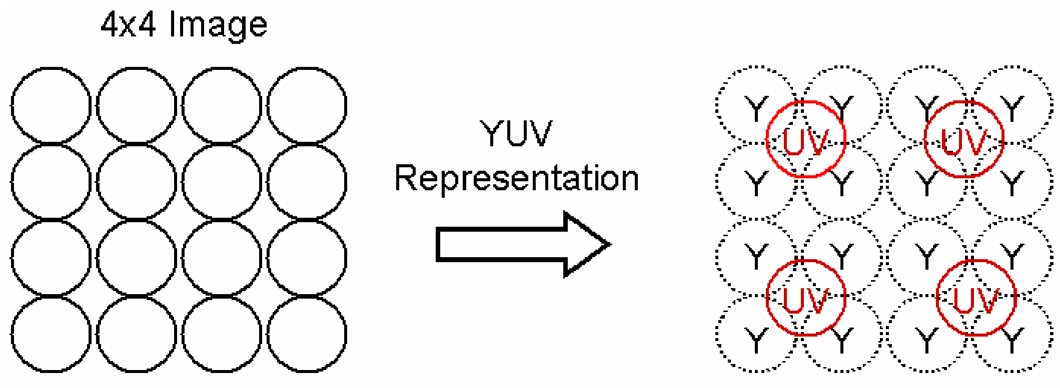


Figure 12: Example 4x4 image and YUV 420 representation.

YUV 420 is the color space that encodes a video while taking human perception into consideration. The Y component is the luminance or brightness of a pixel and UV is the chrominance, which is the color component. YUV 420 specifies that for every four luminance components, there is one chrominance component representing the color of those four pixels. In Figure 12, the UV components (shown in red) are each assigned to four luminance components.

3.2 Variable Spatial Resolution

The first alternative power saving algorithm is variable spatial resolution. The VSR algorithm is based on downsampling the width and height of each frame by a factor of 2, as seen in Figure 13.

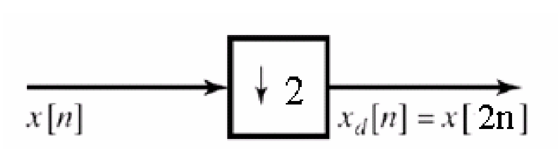


Figure 13: Downsampler Block Diagram.

When implementing VSR, I had to consider how each video frame was represented in the YUV 420 format. The luminance component is downsampled as represented in Figure 13 since there is one luminance component for every pixel that makes up the frame. However, there is only one chrominance component for every four luminance components; therefore the average of four consecutive chrominance values were calculated and used for the down sampled luminance component.

VSR is similar to VFR in that it uses the identification of signing or not signing frames during activity recognition to choose when to down sample the captured video frames. The VSR algorithm maintains the video transmission rate of 11 fps, but down samples the not signing video frames to 1/4 of the original size before being sent to the encoder. After the down sampled frame is transmitted, it is decoded and enlarged to QCIF (176x164) and displayed on the cell phone screen with a blurred video quality.

Figure 14 is an example of the degree of video quality degradation when VSR is applied to a not signing frame. Section 3.4 will discuss the power measurements from the VSR implementation.

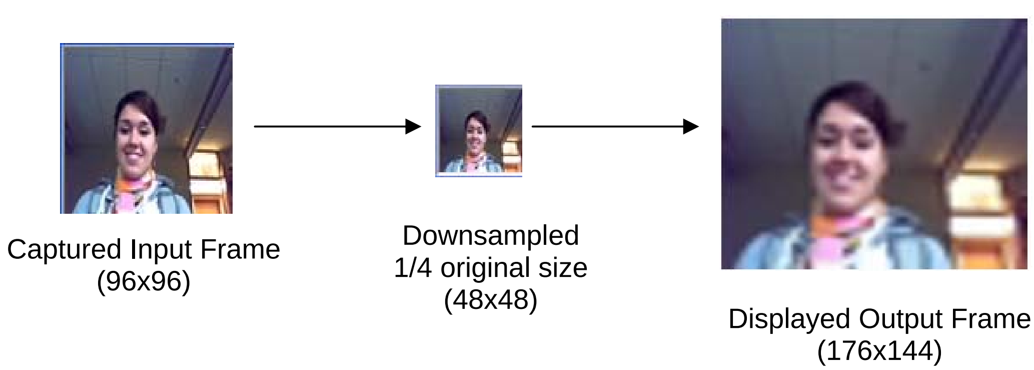
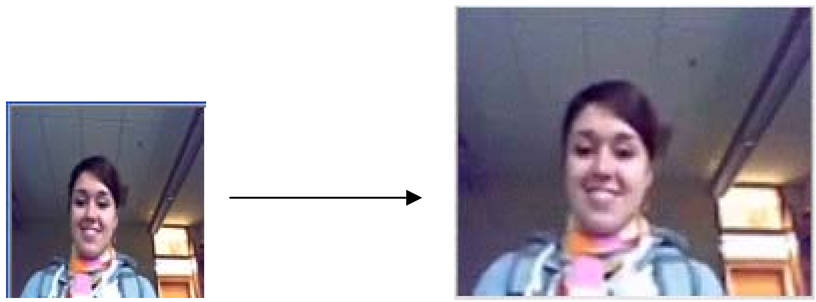


Figure 14: Example of a not signing frame downsampled to 1/4 of original size, which produces a blurry video quality.

Figure 15 is an example of the video quality when no power saving algorithms are applied. This is the default implementation of MobileASL.



Captured Input Frame
(96x96)

Displayed Output Frame
(176x144)

Figure 15: Example of a not signing frame with default implementation of MobileASL.

3.3 Combination of Variable Frame Rate and Variable Spatial Resolution

The second alternative power saving algorithm is the application of both VFR and VSR. When frames are identified as signing, the 96x96 captured video is transmitted at 10-12 fps. When not signing frames are identified, frames are down sampled to 1/4 of their original size before being sent to the phone's encoder for processing and are transmitted at the lower frame rate of 1 fps. The phone receiving the transmitted frames will decode and enlarge the frames to QCIF for viewing on the cell phone. When the combination of VFR and VSR is implemented, the user will view video that is both choppy and blurry. Applying both algorithms results in extending the duration of one full battery charge even more than VFR and VSR, as discussed in section 3.4

3.4 Cell Phone Battery Power Study

With the successful phone implementation of VSR, I want to quantify the battery duration of applying only VSR, and both VFR and VSR, and compare the results to the application of VFR and the default setting of MobileASL. All power saving experiments were conducted using the current version of MobileASL. I want to determine the maximum battery duration of the HTC TyTNII cell phone for each algorithm (VFR, VSR, and both VFR and VSR). In order to compare the results for each algorithm, the battery life of the default (no algorithms applied) MobileASL implementation was also measured. The maximum battery life occurs when the power saving algorithm is continuously implemented because fewer resources are used to encode and transmit video depending on the selected power saving algorithm. Therefore, this power study was conducted using the ideal case when the selected power saving algorithm is constantly implemented.

The manufacturers of the HTC TyTNII cell phone specify that a full battery charge can hold 1350 mAh [18]. The minimum current drain for this particular cell phone to operate is 128 mA and the minimum average percent CPU usage is 22.4% to operate the Windows Mobile 6.1 operating system. With this knowledge, a simple formula was used to calculate the battery life of the cell phone:

(Eq. 1) Battery Life in Hours = 1350 mAh / X current drain (mA)

A comparison of battery drain, current consumption, and CPU usage was used during analysis to gain better insight of how the MobileASL application consumes the phones resources. When conducting these experiments the data were collected with both the

MobileASL application and Windows Mobile 6.1 operating system running, unless otherwise noted.

3.4.1 Set Up

For the purpose of this power study, when I say that two cell phones are "holding a conversation," I mean that two cell phones were running MobileASL and transmitting data to one another. To simulate the not signing (just listening) portions of a conversation, two phones in conversation with each other were placed so that they faced a static object (i.e. the wall) for 30 minutes. A publicly available software tool [5] was used to monitor the battery consumption, current drain, and CPU usage of each cell phone during each experiment. There were four experiments conducted: (1) VFR implemented only; (2) VSR implemented only; (3) both VFR and VSR implemented; and (4) no algorithms applied (default). It is important to note that before each experiment, each cell phone needed to be fully charged to capacity to be consistent across all experiments. Also, since the HTC-TyTn II cellular phones use a lithium-ion battery, which degrades over time, the data collected from two different cell phones were averaged and used in analysis.

3.4.2 Battery Consumption

For each experiment, the rate at which the battery drained was logged every five seconds for 30 minutes. As in Chapter 2, regression analysis demonstrated that the battery drain is linear for each experiment, so the battery drain data were extrapolated to determine when the battery discharged to 0%. Figure 16 shows the extrapolated data for the average battery life of the HTC TyTNII cell phone for each experiment. Table 1 lists the average battery duration for each power saving algorithm and for the default setting.

