A COMPUTATIONAL MODEL OF THE INTELLIGIBILITY OF AMERICAN SIGN LANGUAGE VIDEO AND VIDEO CODING APPLICATIONS

A Dissertation

Presented to the Faculty of the Graduate School

of Cornell University

in Partial Fulfillment of the Requirements for the Degree of

Doctor of Philosophy

by

Francis Michael Ciaramello

May 2011

A COMPUTATIONAL MODEL OF THE INTELLIGIBILITY OF AMERICAN SIGN LANGUAGE VIDEO AND VIDEO CODING APPLICATIONS

Francis Michael Ciaramello, Ph.D.

Cornell University 2011

Real-time, two-way transmission of American Sign Language (ASL) video over cellular networks provides natural communication among members of the Deaf community. Bandwidth restrictions on cellular networks and limited computational power on cellular devices necessitate the use of advanced video coding techniques designed explicitly for ASL video. As a communication tool, compressed ASL video must be evaluated according to the intelligibility of the conversation, not according to conventional definitions of video quality. The intelligibility evaluation can either be performed using human subjects participating in perceptual experiments or using computational models suitable for ASL video. This dissertation addresses each of these issues in turn, presenting a computational model of the intelligibility of ASL video, which is demonstrated to be accurate with respect to true intelligibility ratings as provided by human subjects. The computational model affords the development of video compression techniques that are optimized for ASL video.

Guided by linguistic principles and human perception of ASL, this dissertation presents a full-reference computational model of intelligibility for ASL (CIM-ASL) that is suitable for evaluating compressed ASL video. The CIM-ASL measures distortions only in regions relevant for ASL communication, using spatial and temporal pooling mechanisms that vary the contribution of distortions according to their relative impact on the intelligibility of the compressed video. The model is trained and evaluated using ground truth experimental data, collected in three separate perceptual studies. The CIM-ASL provides accurate estimates of subjective intelligibility and demonstrates statistically significant improvements over computational models traditionally used to estimate video quality.

The CIM-ASL is incorporated into an H.264/AVC compliant video coding framework, creating a closed-loop encoding system optimized explicitly for ASL intelligibility. This intelligibility optimized coder achieves bitrate reductions between 10% and 42% without reducing intelligibility, when compared to a general purpose H.264/AVC encoder. The intelligibility optimized encoder is refined by introducing reduced complexity encoding modes, which yield a 16% improvement in encoding speed.

The purpose of the intelligibility optimized encoder is to generate video that is suitable for real-time ASL communication. Ultimately, the preferences of ASL users determine the success of the intelligibility optimized coder. User preferences are explicitly evaluated in a perceptual experiment in which ASL users select between the intelligibility optimized coder and a general purpose video coder. The results of this experiment demonstrate that the preferences vary depending on the demographics of the participants and that a significant proportion of users prefer the intelligibility optimized coder.

BIOGRAPHICAL SKETCH

Francis (Frank) Michael Ciaramello was born in a car parked haphazardly somewhere along PA Route 320 en route to the hospital in the suburbs of Philadelphia on April 16, 1982. In May 2000, Frank graduated as the (peer elected) salutatorian from Brophy College Preparatory in Phoenix, AZ, where he learned the value of being a “man for others”. Frank continued his education at Arizona State University, where he learned the importance of a good study group and a well-crafted beverage. He received the B.S.E. degree (summa cum laude) in electrical engineering from ASU in 2004. He joined the School of Electrical and Computer Engineering at Cornell University in August 2004. Frank remained at Cornell for 7 years, having truly experienced the joys of Ithaca and Cornell and the freedom of graduate student life. He received the Ph.D. degree in electrical engineering from Cornell University in May 2011.

To Nora and Noelle, may they discover a passion for their own endeavors.

Ad maiorem Dei gloriam

ACKNOWLEDGEMENTS

The completion of this dissertation would not have been possible without the support, guidance, encouragement, and time of many individuals. First, I would like to thank my advisor, Professor Sheila Hemami. This dissertation exists in no small part because of her patience during the course of my research and her enthusiasm for my work. In addition to (and likely more important than) the research demonstrated here, her insightful and constructive feedback on my papers and presentations have developed my ability to communicate highly technical ideas to a broad audience. I am also very grateful to Professor Hemami for the opportunities to teach, to attend conferences, to share a pig, and to live abroad; the value of these experiences cannot be exaggerated.

I am thankful for my academic collaborators. My committee members Professor C. Rick Johnson and Professor Geri Gay offered valuable support throughout my research. I'm particularly grateful for the many conversations in Rick's office conducted under the watchful eye of a clay soldier. Professor Patrick Le Callet, as well as the many members of the Image and Video Communication group (IVC) at l'Université de Nantes, broadened my perspective on image and video quality research and broadened my cultural experiences. I was incredibly fortunate to work with Dr. Amy Reibman as an intern at AT&T Labs Research. I am inspired by Amy's constant desire to learn and to evolve as a researcher. Her career advice and perspective on life, typically given to me over Thai, Indian, or Chinese food, were always appreciated.

Being in the Ph.D. program for 7 years has allowed me to interact with many incredibly talented, smart, and fun graduate students. While there are too many to list (and they all know who they are), my experience as a graduate student is defined by the relationships that I have developed and the time that I have shared with my peers. I would like to acknowledge my fellow members of the VCL. In particular, I appreciated the compliments and commentary provided by Dr. Matthew Gaubatz, who, despite my own doubts, considers my programming and debugging skills to be excellent. The many shared experiences with my academic sibling, Dr. David Rouse, cannot easily (or feasibly) be detailed here. It is hopefully sufficient to simply state that these experiences enhanced this work in indescribable ways.

I would like to recognize the support of my parents, who set the example for balancing a hard work ethic and a dedication to their family. How my mother completed a dissertation with four children while my father maintained his sanity will forever remain a mystery to me. My children, Nora and Noelle, fill my days with laughter and craziness and have taught me what is truly important in life.

Finally, I thank my loving wife Mary, without whom this work would not have been possible. Throughout our time together, Mary's confidence in me has never failed, even (and especially) when my own self-confidence wavered. Her constant encouragement drives me to success. I am truly blessed to be able to share my joys and sorrows, failures and successes, cookies and french fries, and my daily happiness with Mary.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

Real-time, two-way transmission of American Sign Language (ASL) video over cellular networks can provide natural communication among members of the Deaf [1] community [64]. For Deaf users of ASL in the United States, current communication technologies include video relay services, Internet-based video conferencing (e.g., Skype or Facetime), TTY/TTD services, and text-based communication. These technologies have significant limitations when compared to the potential capabilities of a mobile communication system for ASL using cellular phones. Video relay services and video conferencing each require a high-bandwidth Internet connection and dedicated hardware, severely limiting the user's mobility. Text messaging technologies, such as TTY/TTD and cellular SMS, can be unnatural and cumbersome for two reasons. First, these technologies rely on written English, which is not the first language of many ASL users. Second, communication via text messaging is significantly slower than with sign language, which has a communication rate comparable to that of spoken languages [34].

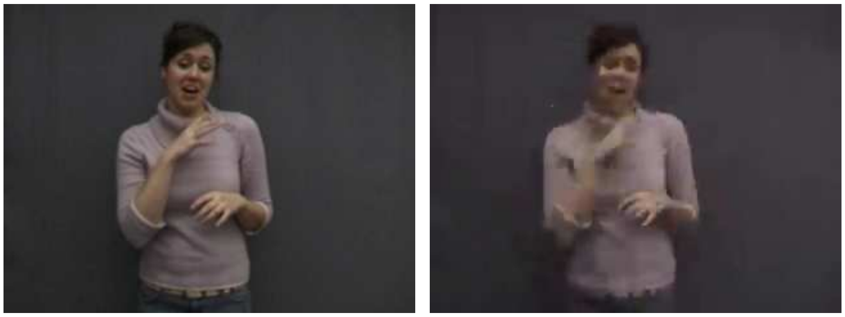
A cellular-based video calling system for ASL requires real-time capture, encoding, and decoding of digital video on a cellular device for transmission over the U.S. cellular network. Cellular devices have limited processing power and battery life, imposing constraints on the complexity of the encoding and decoding algorithms. The limited bandwidth of the U.S. cellular network constrains the data rate at which ASL video can be transmitted. The bandwidth available on GPRS networks is asymmetric and is limited to at most 40 kbps downlink and 20 kbps uplink [35]. Newer network technologies, such as 3G/4G, provide significantly in-

[1] Capitalized Deaf refers to people who are active in the signing Deaf Community and Deaf Culture.

creased data rates but are not ubiquitously available. Furthermore, regardless of the network technology, the true available bandwidth depends on several factors, including the user's distance from a cell tower and the number of concurrent users on the network. Delivering intelligible ASL video at extremely low data rates alleviates the geographical limitations and capacity restrictions imposed by the network, maximizing accessibility of the technology among the Deaf.

Modern video coding standards achieves excellent rate-distortion performance for generic video content [89]. However, the current state of the art encoders cannot reliably produce intelligible ASL video at the extremely low rates available on the cellular network [13]. As a motivating example, an ASL video is encoded using x264 (an open-source H.264/AVC coder) at 15 kilobits-per-second (kbps) and 15 frames-per-second (fps) is deemed unintelligible by fluent ASL users. Figure 1.1 illustrates a sample frame from the unintelligible video and highlights the severity of the distortions in the signer's face and hands at such a low bitrate. Clarity in these regions is critical for intelligible communication and is not sufficiently maintained by even a state of the art encoding algorithm, when constrained to worst-case cellular data rates.

Two-way video communication on cellular devices requires real-time video encoding and decoding. While these devices are becoming increasingly powerful, they still offer little computational power when compared to modern desktop computers. Furthermore, on a mobile device, any reductions in the computational complexity of the video encoder and decoder have a direct impact on the battery life of the system [18]. For mobile video communication to be useful, low complexity algorithms suitable for mobile devices must be employed.



(a) Original video frame.

(b) Sample frame taken from video encoded using x264 [2] at 15 kbps and 15 fps.

Figure 1.1: An advanced H.264/AVC encoder is inadequate for maintaining intelligibility in compressed ASL video at low bitrates. The severity of the distortions in the signer's face and hands result in unintelligible video, as rated by fluent ASL users [13].

Prior ASL-specific encoding systems primarily focus on addressing the bandwidth constraints. These systems exploit the inherent structure of ASL, increasing the compression efficiency over more general encoders while attempting to maintain the intelligibility of the ASL. Because ASL video is known to contain only a single signer, the ASL-specific encoding systems apply spatially-varying levels of compression to maintain clarity in the signer at the expense of clarity in the background [4, 54, 66, 67]. These systems appropriately consider the structure of ASL video, allowing for increased compression without affecting intelligibility, when compared to general encoders. However, each of these encoding systems relies on a set of heuristics to distribute rate between the signer and the background. Furthermore, these systems either make assumptions that intelligibility is unaffected by the encoder or perform subjective experiments to determine the intelligibility of the compressed video. The assumptions made by these systems will not be valid at every operating point and performing subjective experiments is prohibitively costly and difficult to incorporate into the design cycle of an encoding system.

This motivates the need for a computational model that can accurately predict the subjective intelligibility of compressed ASL video. Such a model provides a method for comparing encoding systems and selecting the one which provides, on average, the most intelligible video, without requiring subjective evaluation. The computational intelligibility model can also be used for system design or refinement; compression algorithms can be designed to maximize an objective intelligibility criteria, and consequently maximize subjective intelligibility, without the need for heuristic encoding techniques.

Contributions

In the absence of prior computational models for intelligibility, a straightforward approach is to apply computational techniques designed to measure video quality or video fidelity. The most common technique for evaluating the fidelity of compressed video is to measure the mean squared-error (MSE) or the peak signal-to-noise ratio (PSNR). Efforts in recent years have focused on developing models that more accurately predict human ratings of perceived quality, in terms of visual aesthetics or perceptual similarity to a source video [7, 55, 69, 87].

One contribution of this dissertation is to demonstrate that objective measures of video quality cannot reliably estimate the subjective intelligibility of compressed ASL video, as rated by fluent ASL users. ASL video is a communication tool and, as such, the video must be evaluated in terms of intelligibility. The techniques described in [7,55,69,87] are designed to predict how an observer will perceive visual distortions in a video and quantify the impact of the perceived distortion on the perceived quality. A fluent ASL user will ignore many of the visual distortions when watching a compressed sign language video; her ultimate goal is understanding.

This work presents a computational intelligibility model for ASL video (CIM-ASL), which is based on models of ASL perception. The CIM-ASL measures distortions only in regions relevant for ASL communication (i.e., the signer's face, hands, and torso), quantifying the impact of these distortions on both the spatial and temporal structure of ASL. The CIM-ASL accurately predicts subjective intelligibility on three separate experimental datasets.

In addition, the CIM-ASL is incorporated into an H.264/AVC compliant encoding algorithm in order to demonstrate the effectiveness of the model when used for generating intelligible ASL video. The H.264/AVC standard is selected for its state of the art compression efficiency. This intelligibility optimized encoder yields intelligibility equal to a general purpose video encoder with bitrate reductions of 10% to 40%. The CIM-ASL is also used to refine the encoding algorithm in the presence of a complexity, allowing for real-time operation on a cellular device.

The intelligibility optimized encoder achieves bitrate reductions by heavily distorting the background video region, while maximizing the fidelity of the signer. A subset of participants in a subjective experiment qualitatively reported distractions due to heavily distorted backgrounds, even when they considered the videos to be intelligible. Allowing the user to adjust the level of background distortion addresses this problem, but lowering the distortion in the background region necessarily increases the distortion in the signer and can lead to an unintelligible video.

The intelligibility optimized encoder is further refined to accommodate varying user preferences. A quality-intelligibility coder is developed, which is parameterized by a single, user controlled value. Depending on the user's preference, the quality-intelligibility coder jointly maximizes the CIM-ASL and a quality criteria, defined as MSE. A user study is conducted to identify the preferences of ASL users in terms of this quality versus intelligibility trade-off, specifically identifying when a user is willing to sacrifice intelligibility (as measured by the CIM-ASL) for an increase in video quality (as measured by MSE). The study demonstrates that the user preferences vary depending on the demographics of groups of users.

The contributions of this dissertation are summarized as follows.

- Evidence that computational models of video quality cannot reliably estimate the subjective intelligibility of compressed ASL video [20, 21].
- A full-reference computational model of the intelligibility of ASL video (CIM-ASL), which is based on models of ASL perception and is accurate in the presence of compression-type distortions [21].
- The design, implementation, and analysis of three subjective experiments that characterize and quantify the intelligibility of distorted ASL video [20, 24, 26].
- Applications of the CIM-ASL for operational rate-distortion-complexity optimization. The CIM-ASL is applied in a rate-distortion optimization algorithm, which provides a closed-loop solution for generating optimally coded ASL video [25]. Additionally, the CIM-ASL is applied to distortion-complexity optimization, which includes the development of novel complexity allocation techniques [28].
- A refined ASL encoder that accommodates user preferences [22] and a perceptual experiment that verifies the need for this accommodation and identifies groups of ASL users who have varying preferences [27].

Organization

This dissertation proposes an accurate computational intelligibility model for ASL (CIM-ASL) and applies the model to several video coding applications. The CIM-ASL is based on models of ASL perception, introduced in Chapter 2. Chapter 2 also describes the effect of video coding distortions on ASL communication, which informs the design of the CIM-ASL. Ground-truth subjective ratings of the intelligibility of distorted ASL video are collected via three experiments described in Chapter 3. The experimental data is used to parameterize and evaluate the CIM-ASL. Chapter 4 presents the CIM-ASL, describing the computational techniques for modeling intelligibility and the parameter optimization procedure. As demonstrated in Chapter 5, the performance of the CIM-ASL, and several computational video quality models, is determined by the statistical accuracy with which the model predicts the ground-truth intelligibility ratings.

Given an accurate computational model, the CIM-ASL is applied in two ways. Following an introduction to motion compensated video coding in Chapter 6, the CIM-ASL is used to create compressed ASL video having maximal intelligibility. The CIM-ASL is applied to perform joint rate-distortion-complexity optimization of an H.264/AVC video coder, as detailed in Chapter 7. Chapter 8 presents a modification to the optimal ASL coder that allows for varying user preferences and identifies groups of ASL users who have varying preferences.

CHAPTER 2

AMERICAN SIGN LANGUAGE COMMUNICATION AND DISRUPTIONS

2.1 Introduction

To accurately predict intelligibility, the CIM-ASL must incorporate models of ASL communication. Furthermore, the CIM-ASL must effectively quantify the impact of distortions on the communication process. Semantic information in ASL is communicated through the hand signs and facial expressions of the signer. The *phonology* of ASL defines the elementary units that combine to differentiate meaningful handshapes and movements from arbitrary gestures. An understanding of ASL phonology is essential for informing computational techniques that will assess intelligibility. The CIM-ASL must identify distortions that interfere with the phonological units in order to quantify the impact of video coding distortions on the subjective intelligibility. Section 2.2 reviews ASL phonology, which describes the formation of semantic information and the communication process from the *signer* to the 'listener', commonly denoted the *receiver*. An additional feature of the communication process from signer to receiver is the face-centric viewing patterns of the ASL receiver. Section 2.3 discusses this phenomenon and its impact on ASL communication.

Understanding the phonology of ASL is critical for the design of the CIM-ASL. However, the robustness of ASL communication in the presence of video coding distortions cannot be determined from the phonology alone. Two classes of subjective experiments, denoted *receiver-centric* and *encoder-centric*, provide insight into the impact of video coding distortions on ASL communication. Receiver-centric studies measure the receiver's performance in response to reductions to visual fidelity, rather than evaluating a specific encoding system. The distortions that are applied to videos in these receiver-centric studies are chosen *a priori* in order to characterize sign language perception in the presence of a specific type of distortion. In contrast, encoder-centric studies analyze the intelligibility provided by real systems and characterize the encoding system. Encoding systems designed explicitly for sign language video can lead to reductions in intelligibility caused by distortions that are not necessarily encountered and quantified by the receiver-centric studies. The encoder-centric methodology facilitates the analysis of the distortions that can occur in real systems, which must be properly accounted for by the CIM-ASL.

Following the discussion of ASL phonology in Section 2.2 and Section 2.3, a discussion of relevant receiver-centric perceptual experiments is provided in Section 2.4. Section 2.5 reviews prior ASL-specific video encoding systems and discusses their associated encoder-centric studies. Finally, Section 2.6 describes the types of distortions that occur in these systems under resource-constrained encoding scenarios.

2.2 Relevant Phonology in ASL Linguistics

The *phonology* of a language defines the smallest units that provide contrasting information. ASL signs combine five basic units to form meaningful gestures: handshape, location, orientation, movement, and nonmanual signals (e.g., facial expressions) [80]. Signs have both spatial structure, defined by the handshape, location, and orientation, and temporal structure, defined by the relevant movements. If any of these five basic elements differs between signs, the signs will carry different meaning. For example, the signs for *apple* and *onion* are identical in both their handshape, movement, and orientation and differ only in location. *Apple* occurs at the top of the cheek while *onion* occurs next to the eye. Many noun-verb pairs, such as *sit* and *chair*, differ only in their movements, having identical spatial configurations [76].

In addition to the basic hand movements associated with some signs, further temporal structure in ASL is described by the hold-movement-hold model [48]. In this model, signs are composed of a sequence of the five basic units and can be characterized by an initial articulation of the hands, defined by the handshape, location, orientation; a movement period; and a final hand articulation. The movement period can refer to either the semantically relevant movements of the hands or the period when one or more of the basic units is in a transient state. Fundamentally, this model highlights that a sign is not an instantaneous event defined by the current hand gesture and facial expression. Signs are inherently formed simultaneously across space and time.

ASL signs are supplemented by additional gestures known as *fingerspelling.* In fingerspelling, individual signs correspond to English letters and are used to explicitly spell out words, proper nouns, or technical terms for which there is no associated ASL sign. Fluent ASL users can fingerspell at a rate of up to 10 letters per second.

2.3 Face-centric Communication

Nonmanual signals, consisting primarily of facial expressions and head movements, add a substantial amount of contextual information to a conversation [6, 47]. A signer's gaze can indicate pronomial references or quotations. Raising or furrowing the eyebrows indicates a question or a negation, respectively. The contextual detail added through nonmanual signs suggests that accurate interpretation of facial expression is essential for understanding ASL.

When having a sign language conversation, the receiver tends to fixate on the signer's face [8, 73]. This has recently been confirmed by eye tracking studies [4, 13, 30, 53] that demonstrate that a fluent sign language user will gaze at a signer's face approximately 95% of the time, with brief excursions to the hands. When these excursions occur, they are almost always because the signer's gaze directs the receiver's gaze to the hands, i.e., the receiver looks away from the signer's face when the signer looks at her own hands [30]. This face-centric nature of ASL communication has been explained in two ways. The first hypothesis claims that features of the human visual system have influenced the evolution of sign language [73]. Visual acuity is at its maximum at the point of fixation and decreases significantly in the periphery. In regions of high visual acuity, an observer can differentiate finer details more easily. For example, fingerspelling occurs near the face, because the receiver is looking at the face and needs to be able to differentiate each of the fingerspelled letters.

The second hypothesis claims that fixation on the face is a result of a greater number of visual 'landmarks' in that region, such as the signer's lips, chin, eyes, or nose [8]. The signer's face contains 29 unique locations at which a sign can be formed, compared to only 19 on the torso and 6 on the arm [48]. The high level of information contained in the signer's facial expressions implies that when coding ASL video for communication, maintaining higher fidelity in the signer's face is important. Maintaining high fidelity in the face matches the viewing patterns of ASL because the receiver is known to be looking at the signer's face and because visual acuity is maximized at the point of fixation.

2.4 Receiver-centric Studies of Temporal and Spatial Fidelity Reductions

A review of ASL phonology describes the use of gestures for communication between ASL users. However, an understanding of the phonology produces only hypotheses that predict the effect of fidelity reductions caused by video coding and transmission distortions on ASL communication. Receiver-centric perceptual experiments with fluent ASL users provides a method for testing the hypotheses provided by the phonology alone. The intelligibility of ASL has been explored in the context of reductions to both spatial and temporal fidelity for either individual signs or for well-formed sentences. Frame size has also been studied, but has no demonstrable effect on intelligibility for resolutions as low as 240×180 [41], which is exceeded by the display sizes of modern mobile devices.

ASL communication is very robust to substantial changes in the image representation. In two separate studies, [77] and [62] both analyze the intelligibility of point-light presentations of individual ASL handshapes. In these experiments, reflective tape was fixed to a signers hand in multiple locations and signs were recorded in low light settings. The signs were presented to participants as moving points of light and participants were able to consistently and accurately discriminate different signs. Evaluating the intelligibility of such sparse representations is a difficult task and beyond the scope of this work. However, these experiments serve to demonstrate the robustness of ASL communication. Despite a radically changed spatial representation, these systems maintain the basic features of the signs, i.e., the handshape, location, orientation, and movement, allowing for accurate recognition.

For natural video sequences, several studies evaluate ASL comprehension for videos with varying frame rates, in the absence of compression, both for individual signs and for conversational sign language. The accurate discrimination of an individual sign depends only on the fidelity of the handshape, orientation, and movement, and does not rely on nonmanual signals. Through an analysis of the biological limits of human movement, the bandwidth of human motion used during signing was found to be limited to 3 Hz [31]. Using stick figure animations, sampling individual signs at the Nyquist rate of 6 frames per second (fps) is sufficient for capturing the relevant movements. This result is supported by several other studies [39,58,79]. In particular, [39] demonstrated that, for individual signs, recognition accuracy was greater than 91% at 6 fps.

However, these intelligibility experiments were performed for individual signs, not full sentences in which facial expressions have a large impact on meaning. For fully formed sentences, [39] demonstrated that accuracy dropped significantly at frame rates below 10 fps. A separate study evaluated the intelligibility of full sentences with variations in the frame rate of fingerspelling segments versus signing segments of an ASL conversation [17]. In this study, periods of signing were presented at frame rates of 5, 10, and 15 fps. Periods of fingerspelling were presented with increased frame rates with respect to the signing segments, e.g., when signing was presented at 5 fps, fingerspelling was presented either at 5, 10, or 15 fps. The subjective intelligibility ratings for a particular signing frame rate were nearly identical regardless of the varying fingerspelling frame rates, indicating that the overall intelligibility of a sequence depends on the frame rate during signing. Furthermore, when signing segments were 10 fps and 15 fps, there was little difference in intelligibility, while each of these cases provided significantly higher intelligibility than videos at 5 fps.

Similarly, [41] analyzed the effects of frame rate on learner comprehension and came to a similar conclusion. Participants viewed stories at frame rates of 18 fps, 12 fps, and 6 fps and repeat the story to a video camera. Statistically significant decreases in story-retell performance only occurred for the 6 fps case. Based on these studies, the intelligibility of videos with increasing frame rates rises rapidly but quickly reaches an asymptote around 10 fps, beyond which further increases in frame rate provide diminishing improvement in intelligibility.

When sufficient frame rate is provided, intelligibility is greatly reduced when spatial fidelity is lost. If the face and hands are not perceived clearly, relevant gestures in ASL cannot be identified and communication will be difficult. In [43], the reliance on text messaging is studied in the context of decreasing video telephony bitrates. Two participants communicated through H.323 videoconferencing terminals, capable of transmitting both video and text messages. The encoding bitrate of the video was fixed at either 400 kbps, 128 kbps, or 64 kbps and participants were free to ask and to respond to questions either by signing or by sending a text message. During a practice session, the participants were instructed to adjust a slider controlling the video quantization factor, trading-off spatial quality and frame rate for fixed frame size (the frame size was not reported). The frame rates vary between less than 10 fps and 15 fps, however, the authors do not report at what frame rate the videoconferencing sessions ultimately took place. At the highest tested bitrate of 400 kbps, participants used text messaging 9% of the time while at 64 kbps, text messaging was used 31%. As the video quality was reduced, participants had difficulty communicating using sign language and relied more heavily on text messaging.

To explicitly evaluate the trade-off between spatial fidelity and frame rate for frame rates beyond 10 fps, [13] analyzed the intelligibility of videos encoded at fixed bitrates of 15 kbps, 20 kbps, and 25 kbps each at frame rates of 10 fps and 15 fps. The videos were encoded at a resolution of 320×240 pixels using an H.264/AVC compliant encoder. For all the tested bitrates, intelligibility was higher for sequences at 10 fps than at 15 fps. Since a fixed bitrate encoding scheme was used, individual frames at 10 fps are less heavily quantized and have fewer spatial distortions. The increase in spatial fidelity was more important for intelligibility than was the corresponding reduction in frame rate, providing that the frame rates were sufficiently high.

Collectively, the above experiments demonstrate that spatial fidelity is the most important for both conversational ASL and for single sign recognition, when the frame rate exceeds approximately 10 fps. If the face and hands of the signer are not clear, the sign cannot be accurately perceived. Conversational ASL is less robust to reductions in temporal resolution than individual signs, implying that relevant events contained in subtle facial expressions, present only in conversational ASL, are no longer perceptible at low frame rates.

2.5 Encoder-centric Studies of Prior ASL Encoding Systems

The previous sections discussed the formation and communication of information in ASL and reviewed several receiver-centric studies that characterize the impact of various video coding distortions on ASL intelligibility. Understanding these components affords the design of ASL-specific encoding algorithms, typically designed to encode intelligible ASL video with limited resources such as bitrate. This section reviews several ASL-specific encoding algorithms, highlighting those that were evaluated in encoder-centric studies.

Early ASL encoding systems operated at low bitrates by transforming the input video into a binary representation [46, 50, 74]. The intelligibility experiments using point-of-light presentations discussed in Section 2.4 demonstrate that a binary image sequence may be a viable representation for intelligible ASL communication. Each of these systems generates a sequence of binary cartoon images which preserves the edges and contours in the original video frames. Because these representations are sparse, they can be encoded efficiently at low bitrates.

While these systems are capable of satisfying the bandwidth constraints of a cellular network, they require specialized encoding and decoding algorithms capable of compressing binary images, making implementation on complexity-constrained devices difficult. Additionally, in each of the binary encoding systems presented, the original videos were recorded in a controlled setting, i.e., smooth and static background with controlled lighting. In all cases, this allows the edge-detection algorithm to only capture edges and contours belonging to the signer. While not addressed in the previous work, natural background scenes (such as an outdoor setting) will likely cause problems for the edge-detection algorithms.

Substantially more effort has been placed on developing block-based, motion-compensated video coders, which provide efficient compression of natural video. Common among all block-based encoding algorithms designed for ASL video is the allocation of more bits to the blocks containing important regions such as the signer's face or hands [4,54,66,67]. More specifically, most ASL-specific algorithms can be placed into two classes: foveated video encoding and region-of-interest (ROI) encoding.

The foveated video encoding algorithms exploit the face-centric viewing patterns of ASL [4, 54]. In foveated video coding, the video frame is encoded with non-uniform, decreasing quality away from the the observer's point of fixation, attempting to match the visual acuity of the human visual system [45]. Because the ASL receiver primarily gazes at the signer's face, the fixation point is assumed. In [4], the face is identified automatically using skin segmentation and facial feature detection, and foveated processing is applied to generate a map of priority regions. Given the location of the face, a foveation model assigns macroblocks to the priority regions. Increasing quantization step sizes are applied to each region, allowing blocks nearest to the face to be coded with more bits than blocks farther away. These modifications conform to the H.264/AVC standard and were applied to four CIF size sequences recorded at 25 fps. At average rates of 217 kbps, this algorithm achieved an average bitrate reduction of 40% over the H.264/AVC reference encoder (JM) without affecting the intelligibility of the sequence, as verified through subjective evaluation.

In [54], three techniques are applied for improving SL compression in H.263: foveation-weighted bit allocation, modified macroblock processing order, and forced SKIP mode in background blocks. The weighted bit allocation decreases the rate allocated to each macroblock as a function of increasing distance from the face. The modified processing order adjusts the analysis of blocks, such that blocks near the face are analysed first. The encoder will obtain information about the face blocks earlier in the encoding process. Finally, a set of background macroblocks at the edges of the frame are identified and are always encoded in the SKIP mode. These techniques allowed more bits to be assigned to the face and regions near the face, but requires that the weights and block labeling are manually tuned prior to encoding. The source content used was 15 fps sequences at both CIF (352×288) and QCIF (176×144) resolutions. A subjective experiment demonstrated that at fixed bitrates of 256 kbps, 128 kbps, and 64 kbps, the proposed algorithm had higher mean opinion scores than the H.263 test model.

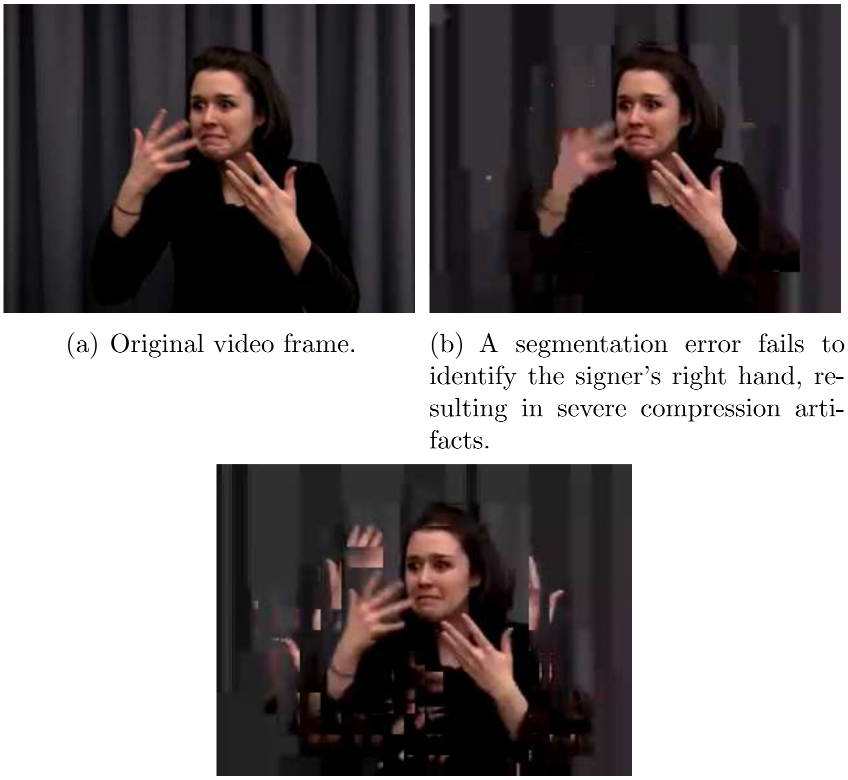
Knowing that information in ASL is communicated through hand gestures and facial expressions, ROI encoding algorithms use segmentation techniques to identify the signer's face and hands and encode these regions with a higher quality than the rest of the video frame. Both [67] and [66] use automatic skin segmentation to identify the ROI. These algorithms assign more bits to the face and hand blocks by adjusting quantizer values and severely compressing all non-skin blocks. In [67], QCIF (176×144) sized videos were encoded at fixed bitrates of 64 kbps and 128 kbps using H.261. Reductions of 10-15% in the number of bits per picture led to slight increases in the effective frame rate, relative to an encoding technique that assigns a uniform quantizer to the entire frame. The effective frame rates for the sequences were 16.3 fps at 64 kbps and 17.7 fps at 128 kbps.

Similarly, a more recent approach encodes the non-ROI using the largest possible quantization step size and an additional preprocessing step that blurs all the non-skin regions, with the intention of reducing blockiness in the background [66]. MPEG-1, motion JPEG, and the Windows Media Encoder were applied to video sequences recorded at 30 fps with a resolution of 160×120. By reducing the quality in the background region, bitrates were reduced by 25% over the cases in which no region-of-interest coding was used. In both [67] and [66], no formal subjective study was performed to evaluate the intelligibility of the compressed ASL video.

Each of these encoding techniques appropriately considers the viewing patterns of a sign language receiver, but heuristically distributes rate between the regions-of-interest. Furthermore, in the two applications that incorporated an encoder-centric subjective experiment [4, 54], the videos were evaluated and encoded at relatively high rates, resulting in fully intelligible video for both the standard approach and the ASL optimized approach. As the encoding bitrate is reduced, the impact on intelligibility of the distortions introduced by the ASL-specific encoding algorithms is different than the impact of distortions caused by a standard encoding approach. The following section analyzes the behavior of these encoding systems at the extremely low bitrates available on cellular networks.

2.6 Distortions Caused by ASL Encoding Systems and Their Impact on Intelligibility

The foveated coding approaches achieve bitrate reductions by increasing the quantization step size for macroblocks with increasing distance from the signer's face. To be consistent with models of visual acuity, the quality degrades gradually between the macroblocks in the highest priority region (the signer's face) and the macroblocks in the lowest priority region (video frame edges). Rate control is per-



(a) Original video frame.

(b) A segmentation error fails to identify the signer's right hand, resulting in severe compression artifacts.

(c) Improperly coding background blocks creates residual hand artifacts that reduce intelligibility.

Figure 2.1: Comparison of distortions that can impact the intelligibility of the video. Intelligibility is reduced when spatial distortions occur in relevant regions of the signer, such as the hands in (b) or when background distortions interfere with the receiver's ability to accurately interpret the motion of the signer, as illustrated in (c).

formed by selecting the desired quantization step size for the highest priority region and computing the remaining step sizes according to the foveation model. Because the foveated coders do not explicitly differentiate between signer and background, the background regions near the signer's face are allocated a significant proportion of the total bitrate. In order to achieve extremely low target bitrates, the foveated coders must select a large quantization step size for the signer's face, resulting in distortions common to all motion-compensated, block-based encoders, such as blurring and ringing, which reduce the perception of finer details in the image [92]. Fine details in the signer's face are critical for the receiver's understanding and distortions here will significantly reduce intelligibility.

The ROI algorithms explicitly identify the signer's face and hands and achieve very low bitrates by allowing the non-ROI regions to be allocated almost zero bits. This encoding technique can potentially generate intelligible videos at lower bitrates than the foveated coding techniques. However, the ROI encoding approach relies on the accurate detection of both the signer's face and hands, increasing the complexity of the encoder due to the need for more advanced segmentation algorithms. If the ROI is not correctly segmented, important regions will be heavily distorted. For example, in the frame in Figure 2.1(b), the signer's hand is incorrectly labeled as belonging to the background. The frame is encoded using an ROI technique, which leads severe compression artifacts in the mislabeled hand, obscuring the handshape and severely reducing intelligibility. Foveated coding techniques are more robust to this type of distortion, because they only require the identification of the signer's face and code all macroblocks in the frame with at least some nominal rate.

In addition to accurate segmentation, the ROI encoding techniques require intelligent processing of the non-ROI macroblocks. A direct application of spatial ROI coding can lead to insufficient rate allocated to the non-ROI macroblocks, creating distortions outside of the ROI that negatively impact intelligibility. In a motion-compensated encoding framework, the lowest possible rate for a coded macroblock is achieved by applying the SKIP mode, which simply copies to the current frame the co-located macroblock in the previous frame. In the ROI encoding techniques, the background macroblocks are commonly skipped to conserve rate for the ROI. If all the background macroblocks are encoded using the SKIP mode, when a particular macroblock contains a face or hand in one frame and contains only background in the next frame, residual pieces of the face and hand remain in the macroblock. An example of a frame with many residual face and hand macroblocks is provided in Figure 2.1(c). These residuals will propogate either until the macroblock is coded as a face/hand or until an intra frame is inserted. Fluent ASL observers note that these types of compression artifacts make it difficult to follow the hand movement and to focus on the signs. Perceptual evidence suggests that the human visual system (HVS) extrapolates the current visual stimulus to predict the location of objects in the next perceived moment [16]. Because these distorted background macroblocks are temporally correlated with relevant objects, such as the signer's hands, they interfere with the expecations of the HVS and inhibit the receiver's ability to accurately interpret the sign movements.

2.7 Summary

This chapter describes the ASL communication process, detailing how information is distributed among the signer's face, hands, and body. Both receiver-centric and encoder-centric perceptual experiments describe the impact of degradations on perceived intelligibility. These studies, as well as an analysis of ASL encoding systems, define the relevant distortions and their perceptual impact, which informs the design of the CIM-ASL.

CHAPTER 3

SUBJECTIVE EVALUATION OF INTELLIGIBILITY

3.1 Introduction

Three experiments verify and refine the conclusions drawn in the studies reviewed in Chapter 2. More importantly, these experiments yield a quantifiable measure of the subjective intelligibility for a collection of distorted ASL videos. This ground-truth experimental data is critical for parameterizing the CIM-ASL and for evaluating the accuracy of the CIM-ASL, as well as other computational models. The three experiments use consistent methodologies and source videos, but vary the treatments applied to the videos. Each experiment was designed to evaluate the effect on intelligibility of different video coding distortions. A specific combination of experiment parameters, namely the encoding algorithm, encoding bitrate, and encoding frame rate, is denoted a *hypothetical reference circuit* (HRC) [84]. A reference video is processed by a particular HRC to create a processed video, which is displayed to and rated by the participant. The HRCs varied in each of the three experiments. An HRC can be used to create specific distortions for a receiver-centric study, as in experiments 1 and 3. It can also correspond to an encoding system evaluated at a specific operating point (or points), as in the encoder-centric experiment 2.

The experimental methodology common to each of the experiments is described in Section 3.2. Specific details for each experiment, including a description of the HRCs and a discussion of the experimental results, are provided in Section 3.3. The procedure described in Section 3.4 identifies and eliminates unwanted bias in the ground-truth subjective intelligibility scores.

3.2 Experimental Methodology

3.2.1 Reference Stimuli

The reference video sequences for each study consist of sign language stories told by a fluent signer at her natural signing pace. The stories were filmed in two different locations: an indoor studio with a static background, denoted *indoor videos*, and an outdoor location on a busy street, denoted *outdoor videos.* The sequences all have a spatial resolution of 320×240, which matches the display of the testing device. The videos used in experiment 1 were filmed at 30 fps, while the videos used in experiments 2 and 3 were filmed at 60 fps.

3.2.2 Test Procedure

The test procedure for each of the three experiments was identical. The subjective experiment followed a single stimulus testing procedure. Participants viewed a processed video and, following the viewing, were asked three questions designed to evaluate their comprehension of the story, the intelligibility of the test video, and the usability of the test video. Specifically, the first question asked about the story content, encouraging participants to remain focused while watching the video. The second question asked "How easy or how difficult was it to understand the video?" and participants responded on a 5-point scale ranging from "very difficult" to "very easy". The response to this question is denoted the *intelligibility score.* The third question asked "If video of this quality was available on a cell phone, would you use it?" and participants responded on a 5-point scale ranging from "definitely no" to "definitely yes". The response to the question is denoted the *usability score.*

Because of the nature of the intelligibility assessment task, no single sign language story was viewed by the same participant twice, eliminating any possible learning effects.

In order to simulate the use of a cellular device for ASL communication, processed videos were displayed to participants on an HTC Apache pocket PC with a diagonal screen size of 2.8 inches and a screen resolution of 320×240 pixels. Participants were allowed to hold the device at a comfortable viewing distance.

3.2.3 Participants

A total of 50 fluent ASL users participated in the three experiments: 18 in experiment 1, 16 in experiment 2, and 16 in experiment 3. Participants were screened for consistency using their answers to the story content questions. If a participant answered this question incorrectly and rated the video with either a 4 or 5, i.e., easy to understand, this rating was flagged as an error. If more than 25% of a participant's scores were flagged as errors, that participant's scores were considered invalid and the participant was discarded as an outlier. If fewer than 25% of a participant's scores were flagged as errors, the participant's scores were considered valid and none were discarded. Using this approach, 2 participants were removed from experiment 3.

Table 3.1: Details for the experimental data from three experiments, which are used in the training and validation of the CIM-ASL. The total number of processed videos in an experiment is the multiplication of the number of videos per HRC and the number of HRCs studied in the experiment. The ratings per processed video corresponds to the number of participants who rated a specific reference video encoded using a specific HRC. This number varies in experiment 3 because of the removal of 2 outlying participants. Each experiment is divided into video subsets, which are used to train different components of the CIM-ASL.

Experiment	Number of Participants	Number of HRCs	Number of Videos per HRC	Ratings per Processed Video	Video Subset	Number of Processed Videos	CIM-ASL Parameters Trained on Subset
Experiment 1	18	18	6	3	10 FPS	54	α_k, β_k
					15 FPS	54	
Experiment 2	16	16	4	4	Indoor	32	α_k, β_k
					Outdoor	32	
Experiment 3	14	15	5	2 or 3	6 FPS	15	a_1, a_2, a_3
					7.5 FPS	15	
					10 FPS	15	
					15 FPS	15	
					20 FPS	15	

3.3 Experimental Results

3.3.1 Experiment 1: Varying spatial quality in a fixed region-of-interest

Experiment 1 was a receiver-centric study that determined the impact on ASL intelligibility of varying levels of spatial distortions [13]. A collection of 18 indoor videos were used in this study, with videos ranging in length from 58 seconds to 177 seconds. In this study, 18 different HRCs were evaluated, corresponding to combinations of the following parameters: three bitrates (15 kbps, 20 kbps, and 25 kbps), two frame rates (10 fps and 15 fps), and three region-of-interest (ROI) rate allocation schemes. The videos were coded using x264, an open-source, standards-compliant implementation of the H.264/AVC standard [2]. For the ROI rate allocation, a fixed region was defined around the signer's face and the quantization parameter for macroblocks in that region was offset by either 0, -6, or -12, resulting in fewer distortions around the face at the expense of increased distortions in the rest of the frame.

Analysis of variance (ANOVA) is used to identify statistically significant effects on the subjective intelligibility ratings and all three studied parameters demonstrate a significant effect. For each tested bitrate, the higher rates are statistically preferred over the lower rates, i.e. 25 kbps is preferred over 20 kbps which is preferred over 15 kbps ($F(2, 34) = 51.12, p < 0.01$). Participants preferred videos encoded at 10 fps over videos at 15 fps ($F(1, 17) = 4.59, p < 0.05$). As highlighted in Chapter 2, 10 fps is sufficient for ASL conversations and, since the videos were encoded at a constant bitrate, the videos at 10 fps have less distortion than the videos at 15 fps. Finally, the participants preferred the -6 quantization parameter offset, while the 0 and -12 offsets were not statistically different ($F(2, 34) = 13.69, p < 0.01$). With no offset, the face was not clear enough and with a -12 offset, the distortion outside of the ROI (e.g., in the signer's hands) was too high relative to the improvement in the signer's face.

3.3.2 Experiment 2: Encoder-centric algorithm comparison

Experiment 2 was an encoder-centric study that evaluated the performance of three different video encoding algorithms designed for sign language video in addition to a traditional H.264/AVC MSE-based rate control algorithm [20]. A collection of 8 indoor and 8 outdoor videos were used in this study, having lengths ranging from 17 seconds to 69 seconds. The average story length for the indoor sequences was 35 seconds and the average story length for the outdoor sequences was 56 seconds. The test set of videos evaluated in this study were generated using 4 different encoding algorithms each operating under both a high bitrate and low bitrate setting. Two different sets of rates were selected for the indoor and outdoor videos such that at each location, the most intelligible videos would be very easy to understand and the least intelligible video would be very difficult to understand. Outdoor videos were encoded at rates of 50 kbps and 80 kbps; indoor videos were encoded at rates of 30 kbps and 45 kbps. The combination of 4 encoding algorithms and 2 different bitrates at each location results in 8 HRCs for the indoor videos and 8 HRCs for the outdoor videos. All the videos in experiment 2 were encoded at 15 fps.

Each of the 4 encoding algorithms evaluated in this study operate within the H.264/AVC standard and provide rate control that meets an average target bitrate. Three sign language specific encoders and a general purpose video encoder were evaluated. The four encoding algorithms used were a traditional MSE-based rate control algorithm [2], described in Chapter 6; a foveated video encoding algorithm [4], described in detail in Chapter 2; and two ROI encoding techniques, one using a spatial ROI and one using a spatial-temporal ROI, in which segmentation labels are propagated into the future for a fixed duration. The ROI encoding techniques were preliminary versions of the coder described in Chapter 7 and they allocate bits primarily to the face and hands of the signer by varying the quantization step size in each coded macroblock.

The ANOVA identifies a significant effect for the encoding algorithm ($F(3, 206) = 3.90, p < 0.01$) and encoding bitrate ($F(1, 206) = 31.46, p < 0.01$). For the high bitrate videos, there is no statistical difference in intelligibility between the x264, foveated, and spatial-temporal ROI encoding algorithms. Both the x264 encoder and the foveated encoder yield statistically significantly higher intelligibility than the spatial ROI encoder. The reduced performance of the spatial ROI encoder is a consequence of compression distortion artifacts in the signer's face and hands due to ROI segmentation errors. The spatial-temporal ROI encoder is less susceptible to segmentation errors because the region labels persist across time, eliminating any short duration segmentation errors. For encoding algorithms that rely heavily on region-based rate allocation, accurate segmentation is an important factor in the final subjective intelligibility. At low bitrates, all three of the ASL-specific encoding algorithms provide statistically significant improvements over x264 in the average intelligibility scores. The differences in intelligibility between the three ASL-specific encoders are not statistically significant. This experiment demonstrates that encoding algorithms designed specifically for ASL can provide statistically significant improvements in intelligibility over traditional, MSE-based encoding algorithms.

3.3.3 Experiment 3: Varying frame rate at fixed levels of compression

Experiment 3 was a receiver-centric study that quantified the reductions in ASL intelligibility due to changes in the temporal resolution, using 15 indoor videos having lengths ranging from 28 to 62 seconds [26]. Processed videos evaluated in this study were generated according to 15 HRCs: 5 frame rates (6, 7.5, 10, 15, and 20 fps) each at 3 levels of spatial distortion (high, medium, and low). In this experiment, a reference video was processed at only one level of spatial distortion and all 5 frame rates. In order to accurately quantify only the effect of frame rate on intelligibility, the spatial distortion was held constant for each frame in a sequence, regardless of the frame rate. As a result, the bitrate varies across each sequence.