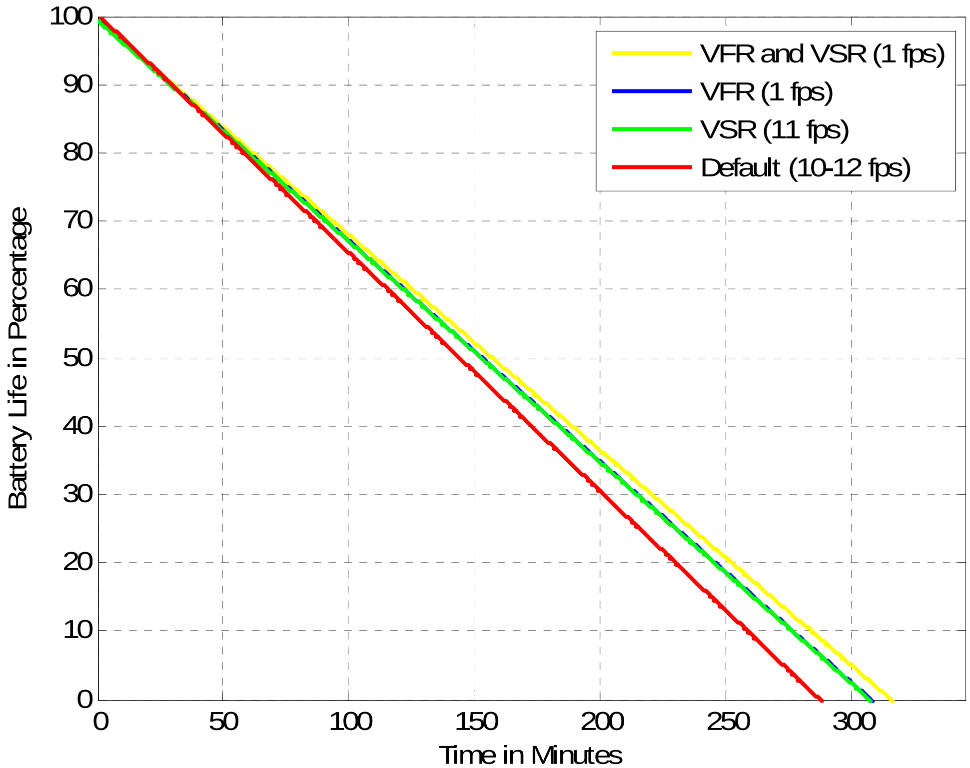


Figure 16: Battery life (percentage) vs. time (minutes) for each algorithm implementation.

Table 1: Average battery life (minutes) for each method.

Method (during 'just listening')	Average Battery Life (minutes)
VFR and VSR (1 fps)	315
VFR (1 fps)	307
VSR (11 fps)	306
Default (10-12 fps)	284

Figure 16 and Table 1 demonstrate that the three battery saving algorithms extend the average battery life of the cell phone. This experiment determined that the average battery life of the cell phone when running the default setting is 284 minutes. The application of VFR, VSR and both VFR and VSR each extend the battery duration on average by 23, 22, and 31 minutes respectively.

The VFR and VSR algorithms performed similarly with only a minute difference, while applying both methods out performed the performance of each algorithm alone by 11 minutes. Looking at the average battery drain for each implementation demonstrates that battery life can be extended when a power saving algorithm is implemented during the not signing sections of a conversation.

3.4.3 Current Drain

In addition to measuring the battery drain, I also recorded the current drained from the cell phone's battery. Similar to recording the battery drain, the value of the current drain was logged every five seconds for 30 minutes. From previously observing battery consumption, I anticipated that applying both VFR and VSR algorithms would consume the least amount of current and the default setting would consume the most current. The current drain for the VFR and VSR algorithms individually was anticipated to have similar current consumption. Figure 17 and Table 2 demonstrate the current drain for each experiment.

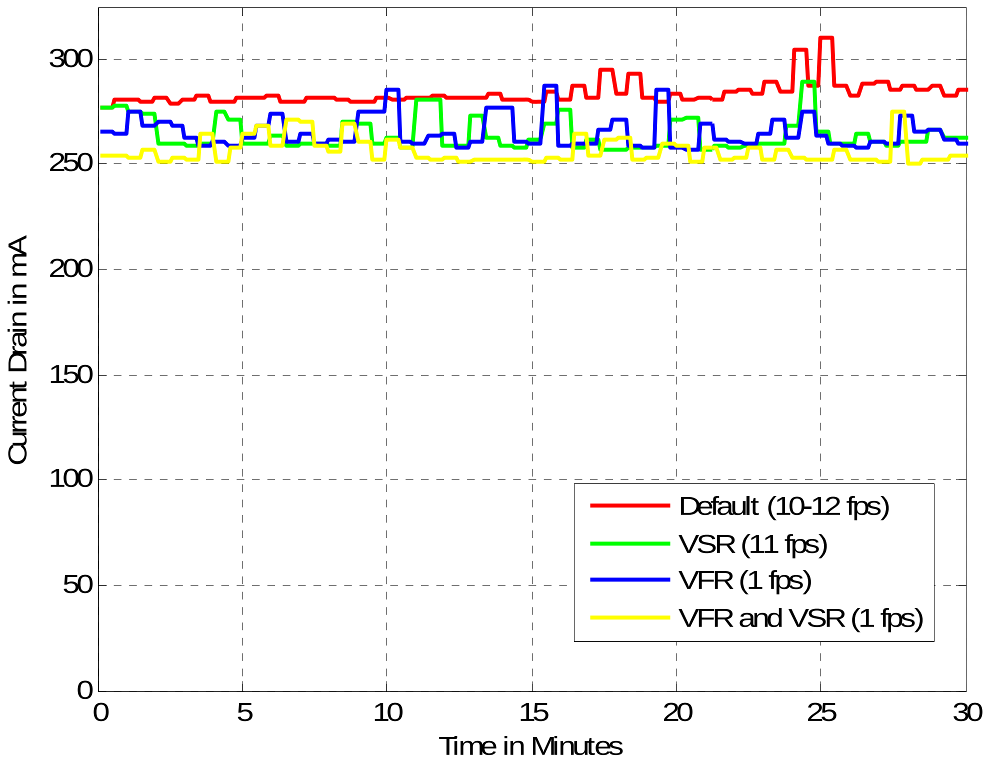


Figure 17: Measured current drain (mA) vs. time (minutes) for each encoding algorithm.

Table 2: Average current drain (mA) for each algorithm.

Method	**Average Current Drain (mA)**
Default	284
VSR	264
VFR	265
VFR and VSR	257

Comparing Figure 16 and Table 1 to Figure 17 and Table 2 demonstrates how the average current drain and average battery duration are related. The default setting drains the most current, and also has the shortest battery life. The current drain for VSR and VFR have similar current drain values, which parallels the similarity of battery duration between these two algorithms. Finally, the applying both VFR and VSR has the least current drain and the longest battery duration. The measured current drain includes running both MobileASL and the Windows Mobile 6.1 operating system.

When running these experiments, I wanted to check that the phones were holding close to 1350 mAh of charge. This was accomplished by using equation 1. I wanted to confirm the results were accurate and the battery charges when starting the experiments were consistent with manufacturer specifications to confirm the accuracy of my results. Also, I also wanted to quantify how much current was just being consumed by MobileASL. Table 3 is the compiled data for battery life; the current drain of the phone when it is just running the Windows 6.1 operating system; the current drain by the MobileASL application only; total current drain; and the total charge held by the battery.

Table 3: Comparison of battery life (minutes) and current drain (mA) of each experiment.

Method	Default	VSR	VFR	VFR and VSR
Battery Life^ (minutes)	284	306	307	315
Current Drain of Phone Only (mA)	128	128	128	128
Current Drain of MobileASL Only^ (mA)	156	136	137	129
Total Current Drain^ (mA)	284	264	265	257
Full Battery Charge^ (mAh)	1344.27	1346.40	1355.92	1349.25

^ Averaged over two phones

Recall that the battery of the HTC TyTNII cell phone is intended to hold 1350 mAh of charge and needs at least 128 mA to run the Windows 6.1 operating system. Therefore, I determined the current drain of the MobileASL application only by using the results of the average current drain for each experiment minus 128 mA.

Table 3 lists the average full battery charge for the cell phones on which the experiments were conducted. On average, the full charge of the battery was 1349 ± 4.79 mAh, which confirms that our formula for the relationship between battery life and current drain correctly used 1350 mAh. These results demonstrate that the phones used for these experiments were capable of being charged to their intended capacity.

3.4.4 CPU Usage

Measuring the battery duration and the current drain alone did not reveal why certain implementations consumed more current than the others. It is suspected that since the VFR and VSR algorithms were altering the temporal or spatial resolutions of the video, this reduced the amount of computations needed to process the video; however, more information was needed which led to monitoring the CPU usage. Figure 18 shows the total CPU usage for each method. Total CPU usage implies the sum of all the applications running on the phone (operating system and MobileASL) and how much of the phone's processor is being consumed.

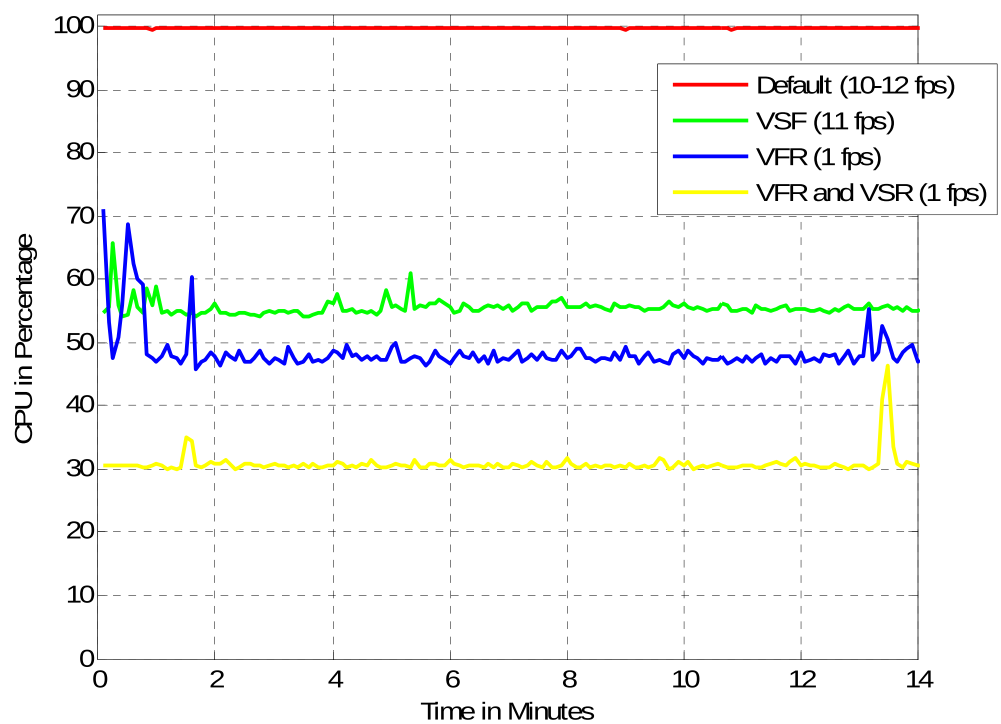


Figure 18: Total CPU usage (percentage) vs. time (minutes).