The ANOVA results for experiment 3 demonstrate a significant effect due to varying frame rates ($F(4, 165) = 8.87, p < 0.01$) and due to varying spatial distortion ($F(2, 165) = 76.94, p < 0.01$). Consistent with the prior experiments described in Chapter 2, videos are 20 fps are not significantly different from videos at 15 fps and 10 fps, but they are significantly better than 7.5 fps and 6 fps. The interaction between frame rate and spatial distortion is not significant ($F(3, 206) = 3.90, p < 0.01$), indicating that the effect on intelligibility of reduced frame rate is consistent across all three levels of spatial distortion.

These three experiments provide a collection of subjective intelligibility scores associated with a set of processed videos. Table 3.1 summarizes the experiments, providing for each experiment the number of participants, the number of HRCs evaluated, the number of processed video per HRC, and the number of intelligibility scores per processed video. The following section describes the processing that is applied to the subjective data. The raw intelligibility scores collected in the subjective experiments are converted to z-scores prior to their use in parameterizing the CIM-ASL (cf. Chapter 4). Converting to z-scores necessarily removes a bias in the intelligibility ratings of a subset of participants, identified through the statistical analysis provided in the following section.

3.4 Processing the Subjective Intelligibility Data

In experiments 1 and 2, demographic data was collected to identify and eliminate any potential sources of bias in the intelligibility scores. In particular, participants in experiment 1 belonged to one of three categories: hearing child of deaf adult (CODA), deaf, or hearing. Participants in experiment 2 were asked to report their preferred language as either English, ASL, or both. These categories are not exactly one-to-one matches, e.g., a deaf individual may prefer to communicate in either ASL or English. However, these demographic categories identify the group of participants who would most benefit from the availability of a mobile device capable of facilitating conversations in ASL, namely people who are deaf or people whose preferred language of communication is ASL. Analysis of variance (ANOVA) demonstrates that this group provides a biased response to both the intelligibility and usability questions.

For the intelligibility scores, ANOVA determines that the demographic group has a statistically significant effect on the scores. (In experiment 1, $F(2, 321) = 11.17, p < 0.001$ and in experiment 2, $F(2, 206) = 28.87, p < 0.01$).) When ANOVA identifies that a statistically significant effect exists, Tukey's multiple comparison test is applied to identify specifically which demographic groups have different mean intelligibility scores.

In experiment 1, participants identified as being deaf responded with statistically significantly higher intelligibility ratings than both the CODA and hearing groups. In experiment 2, participants who reported ASL as their preferred language responded with statistically significantly higher intelligibility ratings than both the group preferring English and the group preferring either ASL or English. In both experiments, higher fluency is unlikely. As an objective measure of fluency, the ratio between years of using ASL and age is computed for each participant. In experiment 1, the CODA group has the same average ratio of experience with ASL as the deaf group. In experiment 2, the group preferring both ASL and English has the same average ratio of experience as did the group preferring only ASL. Discounting higher fluency, it is more likely that both the ASL group and deaf group were biased toward higher intelligibility ratings. This is a consequence of the increased desire a deaf user likely has for a mobile phone that offers video communication; making cell phone calls in their preferred language is currently unavailable for only this group of participants.

This phenomenon is confirmed by applying ANOVA to the subjective usability ratings. Similar to the results for intelligibility, the participant's demographic category is a significant effect and both the deaf group and the group preferring ASL responded statistically significantly higher to the usability question, emphasizing their increased desire for such a technology.

In order to confidently apply the subjective intelligibility scores to the training and testing of the CIM-ASL, this bias must be eliminated. The subjective intelligibility scores are converted to z-scores, which has the desired consequence of removing information about between subject differences [81]. The z-scored intelligibility scores are used for all further analysis. Subsequent use of *subjective intelligibility scores* refers to the z-scored values. The processed videos and their associated intelligibility scores are used to train the model parameters and to test the performance of the CIM-ASL, in addition to other computational models of video quality.

3.5 Summary

Three perceptual experiments were presented to evaluate the subjective intelligibility of degraded ASL video. The experiments confirm the results described in Chapter 2 and provide ground-truth ratings of intelligibility for a collection of videos. The use of z-scored data is justified via ANOVA, which confirms a statistical bias in a group of participants. The results of these experiments will be used for training and testing the CIM-ASL.

CHAPTER 4

CIM-ASL: A COMPUTATIONAL INTELLIGIBILITY MODEL FOR COMPRESSED ASL VIDEO

4.1 Introduction

The ASL optimized encoders discussed in Chapter 2 either relied on assumptions that intelligibility was unaffected by the coder or performed encoder-centric subjective studies to determine the intelligibility of the compressed video. The assumptions are not necessarily valid in all scenarios and performing subjective experiments is costly and difficult to incorporate into the design cycle of an encoding system. The CIM-ASL provides a method for reliably predicting the performance of an ASL specific encoding system without the need for subjective testing. Furthermore, a suitable model can be used to create an ASL optimized video encoder by incorporating the model into a rate-distortion optimization algorithm (cf. Chapter 7). The CIM-ASL must accurately quantify the impact of distortions that reduce intelligibility, particularly those highlighted in Chapter 2, by applying knowledge of ASL phonology.

As illustrated in Chapter 2, information in ASL is communicated through gestures in the signer's face and hands. The gestures are formed in a systematic way and can be described by their spatial and temporal configurations. We define the *spatial coherence* as the consistent and semantically valid organization of a sign across space. The spatial coherence is determined by the location, orientation, handshape, and nonmanual signals associated with a particular sign. The loss of spatial fidelity in the signer has a strong effect on intelligibility, because the distortions that impact spatial fidelity disrupt the spatial coherence of the sign.

Disruptions to spatial coherence can be quantified by measuring the distortion in only the pixels containing the signer's face, corresponding to the nonmanual signals; hands, corresponding to the handshape and orientation; and torso, corresponding to location of the hand relative to the torso.

We define the *temporal coherence* as the consistent organization of a sign across time; it is determined by movements in the signer's hands and by the sequential transitions in spatial configurations characterized by the hold-movement-hold model. Temporal coherence in ASL video can be disrupted either by distortions that reduce the perception of smooth motion across frames or by distortions that obscure the transitions defined by the hold-movement-hold model. Frame rate reductions in the coded ASL video affect both the perception of smooth motion and obscure the hold-movement-hold transitions; appropriately considering the frame rate is therefore essential for quantifying intelligibility.

In addition to decreasing frame rate, temporal coherence is disrupted by the following two video compression artifacts. First, improperly coded background blocks in the frame can result in distortions that affect the receiver's ability to follow the hand movements (see Figure 2.1(c)). These erroneously skipped blocks are differentiated from background blocks that do not interfere with the temporal coherence, in order to quantify their impact on intelligibility.

Second, because ASL signs are defined by a sequence of the basic components, any disruptions to this sequence will affect the temporal coherence of the ASL video. Large increases in spatial distortions occurring over multiple frames can reduce intelligibility by making relevant portions of the hold-movement-hold sequence imperceptible. Detecting and measuring large temporal fluctuations in the distortions can quantify this disruption to the temporal coherence.

Only video coding distortions that disrupt either the spatial coherence or temporal coherence of a sign will reduce intelligibility (other distortions may cause annoyance, but will not impact intelligibility). The goal of the CIM-ASL is to quantify the extent to which such distortions impact the spatial and temporal coherence. The CIM-ASL is a full-reference distortion measure, which compares an uncoded reference video to a distorted test video and assumes the reference ASL video is maximally intelligible, i.e., easy to understand in the absence of any compression. The full-reference CIM-ASL can be used to estimate the subjective intelligibility of coded ASL video and to optimize an ASL-specific encoding system in order to generate compressed video having maximized intelligibility.

The remainder of this chapter details the CIM-ASL, which computes distortions in regions relevant to ASL communication with respect to their impact on spatial and temporal coherence. Section 4.2 explains the segmentation of the input video sequence into the relevant regions, namely, the face, hands, torso, and background. Given the region segmentation, the CIM-ASL quantifies the disruptions to the spatial coherence by measuring the distortion in the signer's face, hands, and torso, as detailed in Section 4.3. Section 4.4 describes three methods for quantifying disruptions to temporal coherence: a model of the temporal variation of the spatial distortions, a model for identifying and quantifying the impact of incorrectly coded background blocks, and a model that computes the reduction of intelligibility as a function of the video frame rate. The region distortions, which quantify disruptions to spatial coherence, are temporally pooled and combined with the temporal coherence measures. The final intelligibility score given by the CIM-ASL is the weighted combination of the pooled distortion measures for each region. The weighting mechanism accounts for the varying importance of the signer's face, hands, and torso. The pooling mechanism and weighting procedure are described in Section 4.5. Optimal values for the model parameters are selected using a heuristic optimization technique, described in Section 4.6.

4.2 Frame Segmentation

The input video is segmented into macroblocks containing either the face, hands, torso, or background; distortions in each of these regions have a varying impact on the spatial and temporal coherence and must be treated separately. The proposed segmentation simply adopts principles from several techniques proposed for segmenting the face and hands in sign language video, combining skin color models [5,36,61] for skin detection with classifier cascades for refined face detection [83]. Appendix A provides a more thorough treatment and analysis of segmentation techniques appropriate for real-time face and hand detection on a mobile device.

Combining the results from the skin segmentation and face detection, the macroblocks containing skin pixels that do not belong to the signer's face are identified as the signer's hands. The torso region is identified as the blocks below the signer's head having a width that is twice that of the face bounding box. The remaining unlabeled blocks are considered background blocks.

The background blocks are further differentiated into *new background* blocks and *sustained background* blocks. A co-located block that contains a face, hand, or torso in frame $n-1$ and contains only background in frame n is labeled as *new background.* All remaining blocks are labeled as *sustained background.* These new background blocks must be treated differently from the sustained background, because the new background blocks potentially contain the temporally correlated distortions highlighted in Figure 2.1(c), which reduce the temporal coherence of the sign.

4.3 Computing Disruptions to Spatial Coherence

Given the frame level segmentations, disruptions to the spatial coherence of ASL video can be computed as a function of the spatial distortions in the signer's face, hands, and torso. The CIM-ASL is computed using only the Y channel of the YCbCr component color space, the standard color space for MPEG and H.26x video encoding. Within each frame, a pixel-level map of distortions can be computed as errors in contrast,

$$e_c(i,j,n) = \frac{(Y(i,j,n) - \overline{Y}(n))}{\overline{Y}(n)} - \frac{(\hat{Y}(i,j,n)) - \overline{\hat{Y}}(n)}{\overline{\hat{Y}}(n)}, \tag{4.1}$$

where $Y(i,j,n)$ and $\hat{Y}(i,j,n)$ are the luminance pixel values of the original and processed videos at spatial location i, j in frame n. Normalization by the average pixel value in the original video frame and processed video frame, $\overline{Y}(n)$ and $\overline{\hat{Y}}(n)$, is an approximation of Weber's law; a fixed amount of error is more difficult to see in increasingly bright images [15].

In the domain of motion-compensated video encoding, the mean of each frame will not be significantly changed due to compression and $\overline{Y}(n) \approx \overline{\hat{Y}}(n)$. In this case, the errors in contrast are approximated by

$$\widetilde{e_c}(i,j,n) = \frac{(Y(i,j,n) - \hat{Y}(i,j,n))}{\overline{Y}(n)}, \tag{4.2}$$

where $\widetilde{e_c}(i,j,n) \approx e_c(i,j,n)$.

Given the pixel-level error map, the spatial distortions within the regions corresponding to the signer can be computed independently, according to

$$d_k(n) = \frac{1}{N_k} \sum_{i,j \in Region\ k} \widetilde{e_c}(i,j,n)^2, \tag{4.3}$$

where $d_k(n)$ is the mean squared error in contrast in frame n for region $k \in \{face, hands, torso\}$ averaged over N_k pixels. The measure $d_k(n)$ is a temporal trace of the distortion within a region across time.

When observing a video sequence, viewers track relevant, moving objects and are more sensitive to distortions in and around these objects. An observer does not integrate the distortions at a fixed pixel location over time, unless the pixel corresponds to a stationary object. For arbitrary video content, recently proposed quality assessment algorithms use motion prediction models to track the trajectory of objects, estimating the quality along those trajectories [7, 55, 69]. For ASL video, motion prediction models are unnecessary because the relevant objects are known to be the face, hands, and torso of the signer. By computing per-frame distortions separately for each region according to Eq. (4.3), the CIM-ASL is explicitly tracking the distortion in the objects across frames. The following two sections quantify the impact of temporal variations in these spatially-defined region distortions and describe the spatial and temporal pooling stage.

4.4 Computing Disruptions to Temporal Coherence

The disruptions to temporal coherence are quantified by three measures: a measure of the temporal variations in the spatial distortions computed in Eq. (4.2), a measure of distortions only in the new background blocks, and a measure of the intelligibility variations due to the video frame rate. Before computing the temporal variations of the per-frame spatial distortions, a temporal median filter is applied to the distortion traces from Eq. (4.3). Because of the temporal structure of sign language phonology, a single sign is formed over several video frames. If a distortion appears in only a small subset of those frames, the observer is still able to interpret accurately the sign being formed. The median filter eliminates short duration spikes in the distortion traces that are not likely to have a strong effect on the overall intelligibility of the sign and is applied to the region distortions according to

$$d'_k(n) = median\left(d_k(n - \frac{\gamma - 1}{2}) \ldots d_k(n + \frac{\gamma - 1}{2})\right), \tag{4.4}$$

where $d'_k(n)$ is the output of the median filter having odd-length γ. The median filter length, γ, depends upon the video frame rate and is selected to correspond to the number of frames in 500 msec, which is the average duration of an ASL sign. Fluctuations in distortions less than half the average sign length are removed.

The filtered region distortions, $d'_k(n)$, capture the spatial errors that reduce the intelligibility of the processed video. However, computing only the average distortion across all frames cannot account for the temporal distribution of distortions, which can have a large impact on the final intelligibility. A measure of the temporal variation of the distortions is adapted from [55]. For each of the regions, the temporal variation is computed as the average of the largest 5% of the positive gradients, which is given by

$$tv_k \quad = \quad avg_{5\%}(\max(\nabla d'_k(n), 0)), \tag{4.5}$$

where tv_k measures the temporal variation for region $k \in \{face, hands, torso\}$.

The temporal variation parameter penalizes abrupt increases in distortion that are not captured by a simple average of the frame level distortions. In particular, the temporal variation will be significantly higher for sequences in which a relevant region has been improperly coded, e.g., due to a segmentation failure in labeling the face or hands. Only positive gradients are used in this computation to avoid penalizing decreases in distortion.

The second measure of disruptions to the temporal coherence quantifies the impact of distortions in improperly coded new background blocks. As described in Section 4.2, new background blocks, or NewBG blocks, are identified as macroblocks that contain a face, hand, or torso in frame $n-1$ and contain only background in frame n. Erroneously encoding a NewBG block using the SKIP mode creates residual distortion artifacts that inhibit the perception of coherent motion in the signer, disrupting temporal coherence, as illustrated in Figure 2.1(c). NewBG blocks that do not contain these artifacts are treated as sustained background blocks because the distortions do not disrupt the temporal coherence (i.e., are encoded appropriately).

In order to differentiate the improperly coded NewBG blocks, the blockwise correlation coefficient is measured in the encoded video frames between the NewBG block in frame n and the co-located block in frame $n-1$. If the NewBG block was coded, the correlation will be low. If the current block was copied from the previous frame and contains residual hand or face artifacts, the correlation will be close to 1. The H.264/AVC encoding standard applies a deblocking filter to the coded video at macroblock boundaries. If the deblocking filter is not used, the correlation between skipped, co-located blocks will be equal to 1. It was empirically verified that a threshold of 0.9 was able to account for the changes caused by the deblocking filter. The distortion in only NewBG blocks having correlation coefficient greater than 0.9 is computed using the distortion contrast measure in Eq. (4.2) and is given by

$$D_{NewBG} = \frac{1}{N}\sum_{n=1}^{N}\frac{1}{N_k}\sum_{i,j\in NewBG}\widetilde{e_c}(i,j,n)^2, \tag{4.6}$$

which is averaged over N_k NewBG pixels and over N frames.

The final measure of disruptions to the temporal coherence quantifies the impact of reductions in video frame rate on intelligibility. For the same level of spatial quality, reductions in frame rate cause consistent and quantifiable reductions in subjective intelligibility [26]. The loss of linguistically important motion at reduced frame rates is modeled by an offset of the intelligibility measure, where the offset is determined by a sigmoidal function given by

$$f(r_f) \;=\; a_1 * \left(1 - e^{-e^{a_2 - a_3 r_f}}\right), \tag{4.7}$$

where a_1 controls the lower asymptote, a_2 and a_3 control the convergence locations and growth rate, and r_f is the frame rate in frames-per-second. A sigmoid is selected because the gains (or reductions) in intelligibility converge as frame rate increases (or decreases). The values for a_1, a_2, and a_3 are fit to experimental data.

4.5 Temporal and Spatial Pooling

The intelligibility score given by the CIM-ASL is computed by first temporally pooling the region distortions across all frames, before spatially pooling each of the regions, according to

$$D_k = \frac{1}{N}\sum_{n=1}^{N} d'_k(n) + \beta_k tv_k, \tag{4.8}$$

where D_k is the temporally pooled distortion trace, $d'_k(n)$, for region $k \in \{face, hands, torso\}$, including the corresponding measure of temporal variation,

tv_k. The temporal variation weight β_k controls the relative importance of the temporal variations with respect to the mean distortion level within a region.

The spatial pooling provides a measure of the distortions that reduce intelligibility for the coded video, which can be mapped to an objective intelligibility score as,

$$D_{Intell} = \sum_{k \in \{face, hands, torso\}} \alpha_k D_k + D_{NewBG} + f(r_f) \tag{4.9}$$

$$\text{CIM-ASL} = \log_{10} \frac{C}{D_{Intell}}, \tag{4.10}$$

where D_{Intell} is intelligibility distortion computed for the entire video and CIM-ASL is the final objective intelligibility score. The constant $C = 110^2$ is chosen empirically to map from a distortion measure to an intelligibility measure, where increasing CIM-ASL implies increasing subjective intelligibility ratings. The weights α_k reflect the relative importance of each region k, consisting of the face, hands, and torso.

The values of α_k, β_k, and the parameters in $f(r_f)$ are optimized using the ground truth subjective intelligibility ratings described in Chapter 3. The optimization procedures and optimal parameter values are described in the following section.

4.6 Parameterizing the CIM-ASL

With the appropriate selection of parameter values based on training data, the proposed CIM-ASL can be used to accurately estimate subjective intelligibility. The proposed intelligibility measure is parameterized by the median filter length γ from Eq. (4.4); by the weights applied to the region pooling, α_k from Eq. (4.9) and temporal variations, β_k from Eq. (4.8); and by the coefficients in the frame rate offset model, a_1, a_2 and a_3 from Eq. (4.7).

The median filter length is determined from ASL linguistics, while the remaining parameters are trained using the ground-truth subjective experimental data (see Table 3.1). Optimal values of α_k and β_k are selected using a genetic algorithm, which minimizes the root mean squared error (RMSE) of a linear mapping from the predicted intelligibility given by the CIM-ASL to the ground truth subjective intelligibility ratings from experiments 1 and 2. Non-linear regression analysis using data from experiment 3 provides the optimal frame rate coefficients, a_1, a_2 and a_3.

Because the training procedure relies on the subjective intelligibility ratings of degraded videos, it is important to highlight the range of distortions present in the training videos. The distortions contained in experiment 1 primarily disrupt only the spatial coherence of the ASL videos. In this experiment, a fixed region-of-interest was defined around the signer's face and this region was always encoded at either the same or higher quality than the rest of the frame. As a result, the distortions in the signer's face are low and do not vary significantly across time. Conversely, the signer's hands and torso will have a relatively high level of distortion, increasing with the increasing region quantization parameter offset.

In experiment 2, 4 different encoding algorithms applied to videos in 2 different locations results in a wide range of distortions, which disrupt both spatial and temporal coherence. The MSE-based rate-distortion optimization generates videos that have relatively consistent levels of distortion in each spatial location across the entire frame, which results in lower quality in the signer's face and hands when compared to the ASL specific encoding algorithms. The ROI-based encoding algorithm occasionally suffers from segmentation errors, causing frames in which the signer's hands are encoded with high levels of distortion, which leads to large temporal variations in the hand distortions.

Experiments 1 and 2 contain distortions that impact both the spatial coherence and temporal coherence, making them appropriate for identifying the optimal values of α_k and β_k. The distortions in experiment 3 were designed specifically to quantify the disruption to temporal coherence caused by frame rate reductions. In this experiment, each video was encoded with a fixed level of spatial quality, independent of bitrate and frame rate. As a consequence, there is no temporal variation of the distortions and there is very little spatial variation between the face, hand, and torso distortions. This experiment is used only for identifying the optimal frame rate coefficients, a_1, a_2 and a_3.

4.6.1 Model parameter optimization procedure

For training purposes, when computing the objective intelligibility, the ROI segmentation maps are generated from the original video as described in Section 4.2 and are manually corrected to remove any segmentation errors, guaranteeing that the results fairly characterize the performance of the proposed measure and not the accuracy of the segmentation algorithm.

The accurate recognition of a single sign, which occurs over multiple video frames, is not affected by distortions with short temporal durations. The median filter length, γ, is selected to be 500 msec, equivalent to the average duration of an ASL sign, such that fluctuations in distortions less than half the average sign length are removed.

Table 4.1: The optimal parameter values for the objective intelligibility measure. The median filter is selected according to the average length of an ASL sign and is identical for each segmented region. The temporal variation weight, β, controls the relative weighting between the average spatial distortion $d'_k(n)$ and the temporal variation tv_k. Because the hands contain more temporal phonological structure than the face, β_{hand} is larger than β_{face}. The spatial pooling weight, α, controls the relative importance of each of the segmented regions. Because the signer's facial expressions carry a significant amount of meaning in ASL, α_{face} is greater than α_k for each of the other regions.

Parameter	Face	Hands	Torso	NewBG	Relevant Equation (k corresponds to either face, hands, or torso)
Median Filter γ	500 ms	500 ms	500 ms	-	$d'_k(n) = median\left(d_k(n - \frac{\gamma-1}{2}) \ldots d_k(n + \frac{\gamma-1}{2})\right)$ (4.4)
Temporal Variation Weight β	2	4	0	-	$D_k = \frac{1}{N}\sum_{n=1}^{N} d'_k(n) + \beta_k tv_k$ (4.8)
Spatial Pooling Weight α	1.6	0.5	0.1	1	$D_{Intell} = \sum_{k \in \{face, hands, torso\}} \alpha_k D_k + D_{NewBG} + f(r_f)$ (4.9)

The region distortion weights α_k and the temporal variation weights β_k for each region are determined using a genetic algorithm (GA) optimization technique, trained on the 10 fps and 15 fps video subsets from experiment 1 and the indoor and outdoor video subsets from experiment 2. The GA is an iterative technique that efficiently searches a large space of parameter values. At each iteration, the GA generates a population, where each member of the population contains a selection of values for α_k and β_k, for all $k \in \{face, hands, torso\}$. Different population members have different values for α_k and β_k and each population member has an associated cost. Successive iterations in the GA propagate the population members with the lowest cost until either the cost no longer improves or an iteration limit is reached.

The GA computes cost in the following way. Given a population member, corresponding to values for α_k and β_k, the CIM-ASL is computed for the processed videos according to Eq. (4.10). For each of the 4 video subsets listed in Table 3.1, least-squares linear regression generates a mapping from the objective intelligibility measure to the subjective scores. The linearly mapped objective measure is denoted the *objective estimate* of the ground truth subjective intelligibility score. The root mean squared error (RMSE) is computed between the objective estimates and the intelligibility scores, as a measure of the accuracy of the objective estimate.

Variations in the number of data points in each of the 4 video subsets taken from experiments 1 and 2 can potentially bias the optimization to be overly sensitive to a single video subset and, consequently, a subset of the relevant distortion artifacts. To mitigate overfitting to the data, the GA jointly optimizes the prediction accuracy on each of the 4 subsets by computing the final cost as the sum of the 4 RMSE values, one for each video subset. The sensitivity of the CIM-ASL to each of the subsets is further explored in Chapter 5.