As Figure 18 shows, the default setting of the MobileASL application uses 99% of the CPU. This large CPU usage could be a possible explanation as to why the battery drains more quickly.

Igor and Cruck investigated how the HTC TyTNII cell phone consumes resources through isolating and measuring different components of the phone. There results found the baseline CPU usage is 22.2% for the cell phone to be functional [18]. Using this knowledge as a reference, the CPU usage for the phone only (when running just the operating system) was measured and the results are shown in Table 4. With these measurements, I determined the percentage of CPU usage for the MobileASL application only. The average CPU usage of the MobileASL application is also shown in Table 4.

Table 4: CPU usage (percentage) for phone only, MobileASL application only, and total system.

Method	Default	VSR	VFR	VFR and VSR
CPU Phone Only^ (%)	23.8	23.7	22.0	20.0
CPU MobileASL Only^ (%)	75.9	31.8	26.4	10.8
CPU Total System^ (%)	99.7	55.5	48.4	30.8

^ Averaged over two phones

As Table 4 shows, the CPU usage of the phone only is around the measured 22.2% found be Igor and Cruck indicating that Windows Mobile 6.1 occupies 22.2% of the operating system. When comparing the CPU usage of the MobileASL application only, there is an impressive drop in the average CPU usage for applying VFR and VSR, with only 10.8% of the CPU used when not-signing frames are transmitted. The default setting still reflects a large CPU usage of 75.9%. The CPU usage of only VFR and VSR shows a slight difference with consuming 26.4% and 31.8% respectively. When the battery duration and current drain were measured for VFR and VSR individually, both algorithms produced similar results. However, here the difference between the two algorithms is more apparent, with VSR consuming 5.4% more CPU than the VFR algorithm. This could be due to the VSR algorithm constantly transmitting 11 fps during the signing and not-signing frames, while in the VFR algorithm, the frame rate is reduced to 1 fps. Finally, applying both VFR and VSR algorithms utilizes the least amount of CPU, which is consistent with the longer battery duration and least amount of current drain.

3.5 Summary

The cell phone battery power study determined that individually, VFR and VSR algorithms both extend the battery life of the default MobileASL implementation by 22 minutes. The combined implementation of VFR and VSR produced a longer extension of battery life of 31 minutes over the default setting. The next step is to investigate which implementations of the power saving algorithms MobileASL users prefer, which is discussed in detail in Chapter 4.

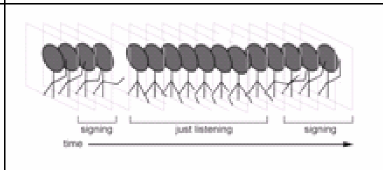
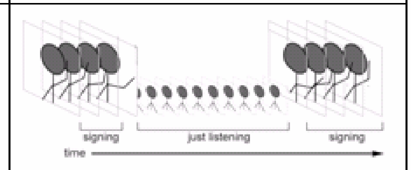
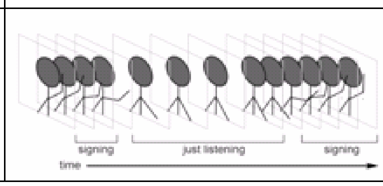
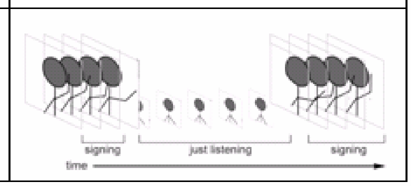
CHAPTER 4: WEB-BASED USER STUDY

A web-based user study was created to evaluate and understand the human perception of video quality when three different power saving algorithms are applied to real-time video transmitted to the cell phones. Recall, our power reduction algorithms rely on activity recognition to differentiate between signing and not signing (just listening) video. A user study is essential to determine if users can detect the changes in video quality caused by each algorithm, especially during the not signing portions of a conversation. The results (discussed in section 4.4) will determine if sacrificing video quality to increase battery life may be acceptable to users.

4.1 Experimental Design

The experimental design of the user study is a 2 x 2 within-subjects factorial design. The two factors (independent variables) are 1) VFR algorithm and 2) VSR algorithm, and the two levels are applying or not applying those factors (i.e., *on* and *off*). Table 5 pictorially demonstrates the four different video applications of interest 1) VFR: off, VSR: off; 2) VFR: on, VSR: off; 3) VFR: off, VSR: on; and 4) VFR: on, VSR: on.

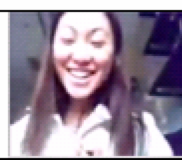
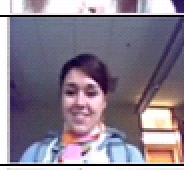
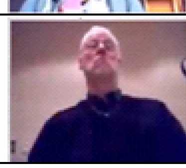
Table 5: 2 x 2 Factorial Within-Subjects Experimental Design

		VSR (factor 2)	
		OFF	ON
VFR (factor 1)	OFF		
	ON		

4.2 User Study Format

This study consisted of a two part web-based survey that people could complete within 7-15 minutes. A web-based study was selected because more participants could be included from across the nation. Since English may not be the natural language of some of the participants, instructional video signed by an ASL certified interpreter were recorded and included in the survey (description in section 4.2). Part 1 consists of background questions (description in section 4.1.1). Part 2 begins with a reminder to the participants that they must install the most current version of QuickTime to view the videos shown in the survey. Next, the participants are presented with an instructions page informing of the survey content (description in section 4.1.2). After the instructions, each participant is randomly assigned to view one of three videos of a person signing in ASL (description in section 4.2.2). The assigned video is shown four consecutive times, but each time a different power saving algorithm is applied (VFR, VSR, VFR and VSR, or default i.e. no algorithm applied), therefore altering the perceived video quality. The participants do not know which encoding algorithm has been applied; they are only told that there may be changes to the video quality. After each video, four questions were asked to gather information about the participant's perception of the video (description in section 4.1.2). Table 6 is an example of three participants who are randomly assigned to a specific video and presentation order of the applied power saving algorithms.

Table 6: Example of video content and presentation order of applied power saving algorithm for the first three participants. In all, 16 participants were obtained.

Participant	Video Content	First Video	Second Video	Third Video	Fourth Video
1		VFR and VSR	VSR	VFR	Default
2		VFR	Default	VSR	VFR and VSR
3		Default	VFR	VFR and VSR	VSR

Participants were not compensated for their time, so we were concerned that they would become uninterested in taking a long survey and stop mid-way through. By dividing the survey into two parts, participants could complete parts 1 and 2 separately and on their own schedule.

4.2.1 User Study: Part 1

Part 1 of the user study consists of background questions of the participant

- What is your age?
- What is your gender?
- Do you speak ASL?
- If applicable, how many years have you spoken ASL?
- If applicable, from whom did you learn ASL?
- What language do you prefer to communicate with family?
- What language do you prefer to communicate with friends?
- Are you Deaf?
- Do you use computer instant messenger services like Skype, G-mail chat, etc.?
- Do you use a video phone?
- Do you use video relay services?

At the end of part 1, the participant is given a unique PIN code and web-site link to access part 2. The PIN code will allow access to part 2 only once.

4.2.2 User Study: Part 2

Part 2 begins with a reminder to the participant that they must install the current version of QuickTime in order to view the videos presented in the survey. Next, users are prompted to enter their unique PIN code to access part 2. Restricting access to part 2 of the survey controls who views the content on the Internet. After entering the PIN, the participant sees the instructions page. Figure 19 is a picture of the instructions page.

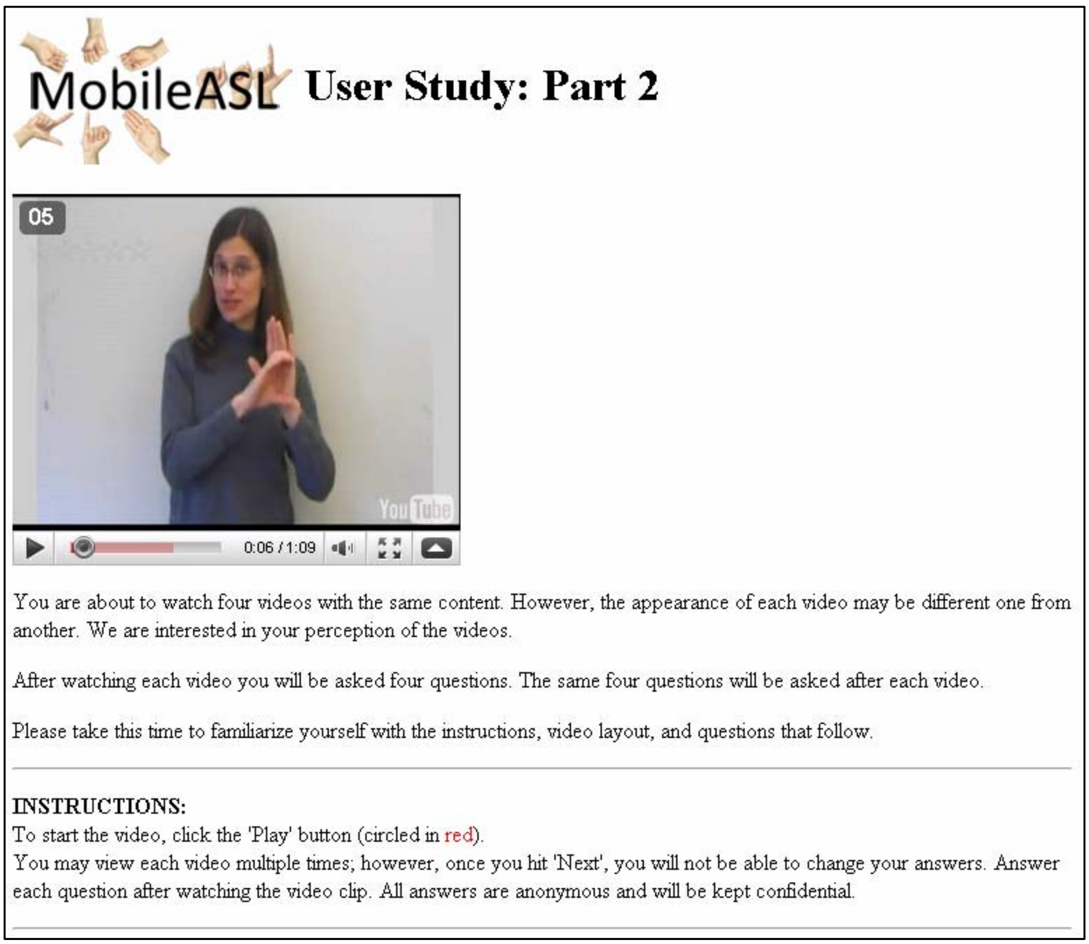


Figure 19: Part 2 instructions page with ASL instructional video.

The remainder of part 2 consists of a 35-second video shown four times, where each time, one of the power saving algorithms has been applied (VFR, VSR, VFR and VSR, default). The content of the video is a one sided conversation with an equal amount of signing and not signing sections. After each video, four questions are asked to understand the users' perception of the video. The four questions are:

1) I notice portions of this video were choppy.
2) The choppy portions of the video are distracting.
3) I notice portions of this video are blurry.
4) The blurry portions of the video are distracting.

The same four questions are asked after each video. The layout of the video and questions is shown in Figure 20.

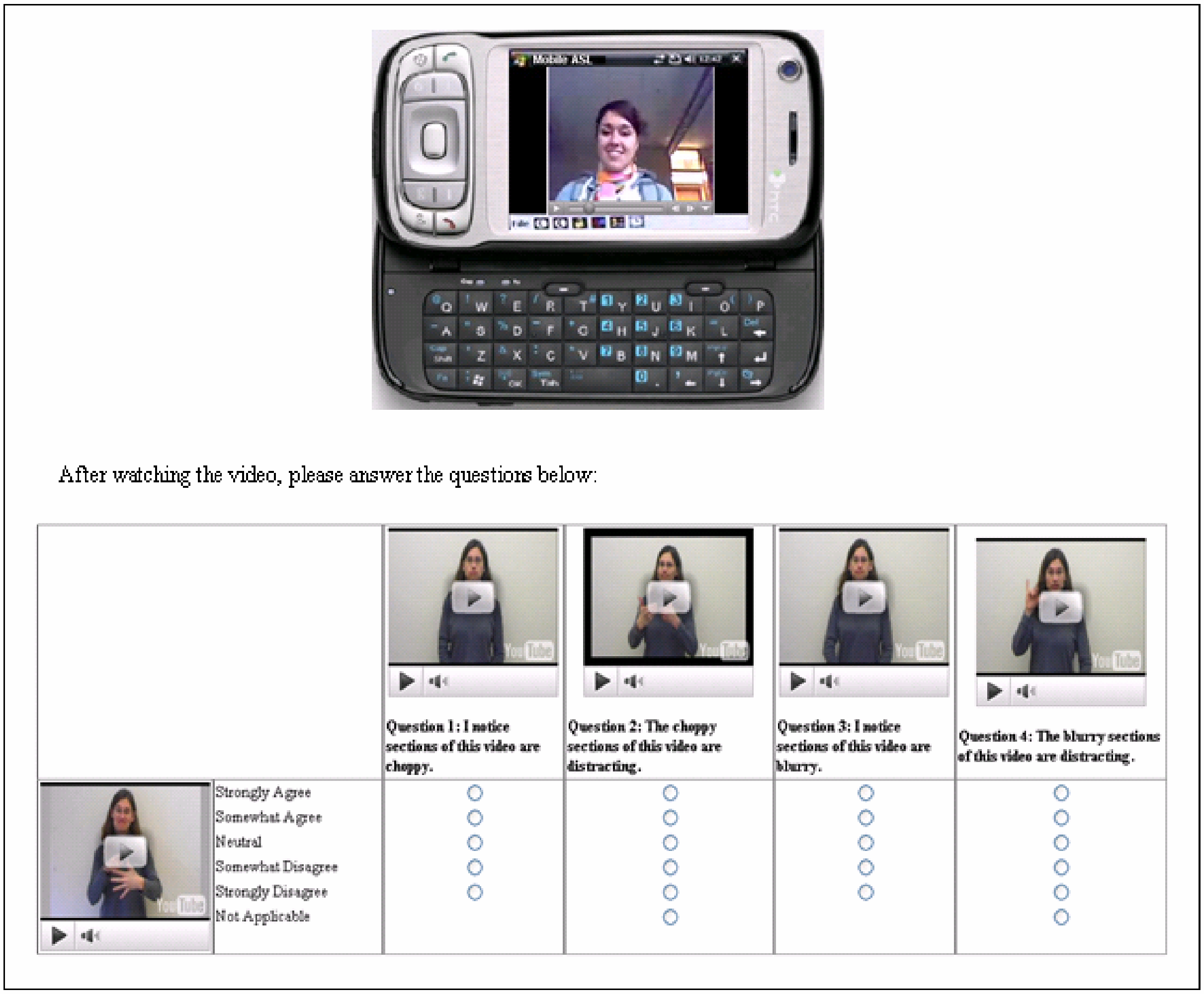


Figure 20: Part 2 video and question layout.

As Figure 20 demonstrates, there are four ASL videos interpreting each question and one instructional video interpreting the five-point Likert scale. The degrees of the five-point Likert scale are: strongly agree, somewhat agree, neutral, somewhat disagree, strongly disagree, and for questions 2 and 4, we provide a 'not applicable' option. The not applicable option was made available for participants if they feel those questions do not pertain to the video viewed. For instance, a participant cannot agree or disagree that the perceived choppiness or blurriness of a video was found to be distracting if choppiness or blurriness were not noticed.

4.3 Survey Methodology

A web-based study was selected over a laboratory study because a laboratory study limits participation to 15-20 participants from in and around the Seattle area. With a web-based user study, anyone who has access to a computer connected to the Internet can participate. The video relay service company, Sorenson, will be sending our survey to their subscribers; therefore we anticipated reaching 10,000 participants nationwide during phase three (see section 4.3) of the deployment of the user study.

The major influence in the creation of the format of the survey was considering how our intended audience (members of the Deaf and Hard of Hearing community) interprets the questions asked. Since English and ASL are not a direct translation, signed instructional videos were included throughout the survey to make sure all participants would be able to understand the instructions and questions asked.

A five-point Likert scale, ranging from strongly agree to strongly disagree accompanied each question to measure the participants perception of video quality. During the early stages of the survey a horizontal Likert scale was used, as shown in Table 7.

Table 7: Horizontal five-point Likert scale.

Questions:	Strongly Disagree	Disagree	Neutral	Agree	Strongly Agree	Not Applicable
I notice sections of this video are choppy.	•	•	•	•	•	
The choppy sections of this video are distracting.	•	•	•	•	•	•
I notice sections of this video are blurry.	•	•	•	•	•	
The blurry sections of this video are distracting.	•	•	•	•	•	•

The marketing director at Sorenson, which conducts user surveys to improve their services, was consulted during the evaluation of this early format. He pointed out that the horizontal Likert scale was too text heavy, so that native ASL speakers may not fully understand all of the written text. There were two suggestions for improving this format. The first suggestion was to simplify the survey questions so that there would be only yes or no responses. The disadvantage of this method was that we would not be able to draw in-depth conclusions from the results; we would only be able to conclude that a person noticed the video quality had changed. The second suggestion was to tailor the questions so that they could be answered using more specific responses such as:

- The video is very choppy.
- The video is moderately choppy.
- The video is slightly choppy.
- The video is not choppy at all.

Although this second method would provide more in-depth results, the concern with this method was that it would make the survey too long and text-heavy since participants have to view four videos and answer four questions per video. Allen, Meyers, Sullivan et al. indicated that vertical Likert scales are more preferable for ASL speakers over horizontal because everyone has a sense of what is “up” and “down” [8, 26]. Therefore, the compromise was to create a vertical Likert scale and to include an ASL video for each question, therefore making the Likert scale bilingual.