Overfitting is further avoided using a 3-fold cross validation technique, which partitions the data into sets of 3 and trains on only two-thirds of the data. Each fold yields slightly different values for α_k and β_k, but the statistical performance is consistent across each fold. The average of the three sets of parameter values (one from each fold) is chosen for the CIM-ASL. For each of the three folds, the RMSE when using the average parameters is statistically identical to the RMSE when using the optimal values for that fold.

4.6.2 Optimal parameter values and discussion

The GA identifies the optimal parameter values as $\alpha_k = [1.6, 0.5, 0.1]$ and $\beta_k = [2, 4, 0]$ for the face, hands, and torso, respectively; these values are summarized in Table 4.1. Recall from Eq. (4.8) and Eq. (4.9), also included in Table 4.1 for reference, the parameter α_k controls the relative contribution of distortions in each region to the overall intelligibility distortion measure. The optimal values of $\alpha_{face} = 1.6$ and $\alpha_{hand} = 0.5$ illustrate that spatial distortions in the signer's face will result in a larger decrease in intelligibility than spatial distortions in the signer's hands. Based on Eq. (4.9), the distortion weight for NewBG blocks is fixed to be 1 and is not included in the optimization procedure. In comparison to the optimized values for α_k, the temporally correlated distortions caused by NewBG blocks reduce intelligibility more than purely spatial distortions in the signer's hand but have a lesser impact than distortions in the signer's face. Ultimately, the optimal parameters are intuitively consistent with the semantics of sign language, i.e., larger weights are applied to distortions in regions containing more information. Distortions in the signer's face have the largest impact on the intelligibility because facial expressions carry a substantial amount of information.

According to Eq. (4.8), the parameter β_k controls the contribution of the temporal variation measure to the total distortion measure within a particular region, before the cross-region pooling, i.e., before the weight of α is applied. A higher value of β_k reflects a larger impact on intelligibility of the temporal variations in the distortions relative to the purely spatial distortion. The optimal values of $\beta_{hand} = 4$ and $\beta_{face} = 2$ illustrate that, prior to applying the region weight α_k, temporal variations in the hand distortions are twice as costly as temporal variations in the face distortions, in terms of the total spatial distortion. Once again, the optimal values are consistent with ASL linguistics. The temporal coherence of ASL is primarily determined by the consistent organization of handshapes across time. Relative to the purely spatial distortion $d'_k(n)$, temporal variations in the hand distortions have a larger impact on the temporal coherence than do temporal variations in the face distortions, which is reflected in the optimal values of β_k. Furthermore, the signer's torso does not contribute to the temporal structure of ASL and is only necessary for spatial coherence, as illustrated by the values of $\alpha_{torso} = 0.1$ and $\beta_{torso} = 0$ for the signer's torso.

Given values for α_k and β_k, the frame rate offset model is parameterized as follows, using subjective data from experiment 3. By the design of experiment 3, the computational intelligibility scores are nearly identical for a fixed level of spatial quality (high, medium, or low), when excluding the frame rate offset. The subjective intelligibility scores decrease for decreasing frame rate and the magnitude of the decrease is observed to be consistent across all three quality levels. This decrease in subjective intelligibility is modeled by the sigmoidal function in Eq. (4.7). Non-linear, least-squares regression is used to select the model parameters,

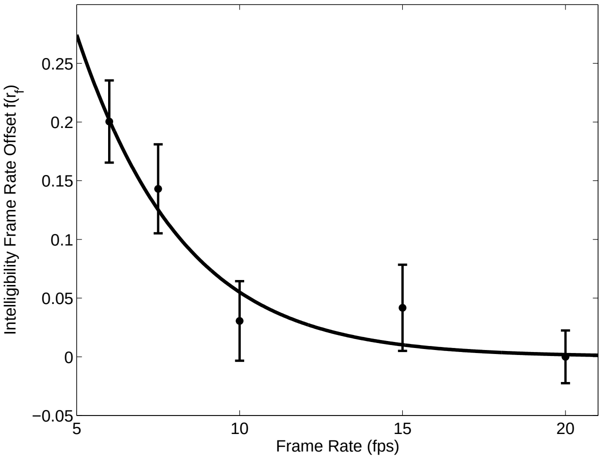


Figure 4.1: Sigmoidal relationship between the increase in intelligibility distortion caused by frame rate reductions, defined by $f(r_f) = a_1 * \left(1 - e^{-e^{a_2 - a_3 r_f}}\right)$. Error bars indicate the 95% confidence intervals. This model achieves $R^2 = 0.93$ and is within the 95% confidence interval for all of the experimental data.

which are given by: $a_1 = 1.3$, $a_2 = 0.26$, $a_3 = 0.34$. The experimental data and the functional mapping are plotted in Figure 4.1. This model is consistent with the receiver-centric experiments discussed in Chapter 2; intelligibility decreases rapidly at frame rates below 10 fps.

4.7 Summary

This chapter described the computational model of intelligibility for ASL (CIM-ASL), which is based on linguistic principles of ASL. The CIM-ASL measures distortions only in regions relevant for ASL communication, using spatial and temporal pooling mechanisms that vary the contribution of region-based distortions according to their relative impact on intelligibility. The parameters incorporated in the CIM-ASL, trained using a heuristic search technique, are intuitively consistent with the perception of ASL.

CHAPTER 5

ESTIMATING SUBJECTIVE INTELLIGIBILITY - STASTICAL PERFORMANCE AND DISCUSSION OF COMPUTATIONAL MODELS

5.1 Introduction

In the absence of alternative measures of intelligibility, the CIM-ASL is compared against computational techniques traditionally applied to video quality assessment. This section analyses the performance of these distortion measures, along with the proposed model, as estimators of subjective intelligibility. The experimental data consists of the processed videos and their associated intelligibility ratings taken from the 3 experiments discussed in Chapter 3 and summarized in Table 3.1. The ability of a computational model to estimate subjective intelligibility is determined via the linear regression between the objective scores computed by the model and the subjective scores, which generates a linear mapping from the objective scores to the subjective scores. The linearly mapped objective score is the objective estimate of the subjective intelligibility. The performance of an objective estimate is evaluated in terms of estimation accuracy, consistency, linearity, and monotonicity [85]. These four criteria are quantified by the statistical metrics of root mean squared error (RMSE), outlier ratio (OR), Pearson's linear correlation coefficient (r), and Spearman's rank order correlation coefficient (ρ), respectively.

When comparing the performance of two objective measures, it is critical to determine whether absolute differences in the statistical metrics are statistically significantly different. For the metrics r, ρ, and OR, the Student's t-test identifies statistically significant differences. For RMSE, the F-test identifies statistically significant differences [85]. Because of the high variance in the subjective scores, due to a limited number of scores per processed video, hypothesis tests are performed using 90% confidence level, rather than the typical 95% level. If two statistical metrics are statistically different at the 90% level, they may not be statistically different at the 95% level. However, if two statistical metrics are statistically equivalent at the 90% level, they will also be statistically equivalent at the 95% level.

Existing objective quality measures fail to accurately estimate subjective intelligibility and the proposed CIM-ASL achieves statistically significant performance improvements over the quality measures. Section 5.2 compares the performance of the objective measures when estimating the intelligibility of individual videos within each experiment. Section 5.3 analyses the performance of the objective measures when estimating the average intelligibility provided by each HRC [85]. An HRC can be considered a particular encoding system, or a combination of encoding algorithm, bitrate, and frame rate. This system-level analysis illustrates the effect on intelligibility due to encoding an arbitrary ASL video using a specific HRC. If, on average, an HRC yields highly intelligible videos, the HRC will be expected to work well in general, e.g., when deployed in a mobile device for real-time ASL communication. If a computational model can accurately predict this system-level performance, the model can be applied in the system selection process without the need for expensive subjective evaluation.

Following the demonstration of the performance of the CIM-ASL, Section 5.4 explores alternative methods for computing the spatial error measure (Eq. (4.2)) and the temporal variation (Eq. (4.5)). This analysis demonstrates that the success of the CIM-ASL can be attributed to the high-level structure of the model and the performance is robust to changes in the low-level implementation.

5.2 Estimating Intelligibility of Individual Videos Within an Experiment

The proposed CIM-ASL is compared against two objective techniques traditionally applied to quality assessment: PSNR and the structural similarity index for video (VSSIM) [87]. PSNR is selected for its simplicity as a signal-based error measure. VSSIM is selected for its demonstrated improvements over PSNR in terms of objective quality assessment. On each video frame, VSSIM computes the structural similarity index (SSIM), which measures the similarity between a reference and a distorted image as a function of local means, variances, and cross-correlations. VSSIM computes a score for a video sequence using a weighted temporal pooling, in which errors in frames with high motion activity are weighted less heavily than errors in frames with low motion activity.

Both PSNR and VSSIM fail to accurately estimate subjective intelligibility in all three experiments, having high RMSE and low correlation coefficients, as summarized in Table 5.1. In all three experiments, the proposed CIM-ASL achieves statistically significantly lower RMSE and higher linear and rank-order correlation than both PSNR and VSSIM. The performance improvement of the CIM-ASL is largest in experiment 2, because the videos in this experiment have the most diversity in the types of distortions present. The four different encoding algorithms studied in experiment 2 create distortions which vary both spatially and temporally to different degrees. The varying distribution of the distortions is especially challenging for the quality estimators PSNR and VSSIM.

The poor performance of PSNR and VSSIM can be attributed to their equal treatment of all distortions in a frame, regardless of their spatial location. Any full-frame distortion measure will perform poorly because it cannot differentiate between distortions in the signer's face and distortions in the background, each of which affect intelligibility in extremely different ways. As a consequence, traditional measures of quality cannot reliably estimate intelligibility.

PSNR can be modified to incorporate knowledge about the underlying structure of sign language. A foveated PSNR is computed by weighting the squared error with decreasing weights for increasing distance from the signer's face. The error weights are adapted from [45], which computes foveated super pixels that increase in size to match the reduction in visual acuity away from the point of fixation. The weight applied to the error for a pixel is inversely proportional to the size of its foveated super pixel, resulting in an objective measure in which distortions in and around the signer's face are more heavily weighted than distortions close to the edges of the video frame. This foveated PSNR model is intuitively consistent with the behavior of an ASL receiver; the ASL receiver is known to be fixating on the signer's face.

When compared to PSNR, the foveated PSNR has statistically significantly improved RMSE and correlation coefficients only for experiment 1. This experiment has few distortions that impact temporal coherence; simply emphasizing the importance of distortions in the face improves the estimation accuracy. However, this is insufficient for experiments 2 and 3, as the distortions present in these experiments affect both spatial and temporal coherence. The proposed CIM-ASL properly accounts for both spatial and temporal distortions and is statistically significantly more accurate, having lower RMSE, than the foveated PSNR in all three experiments, despite the consistency between a foveation model and the known fixation patterns of ASL receivers.

5.3 Estimating Average Intelligibility of an HRC

The analysis provided in Section 5.2 yields an objective estimate for every processed video by applying the appropriate linear mapping to the objective distortion measure. Within each experiment, the mean objective estimate and mean subjective intelligibility score can be computed for an HRC by averaging the scores across all the videos processed by that HRC. For example, experiment 1 consisted of 18 HRCs, each of which was applied to 6 reference videos, resulting in 6 compressed videos per HRC. The objective and subjective scores for a single HRC are computed as the average score of all 6 videos processed using that HRC. For experiment 1, this results in a collection of 18 averaged subjective and objective scores, each corresponding to a single HRC.

The number of data points in an experiment, after averaging across videos, is equal to the number of HRCs evaluated in that experiment, summarized in Table 3.1. Computing the four statistical metrics on these averaged data points determines the accuracy of the objective model when estimating the subjective intelligibility of an HRC. Averaging over each of the stories mediates any potential differences between them, such as variations in story complexity.

For all the experiments, the proposed CIM-ASL provides a more accurate estimate of the average subjective score for an HRC, as demonstrated by the statistically significantly lower RMSE values summarized in Table 5.1. For experiments 1 and 3, the CIM-ASL performs very well, having correlation coefficients r and ρ

near one and RMSE and OR values near zero. The performance of the CIM-ASL is slightly lower in experiment 2. However, noting the extremely poor performance of PSNR and VSSIM, experiment 2 provides the most challenging set of videos for accurately predicting subjective intelligibility. This experiment is most consistent with real usage scenarios, having both indoor and outdoor videos encoded at multiple bitrates using multiple encoding algorithms.

5.4 Robustness to Temporal Variation and Spatial Distortion Measures

Section 5.2 and Section 5.3 established that the CIM-ASL is a feasible model for estimating subjective intelligibility and significantly outperforms objective techniques typically applied to video quality assessment. This section demonstrates that the accuracy of the CIM-ASL is unaffected by alternative methods of computing the spatial error map and the amount of temporal variations. The high-level structure of the model therefore provides the proper framework for estimating intelligibility and the performance is robust to the low-level details. The individual components can be selected to suit the intended application.

Fine scale partitions of the experimental data more effectively highlight the impact on the CIM-ASL when applying different methods for computing the individual components. In this section, the objective model performance is computed separately on each of the video subsets from experiments 1 and 2 listed in Table 3.1 and not on the collective experimental data, as in the previous sections. The five video subsets from experiment 3 are excluded from this analysis because the videos within these individual subsets contain only a single frame rate and a single encoding algorithm operating at three levels of spatial quality. Any distortion measure that exhibits monotonic behavior can accurately predict intelligibility in these extremely homogeneous video subsets. The remaining four subsets from experiments 1 and 2 contain varying distortions that challenge the necessity of the individual components in the CIM-ASL.

For the video subsets used in this analysis, Section 5.4.1 develops a performance bound based on the minimum achievable RMSE for each subset. The following two sections demonstrate the robustness of the CIM-ASL when selecting the temporal variation and spatial distortion measures. Specifically, different methods of computing the temporal variation are evaluated in Section 5.4.2 to demonstrate the necessity of using an appropriate measure of the temporal variations of the spatial distortions. In Section 5.4.3, different pixel-based error measures applied within the proposed framework yield nearly identical performance.

5.4.1 Computing a performance bound on individual video subsets

Section 5.2 established that objective quality estimators, namely PSNR and VSSIM, cannot reliably estimate intelligibility. Because of the poor performance of these quality estimators, they cannot provide an adequate performance benchmark for the proposed CIM-ASL; a more meaningful benchmark is required. For a single video subset, a performance benchmark for the CIM-ASL is developed by training the model parameter values (α_k, β_k) using only one video subset.

As described in Chapter 4, the values for α_k and β_k used in the CIM-ASL are trained jointly on all four video subsets. The GA optimization, when trained on a single video subset, selects parameters α_k and β_k that achieve the lowest possible RMSE for only that subset, independent of the other three video subsets. Because these parameters are trained on only a single video subset, they will be overly sensitive to the distortions contained in that subset and will not be able to accurately estimate the effects of distortions contained in the remaining three subsets. Despite this, the performance achieved in a single video subset, when overtraining the model parameters to only that subset, serves as a benchmark for the CIM-ASL, when properly trained to avoid overfitting.

In all statistical metrics, the CIM-ASL performs statistically identical to the performance benchmark, as summarized in Table 5.2. The relatively low correlation coefficients, r and ρ, for the indoor videos from experiment 2 for both the benchmark and the proposed reflect the difficulties in estimating intelligibility for this set. Despite differences in the correlation coefficients, the estimation accuracy, quantified by RMSE, is consistent across all 4 video subsets. The following sections compare the performance of the intelligibility measure when varying the individual components.

5.4.2 Robustness to temporal variation measures

This section evaluates the performance of the CIM-ASL when using two different measures of temporal variations, each of which achieves statistically identical performance. The proposed model is also computed using only the spatial component, which results in a performance decrease compared to the full intelligibility measure. This demonstrates that an appropriate measure of the temporal variations of the distortions is required to accurately estimate intelligibility, but the framework is robust to the exact measure used.

Because the computation of the temporal variation component of the CIM-ASL, as computed in Eq. (4.5), relies on a percentile of temporal gradients, it can only be computed if the entire sequence is available. A real-time computable measure of temporal variation is necessary for implementing the intelligibility measure in a rate-distortion optimization scheme for real-time video encoding (cf. Section 7.1). Such a suitable real-time measure achieves identical statistical performance.

The *percentile-based* measure in Eq. (4.5) will almost always be non-zero (except in the case of constant distortion in every frame). Despite this, it does not always have a large contribution to the final intelligibility value, e.g., when the average distortion is high but does not significantly vary across frames. An empirical analysis of the percentile-based temporal variation measure reveals that it primarily contributes to the overall intelligibility measure when the average spatial distortion within a region increases by more than 30% between frames.

The real-time measure of temporal variation, denoted the *threshold-based* temporal variation, is the average gradient for only frames in which the spatial distortion has increased by more than 30% from the previous frame, after applying the median filter. Due to the median filter, this computation is not truly causal, but it can be computed instantaneously given a buffer for the filter. The relevant frames are identified as

$$N_{k,30\%} = \left\{ n \leq n_{current} \,\middle|\, \frac{\nabla d'_k(n)}{d'_k(n-1)} > 0.3 \right\} , \tag{5.1}$$

where $N_{k,30\%}$ is the subset of all frames n, up to frame $n_{current}$, for which the median-filtered spatial distortion, $d'_k(n)$, has increased by more than 30%. The subset of frames are region-specific, with $k \in \{face, hands, torso\}$, and may be disjoint for each of the regions. The threshold-based temporal variation computes the average of the gradient only in frames $N_{k,30\%}$ and is given by

$$tv_k \quad = \quad \frac{1}{\|N_{k,30\%}\|} \sum_{n \in N_{k,30\%}} \nabla d'_k(n) \tag{5.2}$$

Both the percentile-based and threshold-based measures of temporal variation compute the average gradient in a subset of the total video frames. The subset defined by the percentile-based measure can only be identified using the entire video sequence, while the subset defined by the threshold-based can be identified in real-time. The CIM-ASL using either of these measures of temporal variation is compared with only the spatial component of the model, which discards completely the temporal variation measure.

In terms of prediction accuracy (RMSE) and correlation (r, ρ), the purely spatial measure exhibits statistically significantly worse performance in only the outdoor videos from experiment 2 (RMSE = 0.602, $r = 0.590$, $\rho = 0.618$). The outdoor videos from experiment 2 contain the largest temporal variations of all the video subsets. Outdoor videos coded using the ROI encoding algorithms are subject to segmentation errors in coding, which leads to large temporal variations in the hand distortions. The MSE-based rate control algorithm allocates more rate to the macroblocks with higher motion activity. When the background activity increases, this coder necessarily allocates more rate to the background at the expense of the signer, resulting in increased distortions in the signer that create large temporal variations in the distortions.

The purely spatial measure performs statistically identical to the full model on the remaining three video subsets. These videos do not have significant temporal variations in distortions. As a result, computing only the spatial component of the CIM-ASL is sufficient for estimating intelligibility in these cases.

The causal computation of temporal variation performs statistically identically to the proposed percentile-based method in all four video subsets and all statistical metrics. The use of an appropriate measure of temporal variation is required for the accurate prediction of subjective intelligibility; however both the percentile-based and threshold-based methods perform equivalently.

5.4.3 Robustness to the spatial distortion measure

The CIM-ASL can be calculated using any spatial error measure, provided that the error can be pooled separately over each of the different regions. In addition to the MSE in contrast computed in Eq. (4.2), both the structural similarity (SSIM) index and the natural image contour evaluation (NICE) are evaluated as potential spatial distortion measures. The SSIM index, commonly used for image quality assessment, computes the similarity between a reference and a distorted image as a function of the mean, variance, and cross-correlation [86]. NICE was designed for image utility assessment, in contrast to image quality assessment, which makes it potentially more applicable for intelligibility assessment. NICE computes image utility by comparing the contours of the reference and test images, identifying errors as differences in the contours [65].

For both SSIM and NICE, the genetic algorithm optimization procedure, described in Section 3.1, identifies appropriate values for α_k and β_k for the corresponding spatial distortion error measures. These values are different for each error measure, but they maintain the high-level, linguistic features, namely, errors in the signer's face are most heavily emphasized. The performance of the CIM-ASL, when using three different spatial error measures, is statistically identical in all statistical metrics for three of the four video subsets. In the fourth video subset, the 10 FPS videos from experiment 1, the CIM-ASL using MSE in contrast performs statistically significantly better only in terms of the OR. Even in this case, using SSIM or NICE in the intelligibility measure perform very well, having low ORs of 0.089 and 0.111.

Fundamentally, these results demonstrate the importance of identifying the distribution of information within the video frame. Because ASL users are extracting information from the face and hands of the signer, properly weighting and pooling the errors in these regions is of primary importance. NICE applies a very different paradigm from pixel-based error measures, such as SSIM and MSE in contrast. Despite these differences, using NICE as the spatial error measure in the proposed model, which properly combines the region distortions, results in very good performance. Ultimately, the choice of a particular error measure is secondary to the selection of an appropriate pooling mechanism and can be chosen to suit the needs of a particular application.

5.5 Summary

This chapter demonstrated that the CIM-ASL accurately estimates the subjective intelligibility of ASL video and exhibits statistically significant improvements over computational models traditionally applied to measure video quality. Furthermore, the CIM-ASL properly models the distribution of information in an ASL conversation and is robust to the specific choice of spatial and temporal distortion measures.

Table 5.1: Comparison of statistical performance metrics for the CIM-ASL, PSNR, VSSIM, and foveated PSNR in two cases: estimating the intelligibility of individual videos in an experiment and estimating the intelligibility provided by the hypothetical reference circuit (HRC), averaged over source videos processed by that HRC. Bold values are statistically identical to the CIM-ASL, which is the top-performing model in all cases. In the case of foveated PSNR, italicized values are statistically significantly better than PSNR. For individual videos, the CIM-ASL performs statistically significantly better than PSNR and VSSIM in terms RMSE, r, and ρ in all three experiments. For HRCs, the CIM-ASL performs statistically significantly better than PSNR, VSSIM, and foveated PSNR in all three experiments in terms of RMSE and r.

		Individual Videos				HRC			
Measure	Video Subset	RMSE	OR	r	ρ	RMSE	OR	r	ρ
CIM-ASL	Exp. 1	**0.412**	**0.100**	**0.862**	**0.865**	**0.148**	**0.111**	**0.982**	**0.988**
	Exp. 2	**0.479**	**0.172**	**0.702**	**0.704**	**0.330**	**0.250**	**0.804**	**0.776**
	Exp. 3	**0.450**	**0.129**	**0.849**	**0.807**	**0.185**	**0**	**0.970**	**0.950**
PSNR	Exp. 1	0.603	**0.156**	0.671	0.657	0.480	0.500	0.760	0.695
	Exp. 2	0.637	**0.234**	0.320	0.331	0.511	**0.375**	0.390	0.382
	Exp. 3	0.579	0.214	0.734	0.688	0.387	0.533	0.867	**0.886**
VSSIM	Exp. 1	0.591	**0.156**	0.687	0.674	0.465	0.500	0.778	0.717
	Exp. 2	0.633	**0.250**	0.335	0.317	0.510	**0.375**	0.394	0.356
	Exp. 3	0.593	0.229	0.718	0.683	0.394	0.533	0.860	**0.889**
Foveated PSNR	Exp. 1	*0.517*	**0.156**	*0.772*	*0.773*	*0.361*	0.556	0.873	*0.897*
	Exp. 2	0.611	**0.203**	0.417	0.373	0.478	**0.375**	0.509	0.409
	Exp. 3	0.558	**0.200**	0.756	**0.743**	0.389	0.400	0.861	**0.911**

Table 5.2: Statistical performance metrics for training on a single video subset, which provides a best-case performance benchmark. When compared with the proposed CIM-ASL, none of the differences in the statistical metrics are statistically significant.

Training Case	Video Subset	RMSE	OR	r	ρ
Re-trained For Each Video Subset	Exp. 1 - 10 FPS	0.376	0.133	0.881	0.879
	Exp. 1 - 15 FPS	0.412	0.022	0.863	0.828
	Exp. 2 - Indoor	0.400	0.125	0.720	0.722
	Exp. 2 - Outdoor	0.426	0.063	0.820	0.787
CIM-ASL	Exp. 1 - 10 FPS	0.394	0.133	0.868	0.881
	Exp. 1 - 15 FPS	0.419	0.022	0.858	0.829
	Exp. 2 - Indoor	0.429	0.156	0.667	0.609
	Exp. 2 - Outdoor	0.432	0.125	0.814	0.819

CHAPTER 6

INTRODUCTION TO MOTION COMPENSATED VIDEO CODING

6.1 Introduction

One of the primary applications of the CIM-ASL is to develop an intelligibility optimized coder explicitly for ASL video. This chapter discusses general principles of motion-compensated, predictive video coding, which provides the necessary background information to understand the intelligibility optimized coder developed in Chapter 7. Of particular focus will be general principles of predictive video coding and operational rate-distortion optimization in the context of the H.264/AVC encoding standard.

6.2 Predictive Video Coding

The primary goal of compression is to remove redundant information from a signal. Predictive video coding removes redundancies from a video signal by first generating an accurate prediction of the pixels in a video frame, then subtracting this prediction from the video frame to form a residual signal. Accurate predictions yield small residual values that can be efficiently compressed. The residual signal is then *transform coded*. A block-based transform maps the residual signal into a transform domain, which provides additional energy compaction. At this stage, lossy compression is achieved via quantization of the transform coefficients. Additional lossless compression is achieved via entropy coding [63]. The block diagram in Figure 6.1 describes a predictive video coder. Each component in the coder is presented in more detail in the following subsections.

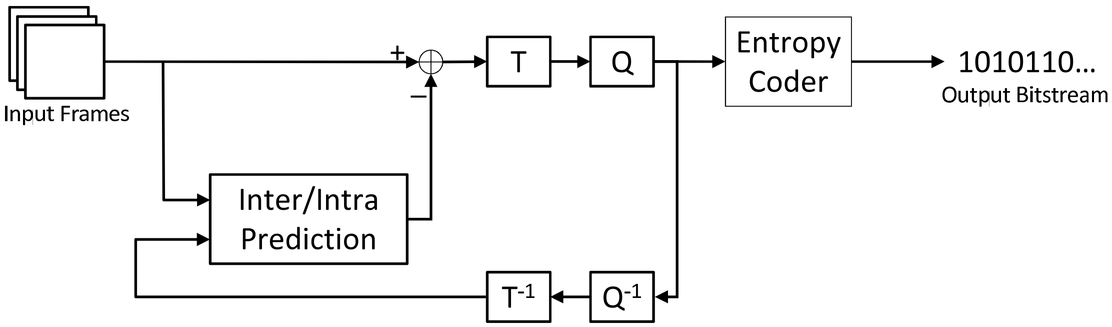


Figure 6.1: Block diagram of a predictive video coder. The blocks T and T^-1 refer to transform and inverse transform. The blocks Q and Q^-1 refer to quantization and inverse quantization.

Forming prediction residuals

Video frames are divided into 16×16 pixel macroblocks and predictions are formed for each macroblock. These predictions must be generated causally, i.e., the prediction for a macroblock can only be computed from previously coded data. In predictive video coding, the predictions can be formed by exploiting either the temporal correlation across frames in the video or the spatial correlation within a video frame.

Temporal predictions are used for *interframe* coding. In interframe coding, a macroblock is predicted from another macroblock in a previously coded frame. In the simplest case, the prediction is the co-located macroblock in the previous frame. This *frame differencing* prediction effectively subtracts the previous frame from the current frame. Unless there is no motion present in the video, the resulting residual signal will be significantly large. In H.264/AVC, this type of prediction mode is denoted `SKIP`.

More accurate predictions are formed using motion compensation techniques. Pixel-wise differences between frames in a video can be attributed to the motion of the objects within the frame, camera motion, or lighting/scene changes. With the exception of lighting and scene changes, these interframe difference are modeled as the translation of objects (corresponding to blocks of pixels) across time. To form an accurate temporal prediction for a macroblock, motion compensation techniques identify a motion vector, which defines the location of the best-matching macroblock in a prior frame. In this case, both the residual signal and the motion vector must be encoded into the bitstream. H.264/AVC allows for motion vectors at both macroblock and sub-macroblock resolutions. The following prediction modes in H.264/AVC denote interframe coding having motion vectors computed for blocks of size W×H: `INTER`16×16, `INTER`16×8, `INTER`8×16, `INTER`8×8, `INTER`8×4, `INTER`4×8, `INTER`4×4.

Spatial predictions are used for *intraframe* coding. In intraframe coding, a macroblock is predicted from spatially neighboring pixels in the same frame. Natural images exhibit a high degree of spatial correlation between nearby pixels. Spatial prediction modes exploit this correlation by extrapolating adjacent pixel values to form the prediction for the current macroblock. In H.264/AVC, the prediction can be formed either for the entire 16×16 macroblock or for 4×4 sub-macroblocks. These intraframe prediction modes are particularly useful for frames in which the temporal motion models fail, such as frames occurring at a scene change.

Transform and quantization

Block-based transforms applied to the residual signal achieve further energy compaction. H.264/AVC applies 2 different transforms, depending on the type of residual data. A 4×4 DCT-based transform is applied to sub-blocks in each 16×16 macroblock. A 4×4 matrix of DC coefficients, extracted from the DCT-based transforms applied to the macroblock, is additionally transformed using a 4×4 Walsh-Hadamard transform.

Mapping raw pixel values into residual signals and applying block-based transforms to the residual signals yields efficient energy compaction for natural videos. However, this amount of compression is insufficient for achieving the target bitrates of nearly all video communication and compression applications; lossless compression techniques are required. H.264/AVC applies scalar quantization of the form

$$Z_{ij} = \left\lfloor \frac{Y_{ij}}{Q_{step}} \right\rfloor, \qquad (6.1)$$

where Y_{ij} is the transform coefficient at spatial location (i, j), Z_{ij} is the quantized value, and Q_{step} is the quantization step size. In H.264/AVC, Q_{step} can take one of 52 values, indexed by QP$\in (0 \ldots 51)$. Q_{step} doubles for every 6 QP values and higher QP values correspond to more coarsely quantized coefficients (i.e. more compression).

Entropy coding

The final stage in the predictive video coder applies an entropy coder for lossless compression and for generating the binary bitstream representation of the video. The entropy coder exploits statistical redundancies in data symbols, where a symbol is composed either of the quantized transform coefficients in a macroblock or of the side information necessary for reconstructing the decoded coefficients (e.g., the motion vectors, prediction mode, or QP). H.264/AVC offers two different entropy coders. A variable-length coder (VLC) represents symbols with a binary sequence, where the length of the binary sequence varies with the probability of the symbol. Symbols that occur with high probability are represented with shorter sequences and fewer bits. Alternatively, an arithmetic coder yields compression that is near the theoretical optimum, at the cost of increased computational complexity for both the encoder and decoder. In both cases, H.264/AVC uses context adaptive entropy coding, where the models of symbol probabilities are adjusted according to the local spatial or temporal statistics.

6.3 Operational Rate-Distortion Optimization

Video coding standards specify a syntax for describing a compressed video as a sequence of bits (bitstream) and a method for decoding the bitstream and reconstructing the compressed video. The decoding methodology defines the set of admissible coding techniques, such as the type of spatial or temporal prediction modes, transforms, and entropy coders. This framework allows for significant flexibility in the design and operation of the encoder. The operation of the encoder is determined via the selection of encoding parameters, which are ideally chosen to minimize the distortion in the video for a given bitrate. An encoder that meets this criteria is considered rate-distortion optimal. This flexibility in the encoder operation ultimately allows for the development of a standard compliant H.264/AVC encoder that is optimized for ASL video, by choosing the CIM-ASL as the distortion criteria (cf. Chapter 7).

A set of rate-distortion optimal encoding parameters is defined as follows. In H.264/AVC, the rate and distortion in a macroblock is determined by the selection of motion vector, mode, and quantizer [90]. The motion vector and mode determine the prediction for a macroblock and are used to generate the residual, which is transformed and quantized. The problem of optimal encoder control becomes choosing a parameter combination $p_i \in P \equiv \{MV \times M \times QP\}$ for each macroblock X_i over all N blocks. These coding decisions will affect total rate, $R(X, p)$, and total distortion, $D(X, p)$. Given a rate constraint, R_{max}, the optimization finds p such that:

$$\min_{p \in P^N} D(X, p) \quad \text{subj. to } R(X, p) \leq R_{max} \tag{6.2}$$

This rate constrained optimization problem is made into an unconstrained problem by using the Lagrangian relaxation technique. This reduces the optimization in Equation (6.2) to:

$$\min_{p \in P^N} J(\lambda, X, p) = D(X, p) + \lambda R(X, p) \tag{6.3}$$

For a fixed λ, the solution to Eq. (6.3) that results in a realized rate, denoted R_t, is identical to the solution to Eq. (6.2) when $R_{max} = R_t$ [72].

The selection of the QP has the largest impact on the rate and distortion in a macroblock. Furthermore, accurate prediction, leading to small residuals, is desirable regardless of the quantization step size. Consequently, the QP for a macroblock is typically chosen prior to identifying the motion vector and mode for that macroblock [88]. The optimization then requires a two-step process: first, find the optimal QP values for each macroblock in the frame according to the Lagrangian cost for a fixed λ, second, find the optimal prediction (including motion vector and prediction mode) according to the Lagrangian cost for a fixed λ and fixed QP.

In H.264/AVC, the QP for the current block is coded as a delta offset from the QP for the previous block. Because of this, the additional rate required to encode large changes in QP can add non-trivial overhead to the bitstream, especially at very low rates. In order to model this dependency, a trellis is built in which each stage corresponds to a macroblock in a row and each node in a stage corresponds to a QP value [57,68]. The Viterbi algorithm is used to search for the path through the trellis that minimizes the Lagrangian cost, J, for a particular row. The algorithm then iterates over all rows in the frame. In terms of number of required Lagrangian cost calculations, this algorithm has a complexity of $O(Q^2 \times N)$, where there are Q possible QP values (52 for H.264/AVC), and N macroblocks in a frame.

The trellis search identifies the optimal QP for each macroblock in each video frame. However, the computational complexity of the trellis search prohibits its use in real-time video encoders. An alternative approach selects a single QP for the entire frame according to a functional model relating λ and the QP, given by the following, for H.264/AVC

$$\lambda = 0.85 \times 2^{\frac{QP-12}{3}}. \tag{6.4}$$

This model is empirically determined by applying the trellis search to a collection of natural video sequences and analyzing the set of optimal QP values [88, 90].

For a fixed λ and QP, the remaining encoding decisions include the selection of the prediction mode and, in the case of interframe coding, the optimal motion vector. As shown in the Section 6.2, the space of possible prediction modes is sufficiently small such that the Lagrangian cost associated with each mode can be explicitly computed. The encoder simply selects the mode with the lowest cost.

6.4 Summary

Predictive video coding achieves efficient compression of video signals by exploiting temporal and spatial correlations in the video to generate predictions for each coded block. The predictions are subtracted from the block to be coded to form a residual signal, which is transformed, lossy compressed through quantization, and losslessly compressed using an entropy coder. Operational rate-distortion optimization techniques are applied in order to identify quantization step sizes and prediction modes that result in the minimum achievable distortion for a given rate constraint.

CHAPTER 7

APPLYING THE CIM-ASL TO OPERATIONAL RATE-DISTORTION-COMPLEXITY OPTIMIZATION

7.1 Introduction

The proposed CIM-ASL accurately estimates subjective intelligibility and can be applied to optimize ASL-specific systems. Incorporating the CIM-ASL into a rate-distortion optimization procedure for H.264/AVC creates a closed-loop encoding system designed for ASL video, denoted the intelligibility optimized encoder. For videos encoded by this system, the rate is optimally distributed between the relevant portions of the signer (denoted the regions-of-interest or ROI) and the background, without the need for heuristics. This intelligibility optimized coder provides significant bitrate reductions compared to a general purpose H.264 encoder (x264 [2]) and a foveated video coder designed for ASL video [4].

The CIM-ASL is also applied to reduce the computational complexity of the intelligibility optimized encoder. Three encoding parameters are developed that allow the encoder to allocate computational resources differently between the ROI and non-ROI. The CIM-ASL is included in an encoder parameter optimization by modifying a fast, offline distortion-complexity optimization algorithm, resulting in parameter selections that demonstrate excellent rate-distortion-complexity performance.

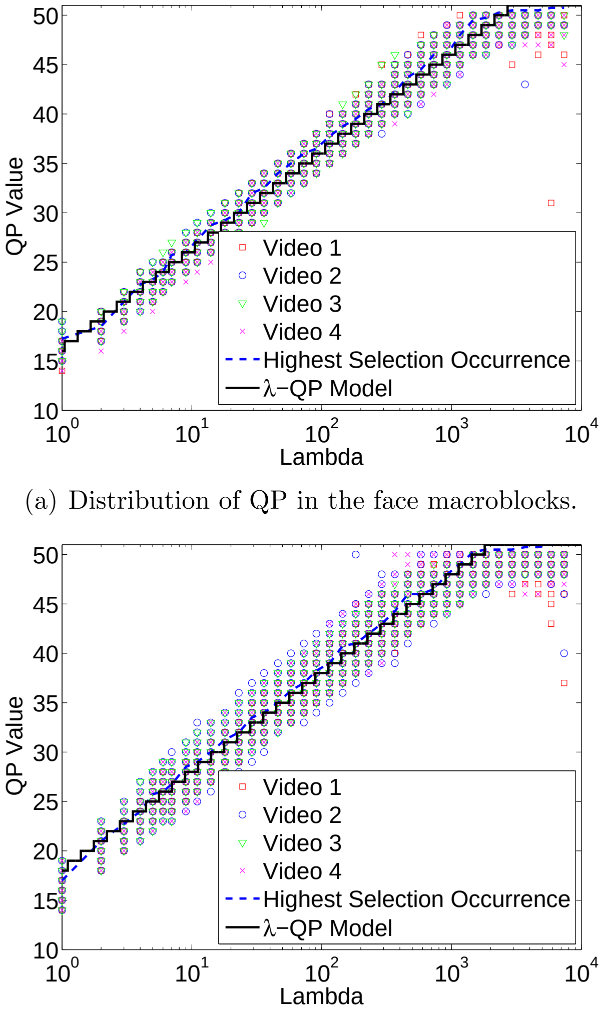
The remainder of this chapter is organized as follows. The intelligibility optimized encoder is described in Section 7.2. The rate-distortion performance of the intelligibility optimized encoder is demonstrated in Section 7.3. Section 7.4

describes the complexity allocation encoding parameters. The encoding speed improvements when using these parameters are demonstrated in Section 7.5. Finally, in Section 7.6, an offline distortion-complexity optimization procedure incorporates the CIM-ASL to efficiently identify optimal operating points in the joint rate-distortion-complexity space.

7.2 Intelligibility Optimized Encoder

The trellis-based rate-distortion optimization procedure described in Chapter 6 selects the optimal quantization parameter (QP) given a fixed Lagrangian parameter (λ). While this cannot be applied in a real-time encoding scenario, applying this algorithm to a collection of ASL videos affords the development of a λ-QP model unique to the CIM-ASL. The trellis-based optimization is applied to four different ASL videos, which vary in the amount of background activity and vary in the size of the regions-of-interest relative to the frame size. As described in the previous chapter, this optimization procedure minimizes the joint rate-distortion (R-D) cost $D + \lambda R$. For the intelligibility optimized coder, the distortion measure is defined according to the the CIM-ASL as intelligibility distortion D_{Intell} from Eq. (4.9).

In the intelligibility optimized encoder, each macroblock in a video frame is labeled as belonging to one of the relevant regions (face, hands, torso, new background, and sustained background) using the algorithms described in Chapter 4. The contribution to D_{Intell} of local distortions varies depending on the region type. Recall that $\alpha_{face} = 1.6$ and $\alpha_{hand} = 0.5$. Consequently, the optimization procedure will select a smaller QP for *face* macroblocks than for *hand* macroblocks (i.e., the face is less heavily quantized than the hands). This is illustrated in Figure



(a) Distribution of QP in the face macroblocks.

(b) Distribution of QP in the hand macroblocks.

Figure 7.1: Optimal QP values, selected via trellis search, versus λ. Ninety percent of the macroblocks were coded with QP values among the clouds of points. The *Highest Selection Occurrence* line is the QP value that is selected most often by the optimization. The empirical λ-QP model is also plotted.

7.1, which plots the distribution of the optimal QP values selected for the face macroblocks and hand macroblocks. The same trend holds for the other regions-of-interest.

Real-time encoding of ASL video using the intelligibility optimized coder requires a model that specifies the QP for each macroblock, given a fixed Lagrangian parameter λ. An analysis of the optimal QP values over a range of tested λ values reveals a functional relationship between λ, QP, and the spatial distortion weight α. This relationship is given by

$$2^{\frac{QP(X)-12}{3}} = \frac{\lambda}{0.65\alpha(X)}, \tag{7.1}$$

where $QP(X)$ is the quantization parameter for macroblock X and $\alpha(X)$ is the weight for macroblock X, determined by the macroblock's segmentation label. For face, hand, torso, and new background macroblocks, the weights are given by $\alpha_k = (1.6,\ 0.5,\ 0.2\,1)$ as listed in Table 4.1. For sustained background macroblocks, $\alpha = 10^{-3}$ to avoid numerical instabilities. Figure 7.1 plots the QP selected by the model versus the empirical distribution of QPs, computed from the collection of optimally coded ASL videos. Note that the model corresponds very closely with the most commonly occurring QP value and this model holds across the different regions-of-interest.

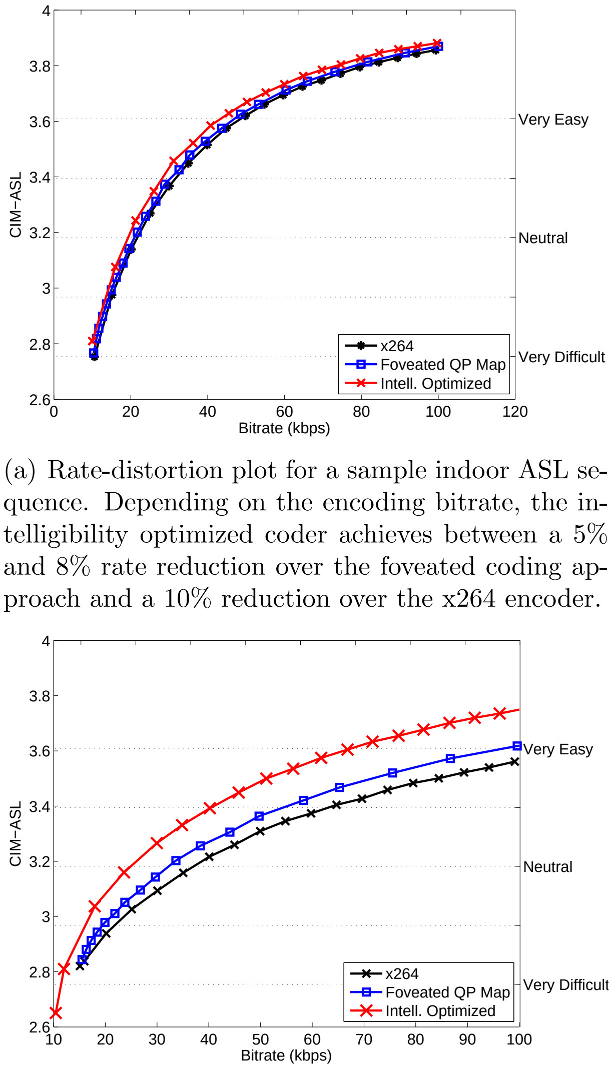
For increasing values of α, corresponding to increasing importance for intelligibility, the quantization step size will decrease. As a result, more important regions in the video frame, as defined by D_{Intell}, are assigned a lower quantization step size and are allocated more rate. Given a quantization step size for a macroblock, the intelligibility optimized coder identifies the remaining coding decisions (i.e., macroblock coding mode and motion vector) that minimize the R-D cost.

7.3 Rate-Distortion Performance Results

The intelligibility optimized coder is compared against x264 [2] and an ASL-specific, foveated encoding technique [4]. As illustrated in Figure 7.2, the proposed algorithm achieves substantial bitrate reductions at fixed levels of intelligibility over both the x264 encoder and the foveated encoder. When compared to x264, the proposed achieves reductions of 10% on indoor videos and between 26% and 42% on outdoor videos. When compared to the foveated encoding algorithm, the proposed achieves reductions between 5% and 8% for indoor videos and between 19% and 31% for outdoor videos. Larger rate reductions are obtained for outdoor video sequences, because of the higher level of background activity. The x264 and foveated encoders allocate a significant amount of rate to the background region. The proposed encoder only allocates rate to macroblocks containing the face, hands, torso, and new background, maintaining high intelligibility at low rates at the cost of high distortion in the sustained background. At fixed bitrates, the proposed coder produces video with higher subjective intelligibility than the x264 coder, as rated by fluent signers [20].

7.4 ROI-based Complexity Allocation Parameters

Bandwidth and computation time are the two major resource constraints for a real-time video communication system. As demonstrated in the previous section, the intelligibility optimized encoder addresses the problem of constrained bandwidth by providing maximal intelligibility for a fixed rate constraint. This rate-distortion performance can be realized because bits are allocated to the regions-of-interest according to their relative importance. A similar approach is used for constraints



(a) Rate-distortion plot for a sample indoor ASL sequence. Depending on the encoding bitrate, the intelligibility optimized coder achieves between a 5% and 8% rate reduction over the foveated coding approach and a 10% reduction over the x264 encoder.

(b) Rate-distortion plot for a sample outdoor ASL sequence. Depending on the encoding bitrate, the intelligibility optimized coder achieves between a 19% and 31% rate reduction over the foveated coding approach and between 26% and 42% reduction over the x264 encoder.

Figure 7.2: Rate-distortion plots for x264 [2], a foveated video coder [4], and the intelligibility optimized coder, which incorporates the CIM-ASL. The left y-axis provides the score given by the CIM-ASL. The right y-axis provides the subjective rating categories corresponding to the CIM-ASL values. For a fixed level of intelligibility, rate reductions are larger for outdoor sequences.

on the computational complexity of the encoder. This section introduces three encoding options that allocate computational complexity to the regions-of-interest.

The intelligibility optimization is implemented as a modification to x264, allowing for the use of all the encoding parameters available to x264, the selection of which provides a trade-off between encoding complexity and R-D performance. Specifically, four encoding parameters available in the x264 are varied to achieve different R-D-C operating points: sub-pixel motion estimation (`subme`); reference frames (`ref`); partition size (`part`); and entropy coding and quantization (`trellis`). The `subme` has 7 options corresponding to the number of iterations for half-pel and quarter-pel motion estimation. Additionally, `subme` controls whether the R-D cost is fully evaluated in the pixel domain or estimated in the transform domain. A maximum of 16 reference frames can be specified using `ref`. Eight different `part` options specify the partition size from 4×4 and above for intra (I), predictive (P) and bi-predictive (B) macroblocks [82]. The `trellis` parameter has four options that include uniform quantization with and without context adaptive arithmetic coding (CABAC) (options 1 and 0); and two schemes that use CABAC and Djikstra's algorithm for finding the quantization for a block of DCT coefficient such that the overall R-D cost is reduced (options 3 and 4). A vector of parameter options is defined as *parameter settings*. An example of a parameter setting is (`subme`=0, `ref` = 1, `part`=1, `trellis`=0), which has the lowest computational complexity. In this section, the average encoding time is used as a measure of complexity.

Three novel encoding parameters are added to the x264 encoder that allow the encoding complexity to vary on a per-block basis, depending on whether the block belongs to ROI or not. In H.264/AVC, as many as 12-15 different partitions are available for a given macroblock (cf., Chapter 6). The first ROI complexity parameter, `nonROI-part`, restricts the partitions used by the encoder for the background blocks. Since distortions in background macroblocks do not contribute to the CIM-ASL, background macroblocks can be encoded with very little rate (and consequently, very high distortion). Motivated by this, the encoder is modified to have two sets of available partition types, one for the ROI blocks and other for the non-ROI blocks. For ease of integration into the pre-existing encoder structures, the `nonROI-part` has the same 8 options as `part`. This allows the search for partitions in background macroblocks to be limited to only the coarsest partitions while still enabling the finer partitions for the relevant blocks.