4.3.1 Video Content

Videos shown on a computer screen may appear different due to screen resolution, color mapping, and the decoder used by the media player. These aspects were considered when creating the videos presented in the survey. To accurately represent mobile phone video on the computer screen, the videos were recorded with the video camera on the cell phone. We recorded three individual conversations of two local Deaf women and a man signing at their own natural signing pace. The content of their conversations include asking every day questions such as how are they doing and what did they do on the weekend. The recorded conversations were then encoded using H.264, which is the same encoder used to transmit video when running MobileASL. Afterwards, the encoded video was converted to MPEG 4 using a publicly available converter [23] that does not add additional artifacts during conversion. It was essential that additional artifacts were not added to the encoded video because it would interfere with perceived quality of the video. These videos were stored on our local servers and were downloaded to a participant's local computer when viewed. The Apple QuickTime media player [4] was used in the web-based survey to play the videos on the computer screen.

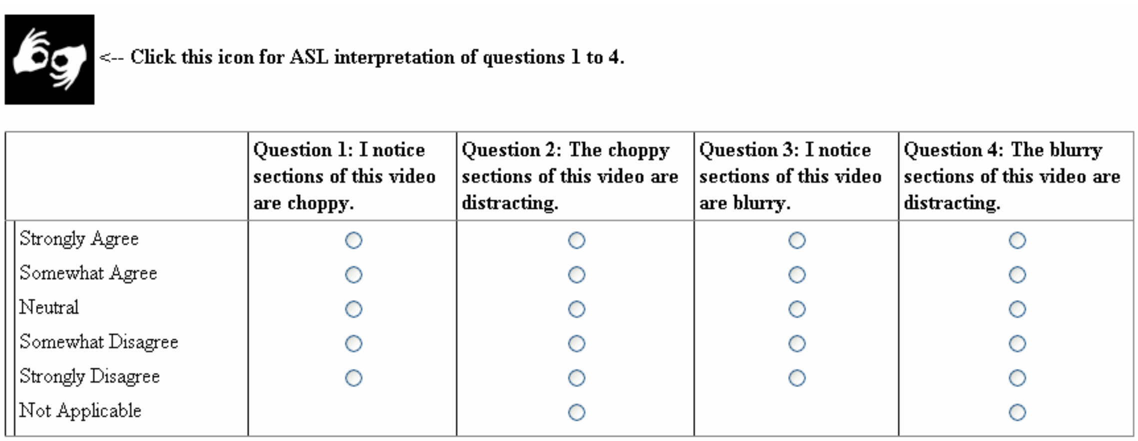
The ASL instructional videos were recorded using a standard video camera. During the early stages of developing the survey, we did consider loading our survey videos onto YouTube [3] for easy accessibility and to take advantage of their streaming capabilities and YouTube's infrastructure. However, YouTube uses the Flash encoder when compressing uploaded video, which introduces artifacts that would interfere with the perception of the applied power saving algorithms. For the instructional videos, the additional artifacts are negligible and do not affect the content of the videos.

4.4 Three Stage Deployment Process

A three phase approach was used in the development of the user study. Phase III is currently a work in progress and discussed in the chapter 5.

Phase I

Phase I of the user study involved creating the questions asked in part 1 and 2 of the survey and recording the survey and instructional videos used in the survey (as described in sections 4.1 and 4.2). Once the website was created for this survey, it was sent to ten local participants within the University of Washington to uncover any glitches in the web programming and determine the capacity of our servers when logging the responses to prevent our website from crashing with thousands of people accessing the survey at one time. During phase I we discovered that some participants thought that the ASL video interpretations of the questions were the actual videos to which the questions were referring. To eliminate future confusion as to which video the questions were referring, we made a button to allow the participant to switch between English text and the ASL interpretation of the questions. Figure 21 demonstrates the English representation of the questions and Figure 22 demonstrates the ASL video interpretation of the questions.



<-- Click this icon for ASL interpretation of questions 1 to 4.

	Question 1: I notice sections of this video are choppy.	Question 2: The choppy sections of this video are distracting.	Question 3: I notice sections of this video are blurry.	Question 4: The blurry sections of this video are distracting.
Strongly Agree	○	○	○	○
Somewhat Agree	○	○	○	○
Neutral	○	○	○	○
Somewhat Disagree	○	○	○	○
Strongly Disagree	○	○	○	○
Not Applicable		○		○

Figure 21: English representation of questions.

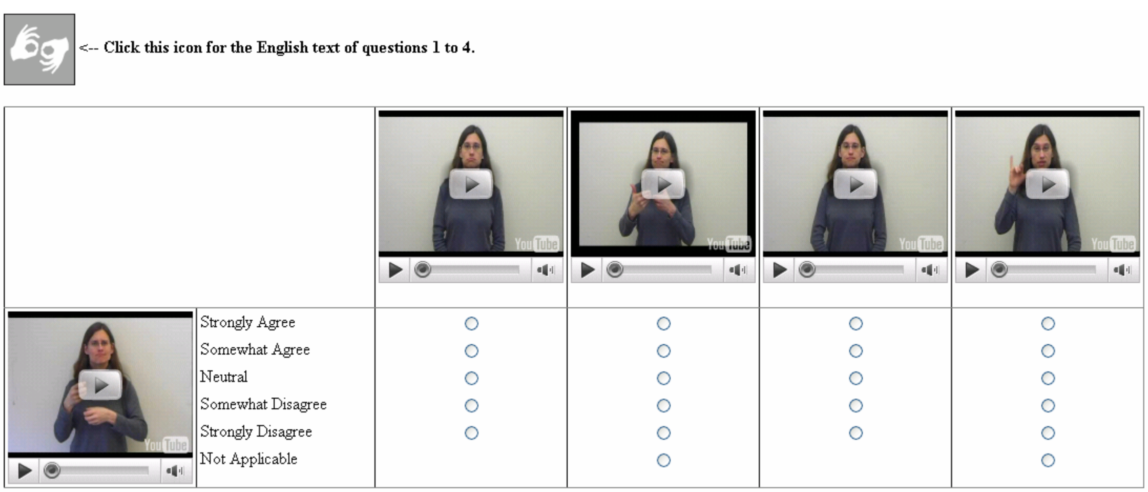


Figure 22: ASL video interpretation of the questions.

By only showing either the English text or ASL video interpretation of the question at one time, potential confusion as to which video these questions are referring will be eliminated.

Phase II

In phase II of the user study, we invited 125 people who had e-mailed Professor Richard Ladner inquiring about MobileASL to become participants. I also used the social networking site, Facebook, for a call for participants. I made a MobileASL University of Washington fan page where people could join to learn more about MobileASL. I also posted our survey to different Facebook groups with interests in ASL to increase participation in the survey. The duration of phase II lasted one week. (Results are presented in section 4.5.)

I am interested in determining the main effects and interactions of the factors for questions 1-4; therefore statistical analysis was conducted on the data. An F-test is not appropriate since the data are not normally distributed, is ordinal in nature, and is bounded by the scale endpoints. Also the rank transform (RT) method is not suitable because it is unreliable for testing interactions, which is a component that I want to see from my experiment (i.e., the interaction between VFR and VSR). Therefore the aligned rank transform (ART) [31] procedure was performed on the data before performing a repeated measures ANOVA on the aligned ranks. This is therefore a nonparametric analysis despite it ultimately relying on an F-test. The ART is applied by first removing from the means for all effects *except* the effect of interest, whether it is a main effect or an interaction, and then the aligned data is ranked, and the results are determined from conducting a repeated measures ANOVA (see section 4.5 for results).

Phase III

The results gathered from phase II prompted changes to the survey layout, as described in chapter 5. In this phase, we are planning to utilize Sorenson's contact list to invite more participants to take part in the user study.

4.5 Phase II Study Results

In this user study, I am interested in the effects of VFR and VSR on video quality perception. In phase II, there were 16 participants fluent in ASL (eight women and eight men). Their age ranged from 19-65 years old and all but one participant is deaf. Fourteen of the 16 participants indicated that they own a cell phone and use it to text message. Finally, all the participants indicated that they use video phones and video relay services.

Question 1

Question 1 asked participants if they notice choppy sections of video when VFR and VSR algorithms are applied. Figure 23 is a histogram plot of the responses for the four applications.

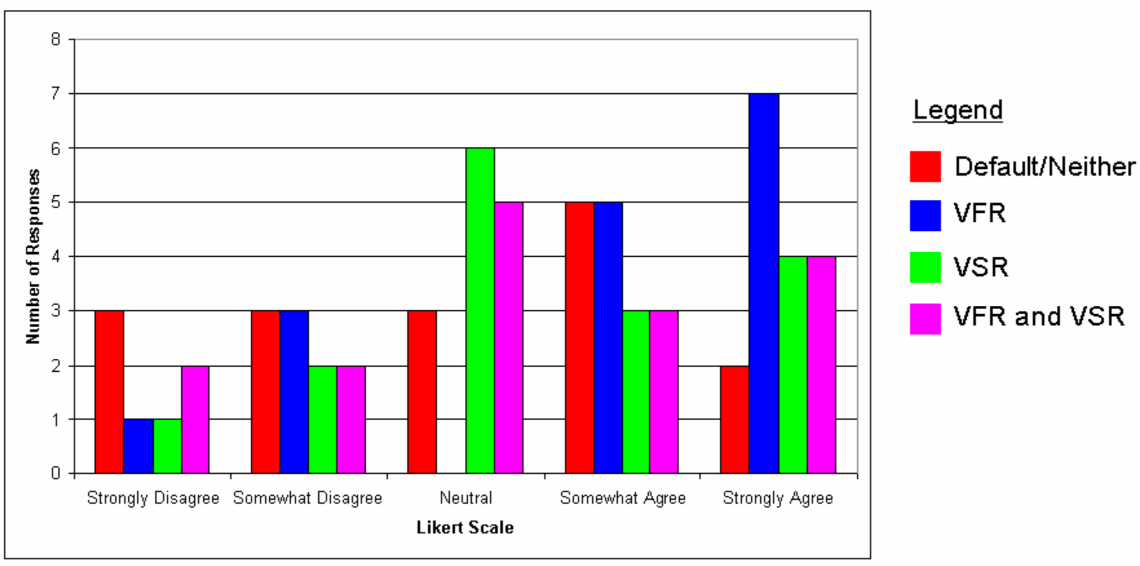


Figure 23: Histogram plot for question 1: "Sections of this video are choppy."