The second parameter, `ROI-subme`, has the same 7 options as the `subme` parameter and is applied to the ROI, while the `subme` option is applied to the non-ROI. In addition to varying the complexity of sub-pixel motion estimation, the `subme` also varies the accuracy and complexity for R-D cost computation. The highest `subme` option computes the actual R-D cost by encoding and decoding a macroblock, while the lowest option only estimates the R-D cost from the coded macroblock. The `ROI-subme` together with `subme`, allows the encoder to use the fast R-D cost estimate on non-ROI blocks while computing the accurate R-D cost and using high complexity sub-pixel motion estimation for the ROI blocks.

The third ROI parameter addresses the complexity of the motion search. In motion-compensated video coding, motion search comprises a significant portion of the total encoding time. To speed up the motion search, a ROI-based motion search parameter `ROI-MS` is included that specifies a potentially different motion search method for the ROI and non-ROI macroblocks. The `ROI-MS` uses the following three fast motion search methods provided by x264 in the order of increasing complexity: diamond (DIA), hexagon (HEX) and uneven multihexagon search (UMH) [2]. The `ROI-MS` uses only the DIA search for the background and has the following 8 options $(1, \ldots, 8)$ corresponding to the motion search in (face, hand/torso, background) regions: (DIA, DIA, DIA), (HEX, DIA, DIA), (UMH, DIA, DIA), (HEX, HEX, DIA), (HEX, UMH, DIA), (UMH, HEX, DIA), (UMH, UMH, DIA), and (UMH, UMH, UMH).

For each of the encoding parameters, higher options often corresponds to higher complexity. For example, a value of `part` = 8 is the most complex and enables the encoder to search over of all possible macroblock partitions. Conversely, a value of `part` = 1 restricts the search to only the coarsest partitions but offers the lowest complexity. The lower complexity options can increase the speed of the encoder but can can result in higher distortions at fixed bitrates.

7.5 Performance of the ROI-based Complexity Allocation Parameters

Each of the three additional ROI complexity parameters is evaluated explicitly in terms of its affect on the R-D-C performance. Both the standard implementation of the x264 encoder and the intelligibility optimized encoder serve as performance benchmarks. In each of these benchmark cases, 8 ASL test videos are encoded at 8 fixed bitrates, ranging from 5 to 75 kbps, using the highest complexity option for each of the 4 parameters described in Section 7.4, without any of the ROI complexity modes enabled. For each fixed bitrate, the CIM-ASL and the encoding time are averaged over the set of 8 test videos.

Using the highest complexity parameter options guarantees that the R-D performance will be optimal, at the expense of average encoding time. The intelligibility optimized encoder demonstrates improved performance over the x264 encoder in terms of both R-D and distortion-complexity (D-C). For the same level of intelligibility, the intelligibility optimized encoder achieves a reduction in rate from the x264 encoder between 10% and 28% and a reduction in encoding time between 15% and 25%, depending on the encoding bitrate.

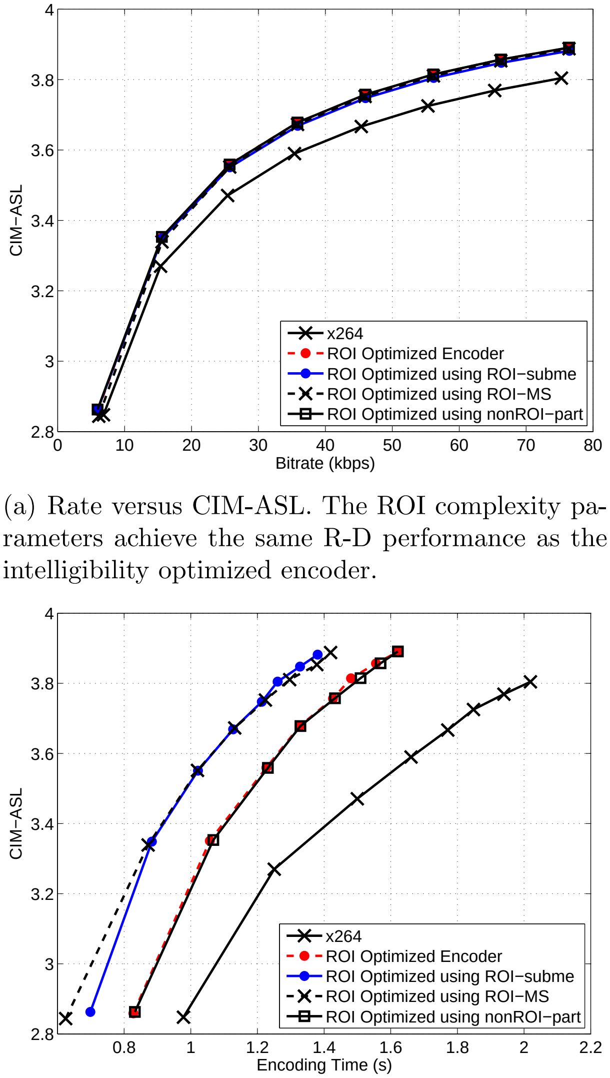
To achieve the same level of intelligibility, x264 must operate at a higher bitrate, because it allocates rate to the non-ROI and the ROI indiscriminately, whereas the ROI encoder allocates rate almost entirely to the ROI. The complexity gains provided by using the intelligibility optimized encoder can be attributed to high distortion in the non-ROI. When using the CIM-ASL for computing R-D cost, distortions in the non-ROI do not contribute to the score. As a result, the encoder is making encoding decisions that minimize the bitrate in the non-ROI. By design, the x264 encoder (and, consequently, the intelligibility optimized encoder), applies several heuristics to quickly encode a macroblock at very low rates, selecting only coarse macroblock partitions or skip modes, in which the co-located macroblock in the previous frame is copied without performing a full motion search.

Each of the three proposed ROI-based complexity allocation parameters are evaluated independently in terms of their impact on the R-D-C performance of the intelligibility optimized encoder. For each test case, the encoding parameter settings are chosen such that the ROI is encoded with the highest complexity options and the non-ROI is encoded with the lowest complexity option. Specifically, the three test cases are: `ROI-subme` = 6, `subme` = 0; `ROI-MS` = 7 (UMH for ROI, DIA for non-ROI); and `nonROI-part` = 8, `part` = 1. The other x264 parameters described in Section 7.4 are all set to their highest complexity.

As illustrated in Figure 7.3(a), applying any of the ROI complexity options results in a negligible effect on the R-D performance. Each of the three cases performs nearly identical to the intelligibility optimized encoder when using the highest complexity settings. Figure 7.3(b) illustrates the average complexity gains achieved by the ROI complexity options. The `ROI-subme` and `ROI-MS` options provide similar speed improvements of approximately 16%. In each of these test cases, the complexity is reduced because of the integer-pixel motion estimation (`subme`) and coarse motion search (`ROI-MS`) performed on the non-ROI. Somewhat surprisingly, the `nonROI-part` yields no speed improvement. Because x264 efficiently eliminates many of the candidate partition sizes, further restricting the possible partition size available for non-ROI blocks does not significantly reduce the complexity of the system.

7.6 Joint Rate-Distortion-Complexity Optimization using DPSPA and the CIM-ASL

The H.264/AVC coding standard only specifies the operation of the decoder, leaving virtually infinite flexibility in the operation of the encoder. The set of encoding parameters discussed in Section 7.4 made available to the encoder determine the achievable bitrate, distortion, and complexity. Ideally, a video encoder will select the parameter setting which results in a compressed video that meets the target rate and complexity constraints while minimizing the distortion, i.e. operates on the convex hull of the R-D-C surface. To find the set of R-D-C convex hull parameter settings, an exhaustive search is required over all parameter settings. For the



(a) Rate versus CIM-ASL. The ROI complexity parameters achieve the same R-D performance as the intelligibility optimized encoder.

(b) Encoding time versus CIM-ASL. The `ROI-subme` and `ROI-MS` parameters result in a 16% reduction in encoding time.

Figure 7.3: The R-D-C space for 5 different encoding scenarios. The x264 encoder and the intelligibility optimized encoder, each running with the highest complexity settings, provide benchmark performance levels. The three ROI parameters are compared against the benchmarks.

parameter settings defined here, an exhaustive search requires 1,605,632 encodings per video per bitrate ($7 \times 16 \times 8 \times 4 \times 8 \times 7 \times 8$). Because it is impractical to perform an exhaustive search of this R-D-C space, fast methods for choosing the appropriate set of encoding parameters must be employed.

The dominant parameter setting pruning algorithm (DPSPA) [82] is applied to determine close to optimal parameter settings without performing a full search. DPSPA is a fast offline algorithm that uses significantly fewer encodings compared to an exhaustive search to estimate the D-C convex hull. For a fixed bitrate, DPSPA provides a collection of parameter settings which correspond to operating points lying approximately on the D-C convex hull, as illustrated in Figure 7.4. These points are nearly optimal in terms of their D-C performance; for a fixed complexity constraint, the resulting distortion is minimized. Applying the algorithm over a range of target bitrates approximates the full R-D-C convex hull. Given a target bitrate and complexity constraint, the optimal parameter setting can be chosen immediately, effectively creating a lookup table which provides the appropriate parameter setting for each target rate and complexity.

Three combinations of training and test sets are created from a collection of 8 indoor ASL videos, filmed on a static background, and 8 outdoor ASL videos, filmed on a busy street. The segmentation into ROI and non-ROI is performed offline for each video. The three cases correspond to training and testing on only the indoor videos, only the outdoor videos, and on both the indoor and outdoor videos. The DPSPA algorithm is applied to a set of four training ASL videos and four test ASL videos each having 176×144 frame resolution, 200 frames and a frame rate of 15 fps. These experiments are conducted on a Windows XP PC having a 2.01 GHz AMD processor and on an HTC TyTN II cell phone having a

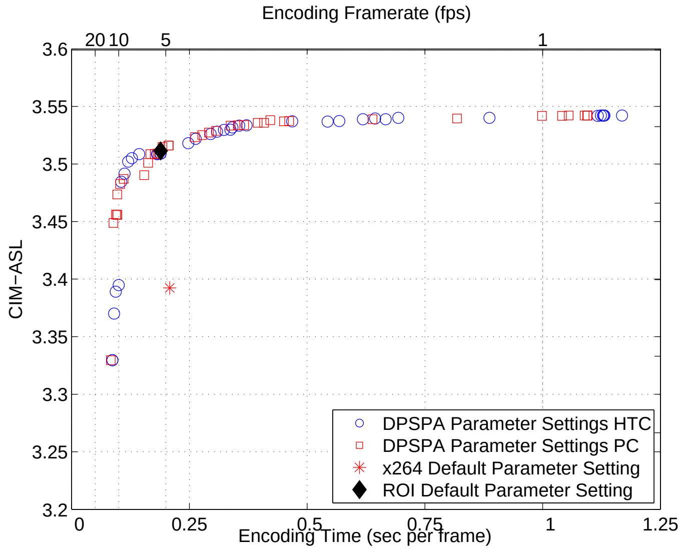


Figure 7.4: CIM-ASL vs. encoding time (lower y-axis) and the corresponding encoding frame rate (upper y-axis) for the outdoor ASL training set at 30 kbps, running on the HTC TyTN II cell phone. DPSPA parameter settings obtained on either the PC or the cell phone have similar performance on the cell phone.

400 MHz Qualcomm MSM7200 ARMv6 processor.

The x264 default parameter setting is the vector $(\texttt{subme} = 5, \texttt{ref} = 1, \texttt{part} = (\mathrm{P}8\times8, \mathrm{B}8\times8, \ \mathrm{I}8\times8, \mathrm{I}4\times4), \texttt{trellis}= 1)$. This parameter vector corresponds to high complexity sub-pixel motion estimation; use of larger number of macroblock partitions; one reference frame; and the use of the context adaptive arithmetic coder (CABAC) with uniform quantization. The default settings do not use any of the region-based complexity optimization options.

The DPSPA algorithm is executed for 15, 30 and 60 kbps. The DPSPA parameter settings are applied to the test set of ASL videos to obtain the average encoding speed improvement and change in intelligibility of DPSPA parameter setting over the x264 default parameter setting. Let $CIM(p)$ and $C(p)$ correspond to the intelligibility distortion and encoding time of a parameter setting p. The change in intelligibility is defined as $\Delta CIM = CIM(default) - CIM(DPSPA)$ and speed gain $= \frac{(C(default) - C(DPSPA))}{C(default)} \times 100$.

As demonstrated in Tables 7.1 and 7.2, the DPSPA parameter settings provide average speed improvements of approximately 45% on the PC and 52% on the cell phone with little decrease in intelligibility. A difference of approximately 0.2 corresponds to a statistical change in subjective intelligibility score (cf. Figure 7.2). Therefore, the average decreases in intelligibility shown in Tables 7.1 and 7.2 will not significantly reduce the perceived intelligibility.

Tables 7.1 and 7.2 demonstrate that for both the PC and cell phone encoding scenarios, the largest speed increase is obtained on the outdoor test videos. Because these videos were filmed on a busy street, the level of background activity is significantly high. The x264 encoder must spend computational resources encoding these non-ROI, whereas the intelligibility optimized encoder can use very coarse, low-complexity parameter options. The overall speed improvement of the intelligibility optimized encoder depends on the relative level of activity in the non-ROI.

Tables 7.1 and 7.2 compare the performance against the x264 default parameter settings, which were chosen heuristically by its developers to provide good R-D performance at a reasonable encoding speed. This default parameter setting, applied to the intelligibility optimized encoder, is denoted the ROI default parameter setting. The ROI default parameter setting results in an overall D-C performance that lies on the DPSPA points, as illustrated in Figure 7.4. While the ROI default parameter setting performs better than some encoder parameter settings for the corresponding encoding speed, it is not fast enough for real-time performance. DPSPA provides points which allow the encoder to run at or above

Table 7.1: CIM-ASL difference (Δ CIM-ASL) and speed gain of DPSPA parameter setting over the x264 default parameter setting on a 2.01 GHz PC for different pairs of training and test videos. Negative value for Δ CIM-ASL indicates a lower intelligibility for DPSPA.

Bitrate	Indoor		Outdoor		Indoor & Outdoor	
(kbps)	Δ CIM-ASL	speed gain	Δ CIM-ASL	speed gain	Δ CIM-ASL	speed gain
15	≈0	31.2%	-0.03	43%	0.01	40.8%
30	0.05	41.3%	-0.05	48.2%	0.01	45.8%
60	0.03	45%	-0.07	54.4%	-0.02	50.7%
Average	0.03	39.2%	-0.05	48.5%	≈0	45.8%

10fps, the nominal limit for full ASL conversations. [39]

DPSPA provides a collection of parameter settings which are appropriate for the specific test device on which it is run. While DPSPA can be executed on the cell phone platform, it is useful to investigate if the parameter settings generated on the PC can still approximate the D-C convex hull on the cell phone. The set of encoding parameters computed by DPSPA when run on the PC is applied to the test videos encoded on the cell phone. Despite differences in the exact parameter settings chosen, the PC-generated settings perform very close to the cell phone-generated settings. Figure 7.4 illustrates the D-C curves for the outdoor test videos at 30 kbps, comparing both collections of parameter settings. In this case, the testing required for DPSPA, and the resulting convex hull lookup table, can be generated on the PC and simply ported to the phone without any loss in performance.

Table 7.2: CIM-ASL difference (Δ CIM-ASL) and speed gain of DPSPA parameter setting over the x264 default parameter setting on a HTC TyTN II cell phone for different pairs of training and test videos.

Bitrate	Indoor		Outdoor		Indoor & Outdoor	
(kbps)	Δ CIM-ASL	speed gain	Δ CIM-ASL	speed gain	Δ CIM-ASL	speed gain
15	≈0	43.6%	-0.01	49%	0.08	49.7%
30	≈0	45.7%	≈0	55%	0.09	53.8%
60	≈0	48%	-0.01	62.1%	0.04	54.5%
Average	≈0	45.8%	-0.01	55.4%	0.07	52.7%

On the PC, the DPSPA often picks all `ROI-MS` options, while on the cell phone (HEX, UMH, DIA) is preferred over (UMH, HEX, DIA) and (UMH, DIA, DIA) options. This shows that on a cell phone, better intelligibility-complexity trade-off is obtained by using higher complexity UMH for the hand macroblocks instead of the face macroblocks. Because the location of the face does not vary significantly between frames, a fast motion search algorithm (HEX) is sufficient for identifying the appropriate motion vectors. The signer's hands movements are much wider over the frame, and accurate motion vectors are identified using a higher complexity motion search (UMH).

As parameter settings are generated from highest to lowest complexity by DPSPA, the `subme` option (associated with the non-ROI) first decreases from its highest to lowest option while the `ROI-subme` is retained at its highest option. Therefore, DPSPA appropriately reduces encoding complexity by choosing `subme` options that favor lower distortion of the ROI over the non-ROI, and lower complexity in the non-ROI versus the ROI.

The x264 default parameter setting is compared with the DPSPA parameter setting having comparable CIM-ASL performance for the three bitrates on the cell phone. Each of the DPSPA parameter settings allow the encoder to operate at or above 10fps. The DPSPA parameter settings include `trellis` = 2 at 15 kbps while using `trellis` = 1 and `trellis` = 0 for 30 kbps and 60 kbps. When `trellis` = 2, the encoder uses trellis quantization for the best R-D performance. At higher bitrates, when the intelligibility is high, DPSPA selects CABAC without trellis quantization (`trellis` = 1) and the less efficient CAVLC entropy coder (`trellis` = 0). The x264 default parameter setting uses `trellis` = 1 at all bitrates. For `ROI-subme` and `ROI-MS`, DPSPA picks integer pixel motion estimation and the use of all DIA, which are both lower in complexity compared to the default options of `subme` = 5 and HEX motion search, respectively.

7.7 Summary

This chapter demonstrated the rate-distortion-complexity performance of the intelligibility optimized encoder. A functional relationship between λ and QP provides rate-distortion performance (where distortion is measured as CIM-ASL) that significantly outperforms a general purpose video coder and a foveated video coder. The three ROI complexity allocation encoding parameters result in 16% speed improvement. Finally, the DPSPA effectively identifies optimal operating points in the joint rate-distortion-compleixty space.

CHAPTER 8

APPLYING THE CIM-ASL TO USER PREFERENCES IN THE QUALITY-INTELLIGIBILITY TRADE-OFF

8.1 Introduction

The coder developed in Chapter 7 provides a fully closed-loop method for optimizing the CIM-ASL, yielding videos having maximal intelligibility given an encoding constraint. The intelligibility optimized encoder achieves bitrate reductions by heavily distorting the background video region, while maximizing the fidelity of the signer. This is in contrast to the MSE optimized encoder described in Chapter 6, which nominally provides consistent levels of distortion across the entire frame (aiming to optimize the aesthetic quality of the video), but is unable to produce intelligible video at low bitrates. A subset of participants in the subjective experiments described in Chapter 3 qualitatively reported distractions due to heavily distorted backgrounds, even when they considered the videos to be intelligible. Allowing the user to adjust the level of background distortion addresses this problem, but lowering the distortion in the background region necessarily increases the distortion in the signer and can lead to an unintelligible video.

The quality optimized coder provides video that does not suffer from extremely large background distortions but may not provide sufficiently intelligible video to the user. The intelligibility optimized coder aims to provide the most intelligible video, but yields potentially distracting distortions away from the signer. This trade-off is denoted the quality-intelligibility trade-off. Ideally, the coder must adapt to both the available resources (e.g., encoding bitrate) and to the user preferences, in order to provide both intelligible video and a high quality of experience for the user (i.e., operate at the appropriate point on the quality-intelligibility trade-off).

This chapter presents computational techniques to suggest optimal operating points that can increase the aesthetic quality of the video while maintaining the intelligibility of the ASL communication. Fluent ASL users evaluate these potential operating points in a paired comparison experiment. Section 8.2 describes how the intelligibility optimized encoder is modified to account for user preferences, providing a parameter that controls the degree to which intelligibility is emphasized over quality. This modification creates a method to maximize the CIM-ASL subject to the user's desired level of quality. The computational performance of the modified coder suggests potential operating points in the quality-intelligibility trade-off, which are discussed in Section 8.3. A paired comparison experiment is conducted to rank the potential operating points and to identify user preferences in the quality-intelligibility trade-off. A detailed description of the experimental methodology is given in Section 8.4. The experimental results, summarized and discussed in Section 8.5, demonstrate that the optimal operating points vary with use demographics, supporting the need for a user-specified trade-off between intelligibility and quality.

8.2 Varying ROI Priority to Achieve a Quality-Intelligibility Trade-off

The intelligibility optimized and quality optimized encoders represent two encoding extremes, either allocating all the rate only to the signer or distributing the rate evenly among every macroblock. When optimizing strictly for intelligibility, the rate allocated to the background is minimized independent of the resulting distortion, creating severe compression artifacts in the background macroblocks. Quality optimized video provides similar levels of distortion across the entire frame, eliminating extreme distortions in the background. However, when optimizing strictly for quality, distortions in the signer can lead to unintelligible video. These two encoding extremes alone are incapable of accommodating the preferences of ASL users and maintaining intelligible video.

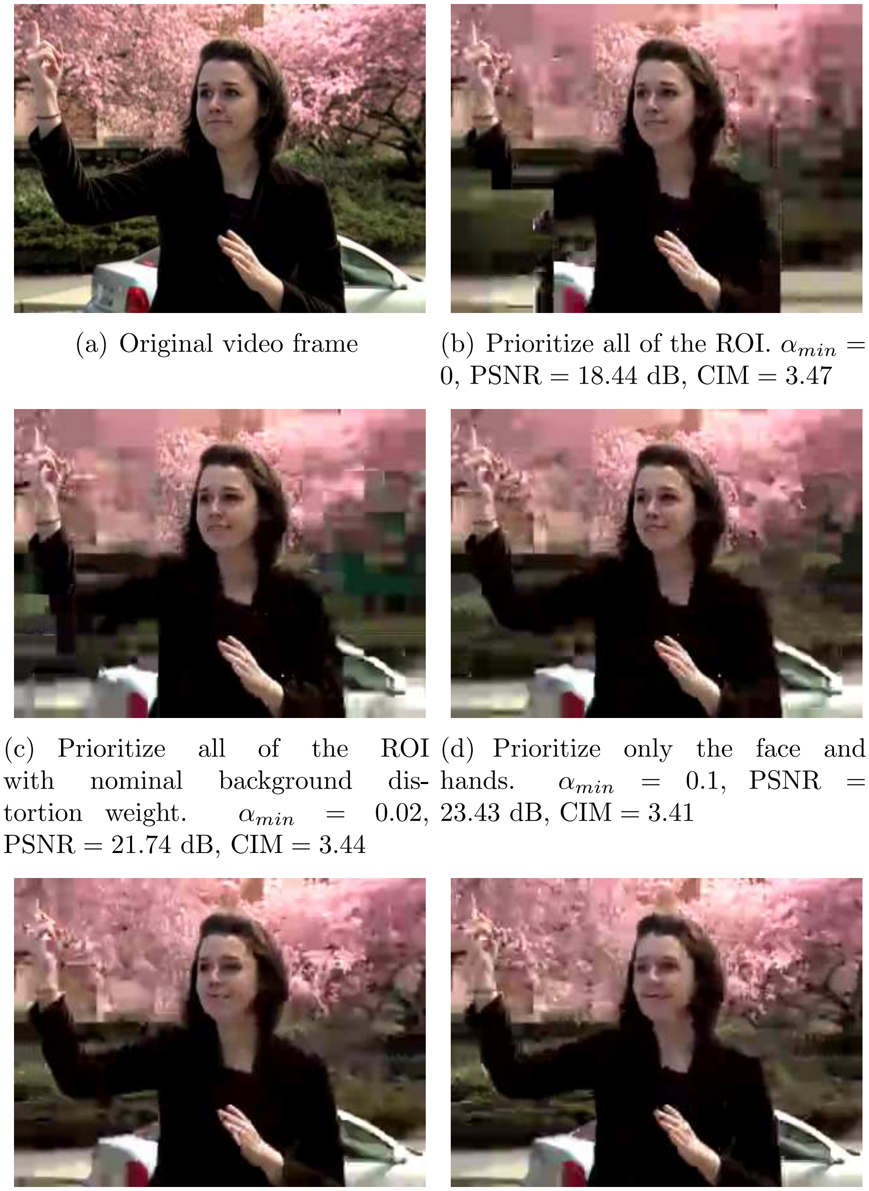
The user-specified quality-intelligibility encoding trade-off parameter is denoted α_{min} and specifies the minimum weight to be applied to all macroblocks in the frame. Specifically, if the weight α_k of any region (including the signer's face, hands, torso, or background) is less then α_{min}, then the weight α_k is changed and set equal to α_{min}. This provides a mechanism to increase the quality in the background, while guaranteeing that the background distortion weight is never higher than the distortion weights for the signer's face, hands, or torso.

Modifying α_{min} controls the degree to which the regions of interest (ROIs) are prioritized over the rest of the frame. A region is considered prioritized if its corresponding distortion weight is larger than α_{min}. A prioritized region will have lower distortion, on average, than the rest of the frame. For example, the intelligibility optimized encoder corresponds to $\alpha_{min} = 0$; the entire ROI (face, hands, torso) is given priority over the background. When $\alpha_{min} = 0.1 = \alpha_T$, the distortions in the background and the torso are weighted equally, and only the face and hands are prioritized because of their higher distortion weight. As α_{min} increases, only the most important macroblocks are prioritized. At the extreme, when $\alpha_{min} \geq \alpha_F$, all of the regions are weighted equally and the encoder behaves as the quality optimized encoder.

To illustrate, consider a sample ASL video, recorded in an outdoor setting with a highly active background and encoded at 55 kbps with different values of α_{min}. Five values for α_{min} are selected to emphasize different operating points and are evaluated in the paired comparison experiment: $\alpha_{min} = 0$ prioritizes the entire ROI, $\alpha_{min} = 0.02$ prioritizes the entire ROI and provides a nominal amount of rate to the background, $\alpha_{min} = \alpha_T = 0.1$ prioritizes only the signer's face and hands, $\alpha_{min} = \alpha_H = 0.5$ prioritizes the signer's face, and $\alpha_{min} = \alpha_F = 1.6$ prioritizes no regions and corresponds to the quality optimized encoder. Frames from this video are presented in Figure 8.1. As α_{min} increases, the relative priority of the ROI necessarily decreases and intelligibility decreases, as illustrated in Figures 8.1(b) through 8.1(f). Decreasing ROI priority is reflected in a decrease in the CIM-ASL, changing from 3.47 to 3.23. For the subjective intelligibility ratings associated with these values, refer to Figure 8.2. Conversely, as α_{min} increases, PSNR increases from 18.44 dB to 25.73 dB. As this example demonstrates, varying α_{min} can provide a user with control over the level of background distortion while still prioritizing the most important regions of the signer. The following section analyzes PSNR and CIM-ASL over a range of encoding bitrates and α_{min} values, in order to suggest appropriate operating points.

8.3 Characterizing the Quality-Intelligibility Trade-off Across Multiple Operating Points

This section analyzes the rate-distortion performance for several fixed values of α_{min} across varying bitrates and the relationship between PSNR and CIM-ASL for varying α_{min} at fixed bitrates. The rate-distortion performance of the intelligi-



(a) Original video frame

(b) Prioritize all of the ROI. $\alpha_{min} = 0$, PSNR = 18.44 dB, CIM = 3.47

(c) Prioritize all of the ROI with nominal background distortion weight. $\alpha_{min} = 0.02$, PSNR = 21.74 dB, CIM = 3.44

(d) Prioritize only the face and hands. $\alpha_{min} = 0.1$, PSNR = 23.43 dB, CIM = 3.41

(e) Prioritize only the face. $\alpha_{min} = 0.5$, PSNR = 25.21 dB, CIM = 3.32

(f) Quality optimized. $\alpha_{min} = 1.6$, PSNR = 25.73 dB, CIM = 3.23

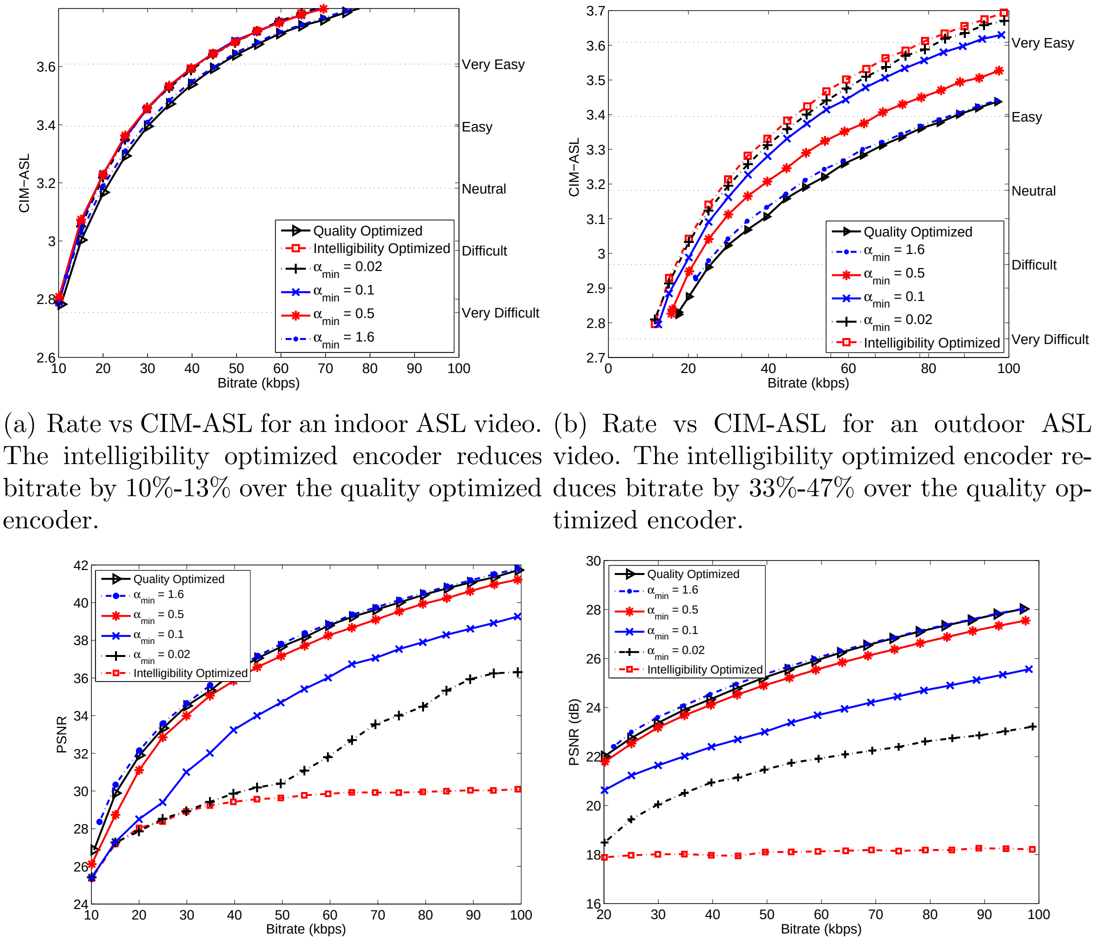
Figure 8.1: Comparison of distortions for different levels of region-of-interest (ROI) priority each at 55 kbps. The encoding option α_{min} specifies the minimum distortion weight to be applied to any region. As α_{min} increases, the torso, hands, and face are allocated fewer additional bits relative to the rest of the frame, causing a decrease in intelligibility. Figure 8.2 specifies the relationship between the CIM-ASL and the predicted subjective intelligibility ratings.

bility optimized encoder and the quality optimized encoder are compared against multiple values of α_{min} across bitrates ranging from 20 kbps to 100 kbps. Figure 8.2 compares PSNR and CIM-ASL for two different ASL videos: a video filmed in a studio with a static background and a video filmed on a busy street with high background activity. In each case, the intelligibility optimized encoder achieves significant bitrate reductions at fixed levels of intelligibility over the quality optimized encoder, demonstrated in Figures 8.2(a) and 8.2(b). The bitrate reductions primarily depend on the level of activity in the background region: 10% to 13% for the indoor video and 33% to 47% for the outdoor video.

Because the intelligibility optimized encoder allocates almost zero rate to the background, the PSNR is dominated by the distortions in the background region. As a result, increasing the bitrate for the intelligibility optimized coder yields a negligible increase in PSNR, as demonstrated in Figures 8.2(c) and 8.2(d). Because it is designed to minimize MSE, the quality optimized encoder achieves the highest PSNR at fixed bitrates, with 4 dB to 10 dB increases in PSNR over the intelligibility optimized encoder.

In addition to comparing the intelligibility optimized and quality optimized encoders, Figure 8.2 also illustrates the effect of varying α_{min}. Setting $\alpha_{min} = 0.02$ applies a nominal weight to the background distortion and results in substantial increases in PSNR with only slight increases in CIM-ASL. Further increasing the α_{min} results in increased PSNR at the expense of intelligibility. When $\alpha_{min} = 1.6$, the modified encoder performs nearly identical to the quality optimized encoder, demonstrating that it effectively behaves as the quality optimized encoder at this point.

The value of α_{min} controls the priority given to the ROI coder. When $\alpha_{min} = 0$,



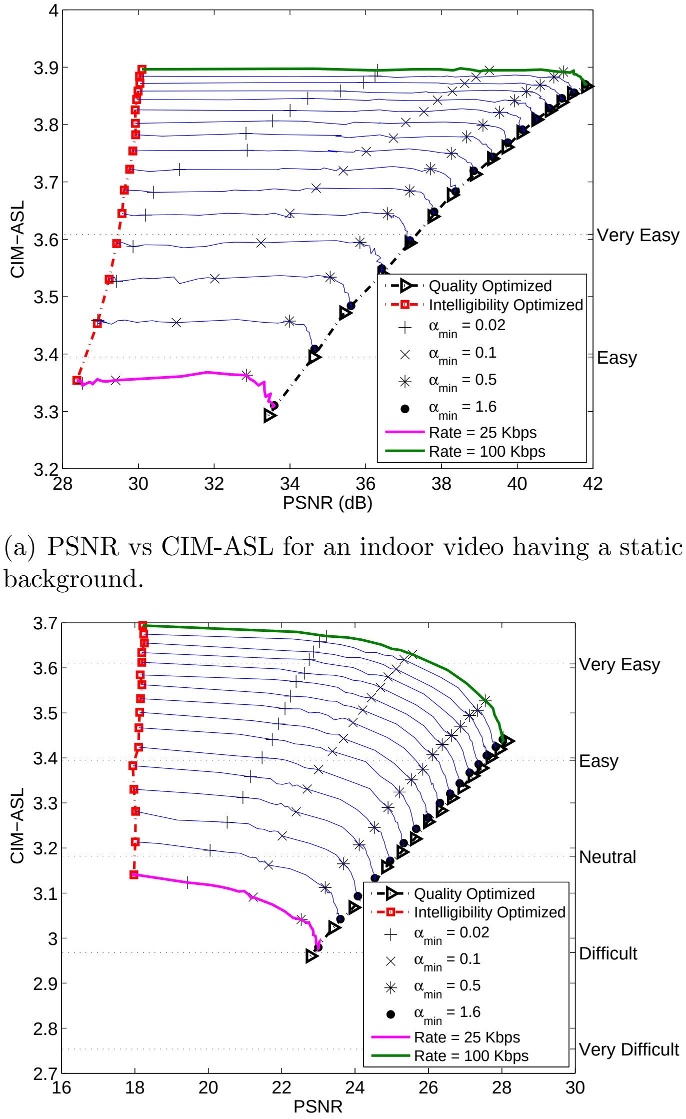
(a) Rate vs CIM-ASL for an indoor ASL video. The intelligibility optimized encoder reduces bitrate by 10%-13% over the quality optimized encoder.

(b) Rate vs CIM-ASL for an outdoor ASL video. The intelligibility optimized encoder reduces bitrate by 33%-47% over the quality optimized encoder.

(c) Rate vs PSNR for an indoor ASL video.

(d) Rate vs PSNR for an outdoor ASL video.

Figure 8.2: Rate-distortion plots for the quality optimized coder, the intelligibility optimized encoder, and several values of α_{min}. For (a) and (b), the left y-axis provides the objective intelligibility distortion measure, CIM-ASL, and the right y-axis provides the subjective rating categories corresponding to the objective distortion values. For (c) and (d), the y-axis provides PSNR. For a fixed level of intelligibility, rate reductions increase for sequences with increasing background activity. When $\alpha_{min} = 0.02$, PSNR increases by several dB and CIM-ASL decreases negligibly. When $\alpha_{min} = 1.6$, all the region distortions are weighted equally and the encoder operates identical to the quality optimized encoder.



(a) PSNR vs CIM-ASL for an indoor video having a static background.

(b) PSNR vs CIM-ASL for an outdoor video having an active background.

Figure 8.3: PSNR versus CIM-ASL for videos with different levels of background activity. Each solid line corresponds to a fixed bitrate and varying α_{min}. The bitrates vary between 25 kbps and 100 kbps in increments of 5 kbps. Depending on the amount of activity in the background, PSNR can be increased by several dB without a significant decrease in CIM-ASL, when compared to the intelligibility optimized encoder.

the encoder is optimizing only for intelligibility. When $\alpha_{min} = 1.6$, the encoder is optimizing only for quality. To explicitly evaluate the trade-off between PSNR and intelligibility afforded by α_{min}, the indoor and outdoor videos are encoded at bitrates ranging from 25 to 100 kbps in increments of 5 kbps. α_{min} is varied from 0 to 0.1 in increments of 0.01 and from 0.1 to 1.6 in steps of 0.1.

Systematically varying α_{min} yields the convex combination of the quality optimized and intelligibility optimized encoders, as illustrated in Figure 8.3. Each curve in the figure corresponds to a fixed encoding bitrate and each point in the curve corresponds to a particular value of α_{min}. As α_{min} increases from 0 to 1.6, the R-D performance of the encoder sweeps the space between the two encoding extremes. When encoding a video for a fixed target bitrate, the value of α_{min} determines the operating point in the trade-off between intelligibility and quality.

The relationship between CIM-ASL and PSNR, as α_{min} varies, depends on the amount of activity in the background region. Decreases in CIM-ASL of approximately 0.2 correspond to a difference of 1 point on a 5 point subjective intelligibility scale. A decrease in CIM-ASL of less than 0.02, i.e., 10% of 0.2, can be considered negligible. When compared to the intelligibility optimized encoder, selecting $\alpha_{min} = 0.5$ increases PSNR in the indoor video between 4.5 dB and 11 dB, depending on the encoding bitrate, with negligible decrease in CIM-ASL, as illustrated in Figure 8.3(a). For the high background activity video in Figure 8.3(b), only a nominal value of $\alpha_{min} = 0.02$ can be selected before the increase in CIM-ASL becomes non-negligible. At this point, PSNR is increased between 1.3 dB and 4.7 dB, depending on the encoding bitrate.

The slope of the PSNR versus CIM-ASL curves is steepest when $0.5 < \alpha_{min} < 1.6$. In this region, when compared to the quality optimized encoder, CIM-ASL increases between 0.03 and 0.08 for a corresponding decrease in PSNR of only between 0.5 dB and 0.6 dB. The signer's face is relatively small compared to the rest of the frame and distortions in the signer's face have the largest impact on CIM-ASL. Prioritizing the signer's face decreases distortions in the corresponding macroblocks and increases intelligibility without creating substantial distortions in the other regions. In the absence of a specific user preference, the coder should choose quality-intelligibility operating points in this high slope region. Because, it is possible to maximize both PSNR and intelligibility for indoor videos by only prioritizing the signer's face, the paired comparison experiment described in the following sections evaluate true user preferences for only outdoor videos.

8.4 Paired Comparison Experiment for Identifying User Preferences in the Quality-Intelligibility Trade-off

The quality-intelligibility coder described in Section 8.2 and the choice of α_{min} controls the trade-off between optimizing for intelligibility and optimizing for quality. A paired comparison experiment is conducted to determine subjective preferences in this trade-off. The primary goal is to identify preferred operating points, if they exist, and to determine under what conditions a user likely to desire a particular operating points.

8.4.1 Stimuli

Reference sign language stories told by a fluent signer at her natural signing pace were filmed at an outdoor location on a busy street having a significant amount of background activity. Videos were recorded at a resolution of 1280×720 pixels and a frame rate of 60 progressive frames per second. For this experiment, the videos are cropped and downsampled in order to match the expected usage conditions, namely a mobile device having a display resolution of 320×240 pixels [19]. This reduced resolution is also required for the simultaneous presentation used in the paired comparison methodology [42]. The videos are temporally subsampled to 15 frames per second, which is above the nominal frame rate required for ASL communication [39].

Three reference stories are selected for the experiment and encoded at one of three bitrates: 20 kbps, 45 kbps, and 80kbps. Each story is encoded at a single bitrate using five different values of α_{min}: 0, 0.02, 0.1, 0.5, and 1.6, corresponding to the five ROI prioritization scenarios illustrated in Figure 8.1. This combination of bitrates and α_{min} values are selected to yield videos that would be rated as difficult to understand (20 kbps), from neutral to easy (45 kbps), and from easy to very easy (80 kbps), as illustrated in Figure 8.4.

8.4.2 Method

The subjective experiment uses a paired comparison methodology with simultaneous presentation, as recommended by ITU-T [42]. Each presentation consists of a pair of coded ASL videos displayed synchronously and side-by-side on a single screen. After watching the video pair, the participant is asked to "please select the video you would prefer to see on a cell phone video call." The collection of video pairs consist of videos generated from the same reference story encoded using two different values of α_{min}.

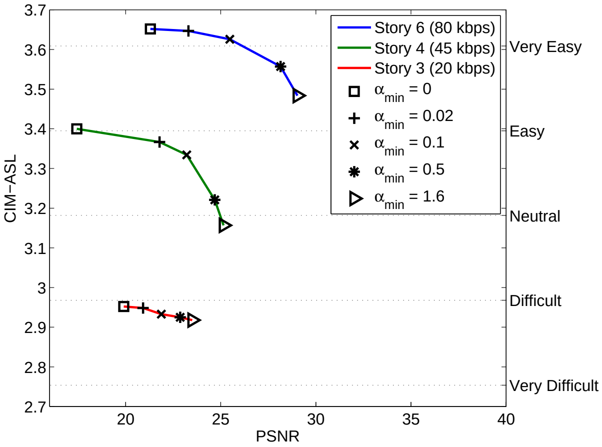


Figure 8.4: PSNR vs CIM-ASL for only the 3 videos and 5 values of α_{min} selected for the paired comparison experiment. The left y-axis provides the CIM-ASL values and the right y-axis provides the subjective rating categories corresponding to the CIM-ASL values.

At each bitrate, the 5 test levels of α_{min} yield 10 pair-wise combinations. The 10 pairs are presented to the participant twice, swapping the left/right display order. None of the test pairs contain videos at different bitrates, assuming that videos at higher bitrates will always be preferred over videos at lower bitrates. This results in 20 paired comparisons per bitrate and 60 comparisons per participant. Following 2 practice examples, the 60 pairs are presented in random order. At the completion of the paired comparisons, participants provide demographic data regarding their level of experience with ASL, their use of video-based communication tools such as video relay services and video phones, and their use of text-based communication tools such as Internet chat and text messaging.

8.4.3 Implementation

Because of the difficulties in recruiting participants who are fluent in ASL, two versions of the experiment were made available: an on-site experiment in a controlled environment at Cornell University and a web-based experiment, in which ASL users in any location could participate. Despite the limitations of web-based perceptual experiments, such as uncontrolled display environments, varying display technologies, and other real-world variability, web-based experiments drastically increase the observer pool and typically provide results that are consistent with lab-based experiments [9, 52].

To guarantee synchronous playback of the video pairs, the on-site experiment was implemented in Matlab, using the Psychophysics Toolbox [11, 44, 59], which offers extremely precise control over the video playback timing. For the web-based experiment, an individual video file was created for each pair by decoding the compressed videos, horizontally concatenating the decoded frames, and re-encoding the side-by-side video at a sufficiently high bitrate such that no new compression artifacts were introduced. The video pairs in both the on-site and web-based experiments were identical, though the web-based version offered a shortened experiment, wherein participants only viewed each pair once, without evaluating the left/right swapped pair. Pairs used in the shortened experiment were selected such that every 2 participants evaluated exactly the same set of pairs as a single participant in the full-length experiment.

8.4.4 Data Processing

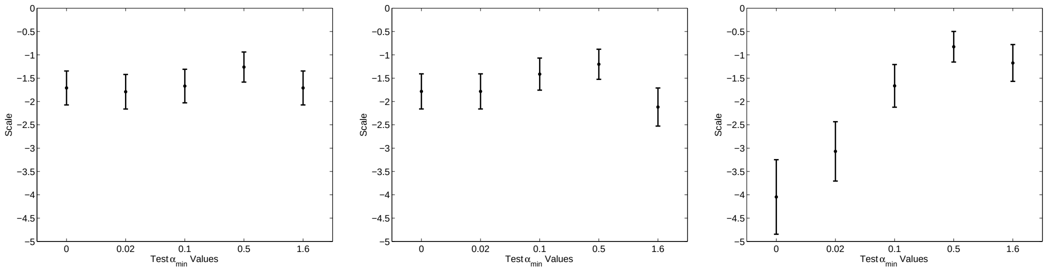
The paired comparison methodology acquires data to estimate the probability that stimulus i is preferred over stimulus j. The Bradley-Terry model provides a framework for mapping the pair-wise probability estimates of preference to scale values for each stimulus [10]. The scale values rank the collection of stimuli, determining the relative preference of each value of α_{min}. Because the stimulus pairs in the experiment never contain videos at two different bitrates, scale values are generated independently at each of the three tested bitrates.

8.5 Results and Discussion

A total of 12 ASL users participated in this experiment: 3 on-site participants and 9 web-based participants. Of the 9 web-based participants, 4 opted for the shortened version, yielding a total of 600 comparisons (200 at each bitrate).

Applying the Bradley-Terry model [10], scale values for each tested α_{min} are computed at each bitrate. Following the methodology discussed in Ref.38, a χ^2_{t-1} hypothesis test with $t - 1$ degrees of freedom ($t = 5$ levels of α_{min}) determines whether the scale values are statistically different from a uniform distribution. If the null hypothesis holds, all values of α_{min} are equally preferable. If the null hypothesis is rejected, at least one α_{min} is preferred over the others. The computed scale values, with 95% confidence intervals, are provided in Figure 8.5. Table 8.1 provides the results of the hypothesis tests for uniformity.

At 80 kbps, the scale values demonstrate a preference when $\alpha_{min} \geq 0.1$, as plotted in Figure 8.5(c). Each of the scale values for $\alpha_{min} \geq 0.1$ have overlapping



(a) 20 kbps.

(b) 45 kbps.

(c) 80 kbps.

Figure 8.5: Scale values generated from the complete set of paired comparison data using the Bradley-Terry model. Error bars indicate the 95% confidence intervals.

Table 8.1: Table of p-values for χ^2_4 hypothesis test on the uniformity of the scale values [38], for different groups of participants. The null hypothesis indicates that the scale values are not statistically different from a uniform distribution, i.e., each α_{min} is equally preferable. Entries in bold indicate that the null is rejected at 95% confidence ($p < 0.05$). The "ASL FL" and "ASL SL" groups correspond to participants for whom ASL is their first language (FL) or second language (SL). The "Heavy Video Use" and "Light Video Use" groups are divided according to their level of experience with video-based communication technologies.