The statistical analysis of the data from question 1 revealed that the participants' perceived choppiness of the video was not detectably affected by the application of either VFR ($F_{1,15}$=1.04, n.s.) or VSR ($F_{1,15}$=0.11, n.s.). However, when both VFR and VSR were applied to the video, there was a significant VFR*VSR interaction for perceived choppiness ($F_{1,15}$=5.3, p<.05). Table 8 reveals this finding.

Table 8: Mean responses for question 1.

		VSR (factor 2)	
		OFF	ON
VFR (factor 1)	OFF	3.0	3.8
	ON	3.8	2.9

As Table 8 demonstrates, when VFR is on and VSR is off, the average response is 3.8, but when both VFR and VSR are on, the average response is reduced to 2.9. (Recall, a 5-point Likert scale was used, where 5 corresponds to strongly agree and 1 corresponds to strongly disagree.) This reduction indicates that the application of VSR reduces the perceived choppiness introduced by VFR.

From the histogram plot for question 1, it would appear that the application of VFR would cause participants to notice choppy sections of video. However, when looking at the mean values for the VFR main effect, as Figure 24 demonstrates, the difference in the means when VFR is off and on is not very large (3.2 vs 3.6, respectively).

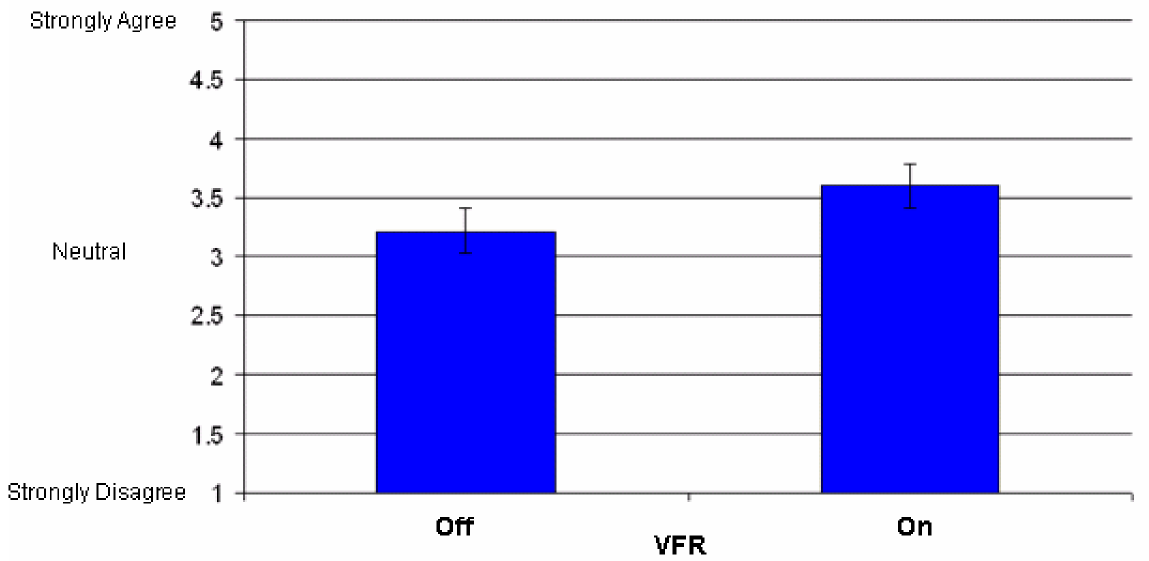


Figure 24: Mean responses for two levels of VFR for question 1.

There was no significant difference in participants' noticing of choppy sections of video when VFR was applied ($F_{1,15}$=1.04, n.s.), despite the histogram plot indicating 7 participants strongly agreed that they notice choppy sections of video.

Question 2

Question 2 asked participants if they found the choppy sections of video to be distracting when VFR and VSR algorithms are applied. Figure 25 is a histogram plot of the responses for the four applications.

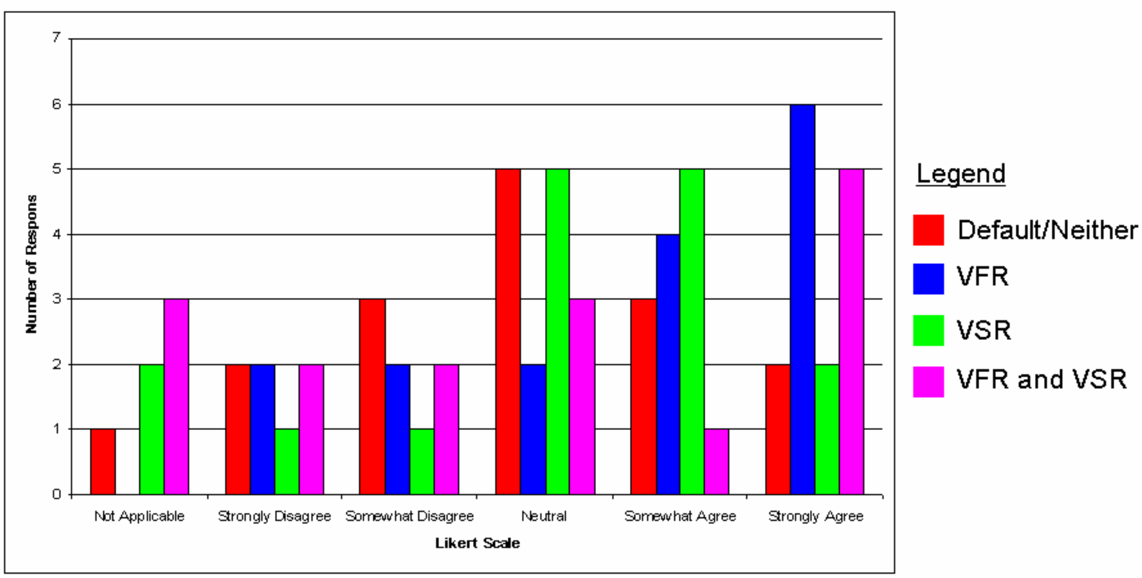


Figure 25: Histogram plot for question 2: “Choppy sections of video are distracting.”

Significance tests did not reveal that participants found choppy sections of video to be distracting for the application of either VFR ($F_{1,15}$=0.61, n.s.) or VSR ($F_{1,15}$=0.58, n.s.). Neither was there a significant VFR*VSR interaction ($F_{1,15}$=2.81, n.s.).

Question 3

Question 3 asked participants if they notice blurry sections of video to be distracting when VFR and VSR algorithms are applied. Figure 26 is a histogram plot of the responses for the four applications.

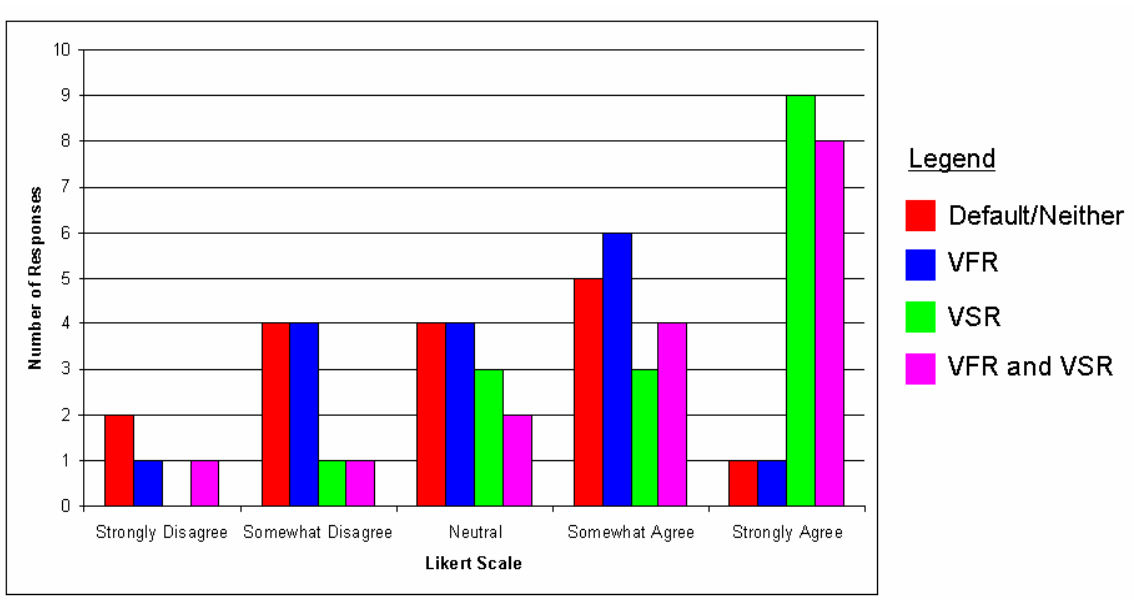


Figure 26: Histogram plot for question 3: "Sections of this video are blurry."

The statistical analysis of the data from question 3 revealed that the participants' perceived blurriness of the video was not affected by the application of VFR ($F_{1,15}$=0.03, n.s.); however it was statistically significant that participants perceived the application of VSR to cause blurriness ($F_{1,15}$=21.2, $p<.003$). There was no significant VFR*VSR interaction ($F_{1,15}$=0.45, n.s.). Figure 27 shows the mean response to the two levels of applying VSR.

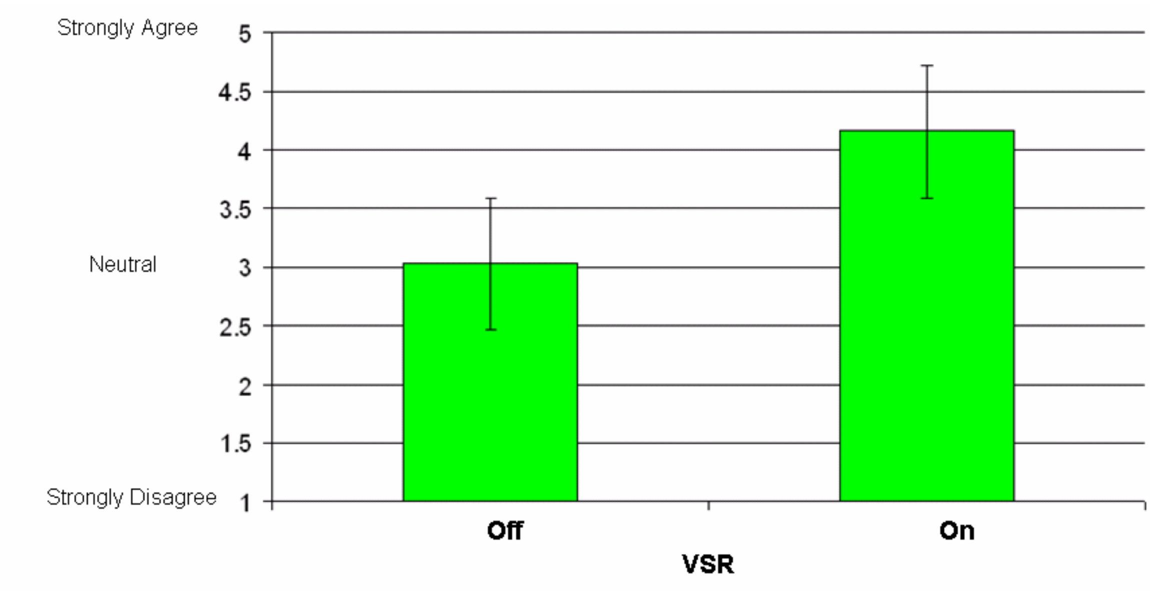


Figure 27: Mean responses for the two levels of VSR for question 3.

As Figure 27 demonstrates, the difference in the means when VSR is off and on was found to be statistically significant (3.0 vs. 4.2, respectively).

Question 4

Question 4 asked participants if they found blurry sections of video distracting when VFR and VSR algorithms are applied. Figure 28 is a histogram plot of the responses for the four applications.

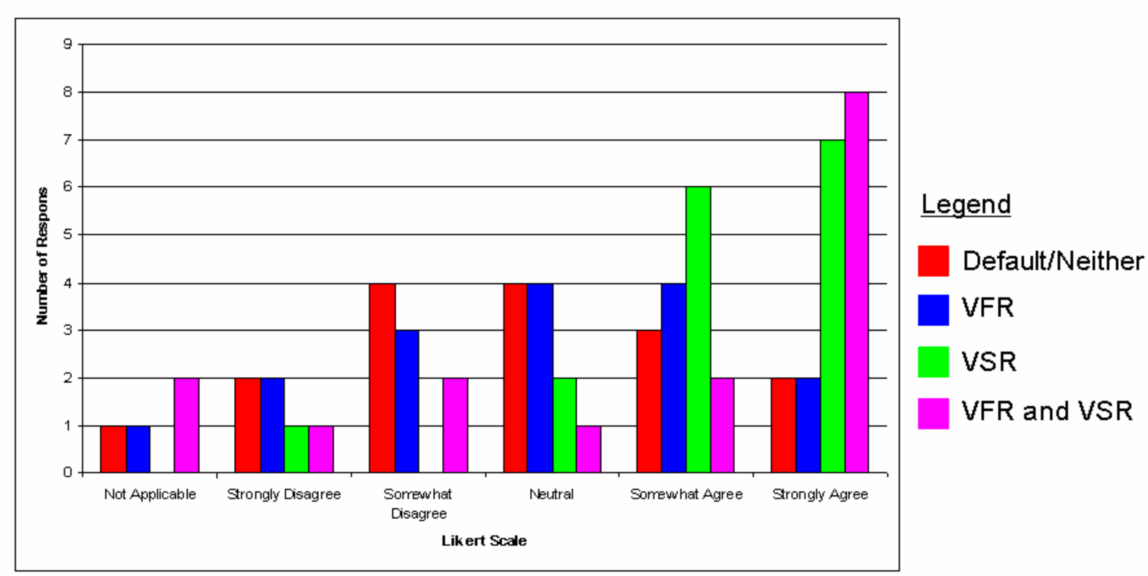


Figure 28: Histogram plot for question 4: "Blurry sections of video are distracting."

Not only did participants perceive blurry video with the application of VSR, but they also found the blurry video to be distracting ($F_{1,15} = 10.1$, $p<.01$). The application of VFR was not found to be distracting ($F_{1,15} = 0.07$, n.s.) nor the VFR*VSR interaction ($F_{1,15} = 0.59$, n.s.). The histogram for question 4 shows that participants found the blurry sections of video distracting. Figure 29 shows the mean response to the two levels of applying VSR for question 4.

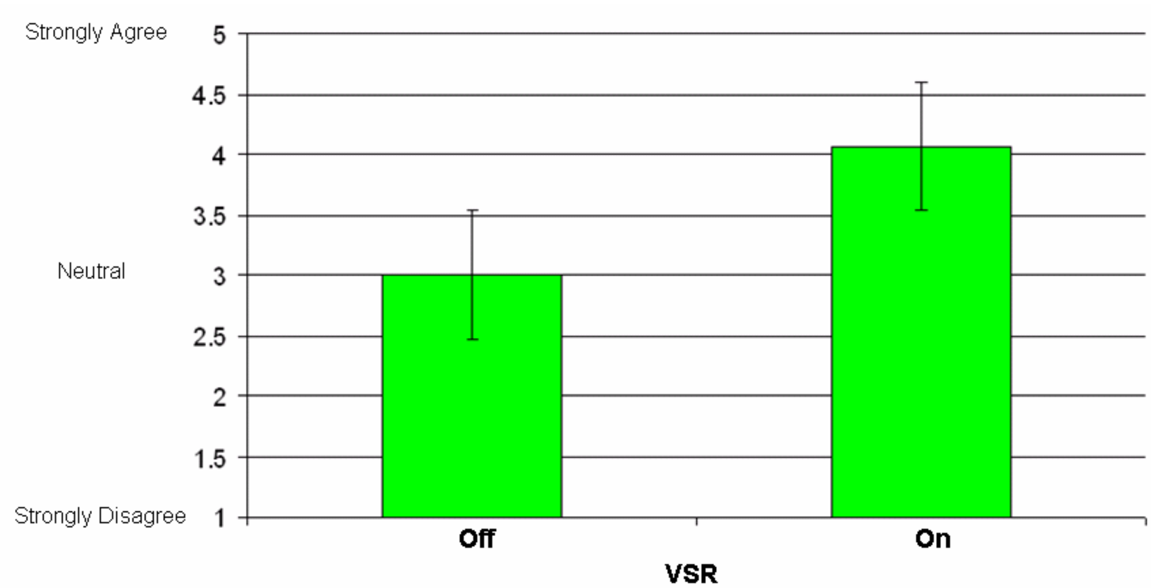


Figure 29: Mean responses for the two levels of VSR for question 4.

As Figure 29 demonstrates, the difference in the means when VSR is off and on is statistically significant (3.0 vs. 4.1, respectively).

As section 4.2 describes, participants were randomly assigned to view 1 of 3 videos where each algorithm was applied so that the participant viewed four versions of the same video. A Friedman test shows that there is no statistically significant main effect of presentation order (the order in which the algorithms were presented to the participant) on the Likert response to question 1 (χ^2=2.588, df = 3, N = 16, n.s.); question 2 (χ^2=2.690, df = 3, N = 16, n.s.); question 3 (χ^2=.233, df = 3, N = 16, n.s.); and question 4 (χ^2=.103, df = 3, N = 16, n.s.). Application of Friedman's test also shows that the person signing in the videos had no statistically significant main effect on the Likert response to question 1 (X^2=.125, df = 2, N = 16, n.s.); question 2 (χ^2=.125, df = 2, N = 16, n.s.); question 3 (χ^2=.125, df = 2, N = 16, n.s.); and question 4 (χ^2=.400, df = 2, N = 16, n.s.).

4.6 Discussion

Only 12.8% of the invited phase II participants completed the entire two part web survey. 34.4% of the invited participants completed part 1 of the survey, but many participants did not continue on to part 2. A possibility for participants not continuing on to part 2 may be that participants may have had a low comprehension of English and therefore did not realize that they had only completed half of the survey. Also, some participants that did continue on to part 2 did not press the ‘play’ button to view the videos in the survey; their responses were not used in the analysis. Because of these issues, we have decided to change the layout of the survey to address these observations before continuing onto phase III.

In chapter 3, we found that the implementation of both VFR and VSR extended the battery life by 30 minutes, which is longer than the extension by VFR and VSR alone. Since the VFR*VSR interaction was found to reduce the perceived choppiness of video, this motivates the implementation of VFR and VSR as the preferred power saving implementation for MobileASL.

CHAPTER 5: FUTURE WORK AND CONCLUSION

5.1 Future Work

5.1.1 Activity recognition classification

We would like to investigate if activity recognition can be improved using the VSR implementation. Currently, the sum of pixel differences between frames is used for classification of signing and not signing frames. Since the implementation of VSR transmits frames that are smaller than the current implementation of MobileASL, we would like to investigate if the smaller frame size and having the full frame rate used with block motion compensation could improve classification.

5.1.2 Phase III User Study

We would like to improve the layout of the user study before beginning phase III of the web-based user study. During phase II, many participants only completed part 1 of the survey which asked background information. The responses that we are more interested in are those where participants watched four videos and responded to questions asked (see chapter 4). The new layout will combine parts 1 and 2 into one survey so participants will not need a PIN code and will not fail to complete the entire survey. We intend to present the content from part 2 at the beginning of the survey and then conclude by asking background questions.

Sorenson has agreed to send our survey to their contacts; therefore we anticipate more participants to complete the survey during phase III. I intend to submit a paper which includes the findings from phase III to the 12th International ACM SIGACCESS Conference on Computers and Accessibility (ASSETS 2010).

5.1.3 Intelligibility of ASL finger spelling and repeated requests

Activity recognition may misclassify signing frames for not signing frames during ASL finger spelling due to low spatial movement. Future work will investigate the intelligibility of ASL finger spelling during the applications of VFR, VSR, and both VFR and VSR. Since the web-based user study used pre-recorded video and asked for the participants' perception of the video quality, we may find different results if users use the different power saving algorithms in real time. A laboratory study would provide instant feedback on the intelligibility of ASL finger spelling and the need for repeated requests during a conversation using MobileASL.

5.2 Conclusion

My research investigated two alternative power saving strategies to extend the battery life of a HTC TyTN II cell phone running MobileASL. I successfully implemented each algorithm on the cell phone and conducted a power study to quantify battery savings. In the battery power study each experiment measured the battery life, current drain, and CPU usage of the cell phone. I discovered that running MobileASL without any power saving algorithm applied consumes 99.7% of the phone's CPU and a full battery charge lasts on average 284 minutes. Implementing VFR or VSR algorithms separately extended the battery life to an average 307 and 306 minutes respectively and lowers the CPU usage to 26% and 32% respectively. Applying VFR and VSR algorithms together extend the battery life to 315 minutes and lower the CPU usage to 10%.

In a preliminary web-based user study, I was interested if participants could perceive changes in video quality, especially during the not signing portions of a conversation and if so, did they find the changes distracting. The experimental design of the user study was a 2 x 2 within-subjects factorial design. The two factors (independent variables) are 1) VFR algorithm and 2) VSR algorithm, and the two levels are applying or not applying those factors (i.e., *on* and *off*). It was not detected that the main effect of applying VFR or VSR caused participants to perceive choppy video quality ($F_{1,15}$=1.04, n.s.) and ($F_{1,15}$=0.11, n.s) respectively, nor did participants find the choppy video quality to be distracting ($F_{1,15}$=0.61, n.s.) and ($F_{1,15}$=0.58, n.s.) respectively. However, there was a significant VFR*VSR interaction, ($F_{1,15}$=5.3, p<.05), which led to the discovery that applying VSR reduces the extent to which participants perceive VFR to induce choppiness. Another significant finding was that the application of VSR did cause participants to perceive blurry video quality,($F_{1,15}$=21.2, p<.003), and participants found the blurriness to be distracting, ($F_{1,15}$=10.1, p<.01).

The battery power study revealed that applying VFR, VSR, and both VFR and VSR all extend the battery life of a cell phone running MobileASL. Applying both VFR and VSR was found to extend the battery life the most. In the user study, it was discovered that VSR reduces the perceived choppiness introduced by VFR. Therefore, my recommendation is for MobileASL to adopt the use of both VFR and VSR algorithms as the power saving algorithm to extend the battery duration of the cell phone.

After conducting the preliminary user study, the most common feedback question received was “when will MobileASL be available?” This demonstrates that there is a need for real-time mobile sign language communication and that my thesis contributes to improving MobileASL technology for mainstream use.

BIBLIOGRAPHY

[1] What is SVRS?. (2010). Retrieved 1/22/2010, 2010, from http://sorensonvrs.com/what/index.php

[2] MobileASL. from http://www.cs.washington.edu/research/MobileASL/

[3] YouTube. (2010). Retrieved 1/20/2010, 2010

[4] QuickTime7. (2010). from http://www.apple.com/quicktime/download/

[5] acbPocketSoft. (2009). acbTaskMan Version. Retrieved 08/01/2009, 2009, from http://www.acbpocketsoft.com/Products/acbTaskMan/acbTaskMan-Overview-7.html

[6] Agrafiotis, D., Canagarajah, C. N., Bull, D. R., Dye, M., Twyford, H., Kyle, J., et al. (2003). Optimized sign language video coding based on eye-tracking analysis. VCIP, 1244-1252.

[7] Aimar, L., Merritt, L., Petit, E., Chen, M., Clay, J., Rullgrd, M., et al. (2005). x264 - a free h264/AVC encoder.

[8] Allen, B., Meyers, N., Sullivan, J., & Sullivan, M. (2002). American Sign Language and End-Of-Life Care: Research in the Deaf Community. HEC Forum, 14(3), 197-208.

[9] Berke, J. (2009). Robert Weitbrecht - Inventor of the TTY. Retrieved 1/22/2009, 2009, from http://deafness.about.com/od/peopleindeafhistory/a/weitbrecht.htm

[10] Buchmann, I. (2003). How to prolong lithium-based batteries.

[11] Cherniavsky, N., Cavender, A. C., Ladner, R. E., & Riskin, E. A. (2007). Variable Frame Rate for Low Power Mobile Sign Language Communication. Paper presented at the ASSETS', Tempe, Arizona, USA.

[12] Cherniavsky, N. (2008). Activity Analysis of Sign Language Video for Mobile Telecommunication. University of Washington, Seattle.

[13] Cherniavsky, N., Chon, J., Wobbrock, J. O., Ladner, R. E., & Riskin, E. A. (2009). Activity Analysis Enabling Real-Time Video Communication on Mobile Phones for Deaf Users. Paper presented at the ACM Symposium on User Interface Software and Technology (UIST), Victoria, British Columbia.

[14] Coates, J., & Sutton-Spence, R. (2001). Turn-Taking Patterns in Deaf Conversation. Journal of Social Linguistics, 507-529.

[15] Collins, S., & Petronio, K. (1998). What Happens In Tactile ASL? In C. Lucas (Ed.), Pinky Extension and Eye Gaze: Language Use in Deaf Communities (pp. 18-37): Galludet University Press.

[16] Communications, S. (2009, 12/19/2009). American Sign Language Grammar. Retrieved 1/20/2010, 2010, from http://en.wikipedia.org/wiki/American_Sign_Language_grammar

[17] Company, N. (2008). Texting More Popular Than Voice Calls in the USA. Retrieved 1/26/2010, 2010, from http://www.cellular-news.com/story/33824.php

[18] Crk, I., Albinali, F., Gniady, C., & Hartman, J. (2009). Understanding Energy Consumption of Sensor Enabled Applications on Mobile Phones. Paper presented at the Annual International IEEE EMBS Conference.

[19] Humphries, T., Padden, C., & O'Rourke, T. J. (1994). A Basic Course in American Sign Language. Silver Spring, MD: T.J. Publishers.

[20] James, C. L., & Reischel, K. M. (2001). Text input for mobile devices: comparing model prediction to actual performance. Paper presented at the CHI '01: Proceedings of the SIGCHI conference on Human factors in computing systems.

[21] Keating, E., & Mirus, G. (2003). American sign language in virtual space: Interactions between deaf users of computer-mediated video communication and the impact of technology on language practices. Language in Society, 32, 693-714.

[22] Klima, E. S., & Bellugi, U. (1979). The Signs of Language. Cambridge, MA: Harvard University Press.

[23] Kurtnoise. (2009). Yet Another MP4 Box User Interface for Windows Users, from http://yamb.unite-video.com/index.html

[24] Lee, S., Pattichis, M. S., & Bovik, A. C. (2001). Foveated video compression with optimal rate control. IEEE Transactions on Image Processing, 977-992.

[25] Merritt, L., & Vanam, R. (2007). Improved rate control and motion estimation for H.264 encoder. Proceedings of ICIP, 5, 309-312.

[26] Meyers, N. (2009). National Institute for Literacy.

[27] Muir, L., Richardson, I., & Leaper, S. (2003). Gaze tracking and its application to video coding for sign language. Paper presented at the Picture Coding Symposium.

[28] Muir, L. J., & Richardson, I. E. G. (2005). Perception of Sign Language and Its Application to Visual Communications for Deaf People. Journal of Deaf Studies and Deaf Education Advance Access, 390-401.

[29] Newstream/Arraycomm. (2003). Martin Cooper: History of Cell Phone. Retrieved 1/27/2010, from http://inventors.about.com/cs/inventorsalphabet/a/martin_cooper.htm

[30] Ong, S. C. W., & Caird, J. K. (1996). The effect of frame rate and video information redundancy on the perceptual learning of American Sign Language gestures. Paper presented at the CHI'96: Conference companion on Human factors in computing systems, NewYork, NY.

[31] Richter, S. J. (1999). Nearly exact tests in factorial experiments using the aligned rank transform. Journal of Applied Statistics, 26(2), 203-217.

[32] Sperling, G., Landy, M., Cohen, Y., & Pavel, M. (1986). Intelligible encoding of ASL image sequences at extremely low information rates: Academic Press Professional, Inc.

[33] World, I. T.The World in 2009: ICT Facts and Figures. Retrieved 10/24/2009, 2009, from http://www.itu.int/ITU-D/ict/material/Telecom09_flyer.pdf

[34] AT&T 3G map from http://www.xti9.com/att/att3gfull.gif