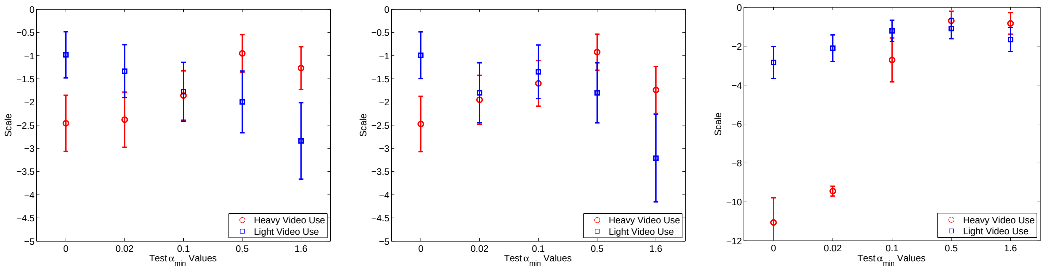
Bitrate	Complete Set	ASL FL	ASL SL	Heavy Video Use	Light Video Use
20 kbps	0.370	0.097	0.084	**1.7e-4**	**0.003**
45 kbps	**0.017**	0.261	**0.022**	**0.003**	**3.9e-4**
80 kbps	**0**	**8.9e-14**	**4.0e-8**	**0**	**0.003**

confidence intervals and can be considered equally preferable. At $\alpha_{min} = 0.1$, because of the relatively high encoding bitrate, the quality-intelligibility optimized coder produces video predicted to be very easy to understand, as seen in Figure 8.4. In this case, the smaller values of $\alpha_{min} = 0$ and $\alpha_{min} = 0.02$ significantly reduce the overall quality (PSNR) while providing only negligible improvements in intelligibility (CIM). This saturation effect implies that when coding an ASL video, when the bitrate is sufficiently high for producing video considered very easy to understand, any additional rate must be allocated to maximize a quality constraint.

At 45 kbps, $\alpha_{min} = 0.1$ and $\alpha_{min} = 0.5$ are preferred over $\alpha_{min} = 1.6$. Referring to Figure 8.4, these two values of α_{min} correspond to the points on the PSNR-CIM curve having the largest slope. These points are preferred because they provide the largest increase in the CIM for the corresponding decrease in PSNR.

At 20 kbps, the scale values are not statistically different from a uniform distribution, indicated by the hypothesis test results in Table 8.1. As illustrated in Figure 8.4, the PSNR-CIM curve at this bitrate is relatively flat; the relative change in the CIM is small compared to the relative change in PSNR, for varying α_{min}. One might expect a preference for the highest quality video, when the change in CIM is small. However, the lowest quality video ($\alpha_{min} = 0$) is still equally preferable to the highest quality video ($\alpha_{min} = 1.6$).

A uniform distribution of scale values can be attributed to one of two statistical models. In the first model, each individual observer has no preference and is arbitrarily selecting one of the two videos in a pair. This case implies that every value of α_{min} yields the same perceptual response and no value is preferred over another. In this case, the selection of an operating point in the quality-intelligibility



(a) 20 kbps.

(b) 45 kbps.

(c) 80 kbps.

Figure 8.6: Scale values generated from paired comparison data of groups of participants who use both video relay services and video phone technology (denoted "heavy video use") and those who do not (denoted "light video use"). Scale values are generated according to the Bradley-Terry model. Error bars indicate the 95% confidence intervals.

trade-off is arbitrary, since all points are truly equal. In the second model, a single observer (or group of observers) demonstrates a preference for a particular α_{min}, while a sampling of the entire population of observers exhibits no preference. In this case, each value of α_{min} is preferred by a specific individual (or group) and that preference varies across individuals (or groups), supporting the need for a user-specified operating point in the quality-intelligibility trade-off.

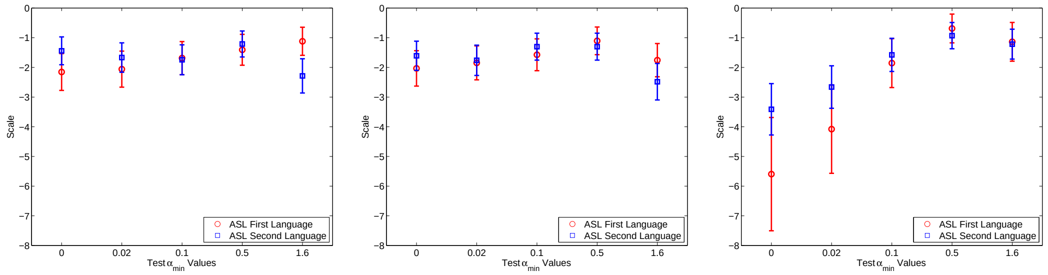
An analysis of the scale values for different groups of participants provides evidence for the second model. In particular, groups divided according to their use of video-based communication technologies have opposite (and non-uniform) preference rankings. Because the collection of data is sufficiently small, the relevant groups have been identified manually, though one could use a recursive procedure for identifying groups having homogeneous preferences [75]. The 7 participants who reported using video relay services and video phone technology are denoted the "heavy video use" group. The remaining 5 participants are denoted the "light video use" group, because some individuals in this group use Internet chat services,

Table 8.2: Table of p-values for χ^2_4 hypothesis test on differences between groups [38]. The null hypothesis indicates that the scale values from each group are statistically equivalent. Entries in bold indicate that the null is rejected at 95% confidence ($p < 0.05$), i.e., the groups are statistically different from each other. The "ASL FL" vs "ASL SL" column compares groups that correspond to participants for whom ASL is their first language (FL) or second language (SL). The "Heavy Video Use" vs "Light Video Use" column compares groups that are divided according to their level of experience with video-based communication technologies.

Bitrate	ASL FL vs ASL SL	Heavy Video Use vs Light Video Use
20 kbps	**0.019**	**7.1e-7**
45 kbps	0.321	**5.4e-5**
80 kbps	0.186	**2.9e-7**

such as Skype, which offer video communication as a secondary feature. At every bitrate, the scale values for each of the two groups are statistically different from uniform, as shown in Table 8.1. Furthermore, using the methods in Ref. 38, a χ^2_4 hypothesis test identifies a significant difference between these two groups at every bitrate, i.e., these two groups have statistically different preferences. The results of this hypothesis test, with p-values, are provided in Table 8.2.

At each tested bitrate, the "light video use" group has a significantly higher preference for $\alpha_{min} = 0$ than the "heavy video use" group. Furthermore, at 25 kbps and 50 kbps, the "light video use" group prefers $\alpha_{min} = 0$ over $\alpha_{min} = 1.6$, as shown in Figure 8.6. Conversely, the "heavy video use" group demonstrates a preference for $\alpha_{min} = 1.6$, where videos are coded for quality. This preference is most evident at 80 kbps, where the values of $\alpha_{min} \geq 0.1$ are preferred unanimously over $\alpha_{min} = 0$ and $\alpha_{min} = 0.02$, causing the large difference in scale values in



(a) 20 kbps.

(b) 45 kbps.

(c) 80 kbps.

Figure 8.7: Scale values generated from paired comparison data of participants whose first language is ASL or whose first language is not ASL. Scale values are generated from paired comparison data according to the Bradley-Terry model. Error bars indicate the 95% confidence intervals.

Figure 8.6(c).

Variations in the preferences of the "heavy video use" and "light video use" groups may be attributable to differences in their prior experience of digital video. Video-based communication technologies typically use a quality criteria when coding video (e.g., they maximize PSNR). In this case, the coding distortions are generally distributed evenly across space. The strictly intelligibility optimized coder ($\alpha_{min} = 0$) produces video in which the signer and the background have significantly different distortion levels. This disparity in the spatial distribution of distortion substantially differs from a quality optimized coder, and, consequently, differs from the prior experiences of the "heavy video use" group, resulting in a preference for the coded ASL video that is more consistent with their expectations.

An alternative grouping of ASL users divides the collection based on the level of experience with ASL. The first group consists of those whose first or primary language is ASL, which commonly includes deaf persons or hearing children of deaf adults. The second group consists of those who have learned ASL as a second language. The differences between these groups are only significant at 20 kbps and not to the same degree of confidence as the differences for the “video use” groups. The p-values are summarized in Table 8.2. Furthermore, the scale values for each of these groups are consistent with those computed from the complete data set, as illustrated in Figure 8.7. Other partitions of the participants yield similar conclusions; a user’s experience with video-based communication serves as the most meaningful predictor of the preferred operating point in the quality-intelligibility trade-off.

8.6 Summary

This chapter presented a modification of the intelligibility optimized coder that provides an optional user-controlled trade-off between optimizing intelligibility, as computed by CIM-ASL, and optimizing quality, as computed by PSNR. Even in videos having highly active backgrounds, PSNR can be increased by at least 4dB without sacrificing intelligibility. The modified coder suggests potential operation points that are studied in a a paired comparison experiment, conducted to evaluate specific user preferences for coded ASL video. High activity outdoor videos at 3 bitrates were coded using 5 test levels for α_{min}. At 80 kbps, users preferred videos coded according to the quality criteria, because the intelligibility of these videos was sufficiently high. At the lower tested bitrates of 45 kbps and 20 kbps, the preferences varied with user demographics. Participants having significant experience using video-based communication technologies preferred video coded according to the quality criteria while those with little experience preferred video coded according to the intelligibility criteria. The existence of these two classes of individuals confirms the need for a user-centric encoding option, because the most desirable quality-intelligibility operating points vary across individuals and across bitrates.

CHAPTER 9

CONCLUSION

As network bandwidths continue to increase, digital video technology is becoming commonplace. Streaming video services make digital video content available on personal computers and, more recently, mobile devices. Video conferencing technologies provide more personal communication in both corporate environments and in the home. The increasing availability and use of digital video poses two fundamental research questions. First, how can we most efficiently compress digital video for transport over a variety of networks having different resource constraints? Second, what are the proper computational criteria for evaluating the quality of the compressed digital video being viewed by the end user? The answer to both of these questions varies significantly across applications. Compression algorithms designed for digital cinema may not be appropriate for video conferencing applications. The definition of quality varies heavily depending on the expectations of the end user.

This dissertation has addressed each of these questions in the context of a real-time videoconferencing system for American Sign Language (ASL) video, which operates on mobile devices in a cellular network. As a communication tool, compressed ASL video must be evaluated according to the intelligibility of the conversation, not according to conventional definitions of video quality. A computational model of the intelligibility of ASL video was developed and shown to be accurate with respect to true intelligibility ratings as provided by human subjects. The computational model was applied in the development of video compression techniques that are optimized for ASL video, yielding a fully closed-loop encoding system for ASL video.

Guided by linguistic principles and human perception of ASL, the full-reference computational model of intelligibility for ASL (CIM-ASL) provides a suitable criteria for evaluating compressed ASL video. The CIM-ASL measures distortions only in regions relevant for ASL communication, using spatial and temporal pooling mechanisms that vary the contribution of distortions according to their relative impact on the intelligibility of the compressed video. The model is trained and evaluated using ground truth experimental data, collected in three separate perceptual studies. The CIM-ASL provides accurate estimates of subjective intelligibility and demonstrates statistically significant improvements over computational models traditionally used to estimate video quality.

The CIM-ASL was incorporated into an H.264/AVC compliant video coding framework, yielding a closed-loop encoding system optimized explicitly for ASL intelligibility. This intelligibility optimized coder significantly increases compression efficiency, yielding bitrate reductions between 10% and 42% without reducing intelligibility, when compared to a general purpose H.264/AVC encoder. Furthermore, the structure of ASL, consisting of multiple regions carrying varying amount of information, facilitates reduced complexity encoding modes that allocate computational resources according to the regions in the video deemed important by the CIM-ASL. These region-based computation allocation techniques yield a 16% improvement in the overall encoding speed, with a negligible effect on intelligibility.

The purpose of the intelligibility optimized encoder is to generate video that is suitable for real-time ASL communication. Ultimately, the preferences of ASL users determine the success of the intelligibility optimized coder. In order to accommodate user preferences, a new encoding methodology was developed, which provides a user-centric mechanism for varying between the intelligibility optimized coder and a general purpose video coder. This user-centric encoder was evaluated in a perceptual experiment, which demonstrated that the user preferences vary depending on the demographics of the participants and that a significant proportion of users prefer the intelligibility optimized coder. This study also revealed that the strongest predictor of a user's preference is her prior experience with video-based communication; heavy video users demonstrate a slight preference for the general purpose video coder.

Future Directions

While this dissertation has primarily been in the context of ASL video, the methodology is applicable and extensible to any video content. Specifically, this work demonstrated that intelligibility can be measured by computing errors in the regions containing linguistically important information places. One of the fundamental results was the flexibility in choosing exactly how the error is computed. For example, the face, hands, and torso of the signer are known to be important, but computing any one of three error measures in these regions (MSE, SSIM, or NICE) yielded an accurate estimation of intelligibility. To state this colloquially, "it's not what you measure, it's where you measure it." For more general video content, it may be possible to identify regions of visual attention (or bands of spatial frequency) that carry the most important information and apply simple error measures in these important regions.

By identifying the existence of distinct demographic groups with opposite preferences, this work argues for a novel, user-centric methodology in video processing. Regardless of the specific context, the future of digital video compression and quality assessment lies in the development of algorithms that appropriately consider the end user. Computational models of video quality must consider the context of the application, i.e., why does the user want to watch this video and what types of degradations are they willing (or not willing) to tolerate. Even within the same application, different users will have different expectations of quality, based on their own experiences and perception. Understanding this difference is crucial in the design of advanced compression techniques. Moving forward, it will no longer be sufficient to provide a one-size-fits-all encoding algorithm. As my research has demonstrated, the criteria for which the video is compressed must be suitable to the application. Furthermore, the expectations of the end users will heavily bias their perception of video quality. Taking a user-centric approach to video encoding requires one to identify variations across users and to determine how those variations can be efficiently accommodated by the encoding algorithm. Providing a user with the right video, suitable for their personal preferences, will ultimately lead to a high quality-of-experience for everyone.

APPENDIX A

REAL-TIME FACE AND HAND DETECTION ON A MOBILE DEVICE

A.1 Introduction

The increase in processing power on modern mobile devices allows for the implementation of more advanced image and video processing algorithms, such as real-time videoconferencing. Rapidly increasing cellular network bandwidth also facilitates the transmission of video across the cellular networks. Two-way video communication in this setting requires real-time processing on a cellular device. While cellular devices are more powerful than in the past, they still offer little computational power when compared to modern desktop computers. Slow processors constrain the complexity of the algorithms that can be implemented in real-time on a mobile device. Furthermore, the bandwidths available on a cellular network are significantly smaller than those available on a wired network. Consequently, advanced compression techniques are required to generate video sequences that are useful to the end users.

In traditional videoconferencing, enhancing the quality of the face regions is an effective method of improving the overall perceptual quality of the video [14, 29, 49]. Videoconferencing systems can also be applied to the specific task of transmitting American Sign Language (ASL) video. Such systems allow members of the Deaf community to communicate in their native language. Within this context, the information itself is contained in the signer's facial expressions and hand gestures. Encoding the face and hands with higher fidelity is essential to preserving the information in the sign language conversation [4, 24]. In both of these cases, identifying and encoding only the important portions of the video can result in a significant bit rate savings.

Many algorithms have been proposed to identify faces in images or to identify and track hands in a video sequence (see [91], [56] for surveys). Unfortunately, a large number of these algorithms are not appropriate for low-complexity devices. This work aims to present and analyze low complexity face and hand detection algorithms that can be implemented on a mobile phone. In this work, three face detection techniques are implemented on a mobile device and evaluated in terms of accuracy and speed. Section A.2 describes the algorithms that are implemented on the mobile device while Section A.3 compares the detection accuracy and speed of each of the algorithms. The shape-based detection algorithm achieves the fastest detection times of 165 msec, but fails to accurately detect the face in all cases. Local binary patterns and the Viola-Jones algorithm are both capable of accurately detection the face, but are significantly slower. In Section A.4, the results of the detection algorithms are combined with an H.264/AVC video encoder in order to encode relevant portions of the video (e.g. the face and hands) with higher fidelity.

A.2 Detection Algorithms

In both videoconferencing and ASL video telephony, encoding only the relevant portions of the sequence at a high quality can yield significant gains in compression. This improved compression is essential for meeting the bandwidth constraints of cellular networks, but requires additional computational complexity for identifying those relevant regions. In this section, the face and hands of an individual are identified through the use of skin segmentation and face detection algorithms.

Based on the detected locations of the face and hands, the 16x16 macroblocks in the video are labeled as either face, hand, or background.

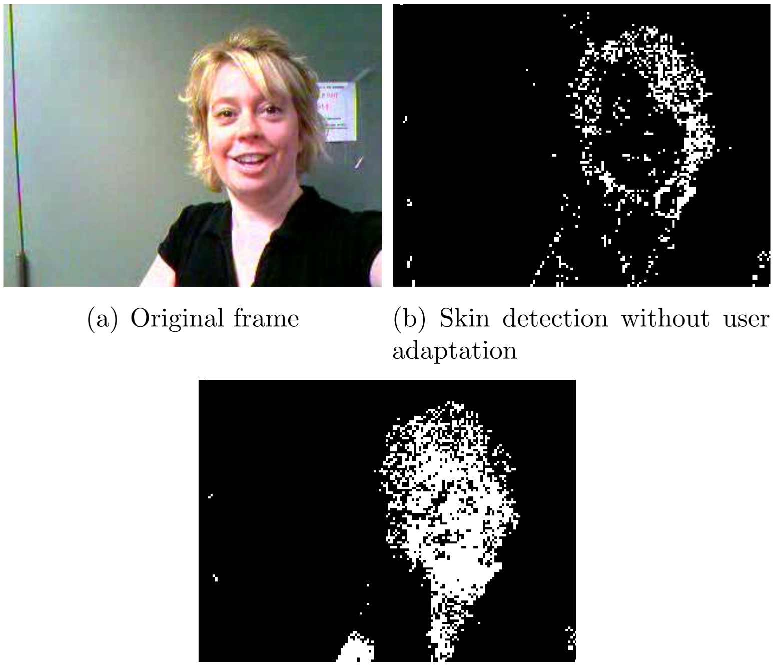
A.2.1 Color and shape based face detection

Face detection can be performed using shape and color information extracted from the image [14]. Skin pixels have a color distribution that is distinct from non-skin pixels [61]. Skin detection is performed in the YUV color space. Because the H.264/AVC encoder also operates within this color space, no color conversion is required to perform the skin detection. The chrominance values (U and V) of skin pixels are modeled as a bivariate Gaussian distribution. The mean μ and covariance matrix Σ of the distribution are generated from a sample set of skin pixels. Skin-color segmentation is implemented by thresholding the Mahalanobis distance, $D_M^2(x)$, between a given pixel's chrominance values x and the skin pixel distribution.

$$D_M^2(x) = (x - \mu)^T \Sigma^{-1} (x - \mu) < \alpha \tag{A.1}$$

The skin segmentation can be improved by incorporating a user-adaptive skin model. During a video call, the skin color statistics are updated to more accurately model those of the current user. Figure A.1 illustrates the improvement in the skin detection by performing this update. In this case, the skin pixels were manually selected and added to the model. In future work, this process can be automated. This update can be done while the call is being connected, by asking the user to hold her hand in a specific location. It can also be done automatically, by first applying face detection then extracting skin pixels, as in [32].

Provided that the skin segmentation is very accurate, a shape-based approach



(a) Original frame

(b) Skin detection without user adaptation

(c) Skin detection with user adaptation

Figure A.1: Comparison of skin detection algorithm with and without user adaptation.

can be used to differentiate between the users face and hands. Given a binary skin map, a connected component analysis is used to identify the size and location of each cluster of skin pixels. Clusters of skin pixels smaller than a fixed threshold are discarded as noise. The remaining skin components are filtered with the morphological erode operator. This shape-based approach erodes the binary skin map using a vertically-oriented elliptical structuring element. Because the human head can be roughly modeled as an ellipse, the face is identified as the largest connected component remaining after the erosion [91].

In the presence of noisy backgrounds or poor lighting conditions, the skin detection can yield a non-trivial amount of false alarms, especially if the background contains skin-colored objects. Because of this, the morphological shape-based face detection fails and feature based techniques are required to identify the face region. Two feature-based detection algorithms are considered: local binary patterns and the Viola-Jones algorithm.

A.2.2 Local binary patterns

The first approach generates features based on local binary patterns (LBP) of luminance pixels [37]. The LBP is calculated from a neighborhood of L pixels surrounding each pixel by thresholding each neighbor based on the center pixel's value and mapping this to a binary number. For example, using a 3x3 neighborhood ($L = 8$), a pixel whose neighbors are all greater than itself will have a LBP of 11111111. As a consequence of this binary representation, there are only 2^8, or 256, possible binary patterns for an individual pixel.

In order to perform the face detection, a set of LBPs are mapped to an appropriate feature as follows. Given a candidate window, the classification feature is the distribution of all of the local binary patterns in that window, e.g. the 256 bin histogram of possible binary patterns. The classifier is trained on a set of 19x19 face images taken from the FERET database [60]. The average of all the face histograms is used in the classification task. Face detection is performed by searching candidate 19x19 windows in the input image. For each window, the histogram of LBP values is computed and compared against the average face histogram using the Chi square distance, as in Equation A.2. H_C and H_T correspond to the candidate and trained histograms, respectively.

$$\chi^2(H_C, H_T) = \sum_{i=1}^{256} \frac{(H_{Ci} - H_{Ti})^2}{H_{Ci} + H_{Ti}} \tag{A.2}$$

If $\chi^2(H_C, H_T) < \beta$, the candidate window is identified as a face. In order to identify faces at multiple scales, the classification algorithm is run on downsampled versions of the original image. Overlapping face regions are identified as a single face with a bounding box corresponding to the average of the overlapping regions.

Since each pixel in a window is compared to each of its neighbors, the LBP classifier requires $O(W_W W_H L)$ operations, where W_W and W_H are the width and height of the window and L is the number of neighbors. To search the entire image, the total number of operations is $O(N W_W W_H L)$, where N is the number of candidate windows and is a function of the image size and number of image scales included in the search.

One of the main computational benefits of the LBP-based classifier is that the features themselves can be computed using only fixed-point operations. This is especially important on a mobile device in which floating-point operations must be emulated, which can be prohibitively slow. The most computationally costly part of the LBP-based classifier is the image scaling, since a pyramid of downsampled images must be generated for each scale that is to be searched.

A.2.3 Viola-Jones classifier cascade

The Viola-Jones face detection algorithm [83] can also be applied to identify the face region. This detection algorithm uses a series of classifier stages. At each stage, simple Haar-like rectangular features are computed in the candidate window. If the window is classified as a face, it continues to the next stage. Each stage is increasingly complex in terms of the number of features, in order to eliminate more non-face windows. Only a candidate window containing a face passes through all the stages in the classifier.

This paper uses the OpenCV implementation of the Viola-Jones algorithm [1], which has been ported for use on the mobile device. The OpenCV package provides a classifier cascade which has been trained for frontal face views. The Viola-Jones classifier has several computational benefits. First, the classifier cascade is organized such that simple classifiers using only a few features can quickly eliminate non-face windows. Second, the use of the integral image representation and simple rectangular features enables the algorithm to detect faces at a range of sizes without rescaling the entire image. The features themselves are scaled to search over larger windows in the image, without having to downsample the original image.

For an individual window, the Viola-Jones classifier requires $O(FS_F)$ operations, where F is the number of features being computed and S_F is the size of the feature (i.e., the number of pixels contained within the rectangular feature). By design, the value of F can vary tremendously. For windows containing a face, the candidate window passes through each stage of the classifier and, in the classifier used here, 2135 features in total are computed. However, a majority of the candidate windows are rejected by the first stage of the classifier, which computes only 3 features. To search the entire image, the total number of operations is $O(NFS_F)$, where N is the number of candidate windows and is a function of the image size and number of image scales included in the search.

The major drawback for implementation on a mobile device is the number of floating point operations. There are 21 classifier stages with between 3 and 200 features per stage. At each stage, the features are computed and compared against a floating point threshold, which results in a very large number of floating point operations, especially for windows which pass through multiple classification stages.

A.2.4 Hand Detection

While simply identifying the face region may be sufficient for generic videoconferencing, further processing must be done for American Sign Language (ASL) video. In ASL, information is conveyed through both facial expressions and hand gestures. In order to optimally encode ASL videos, the hands must also be identified. Following both skin segmentation and face detection, the signer's hands are identified as the large skin clusters not corresponding to the signer's face.

A.3 Accuracy and Computational Results

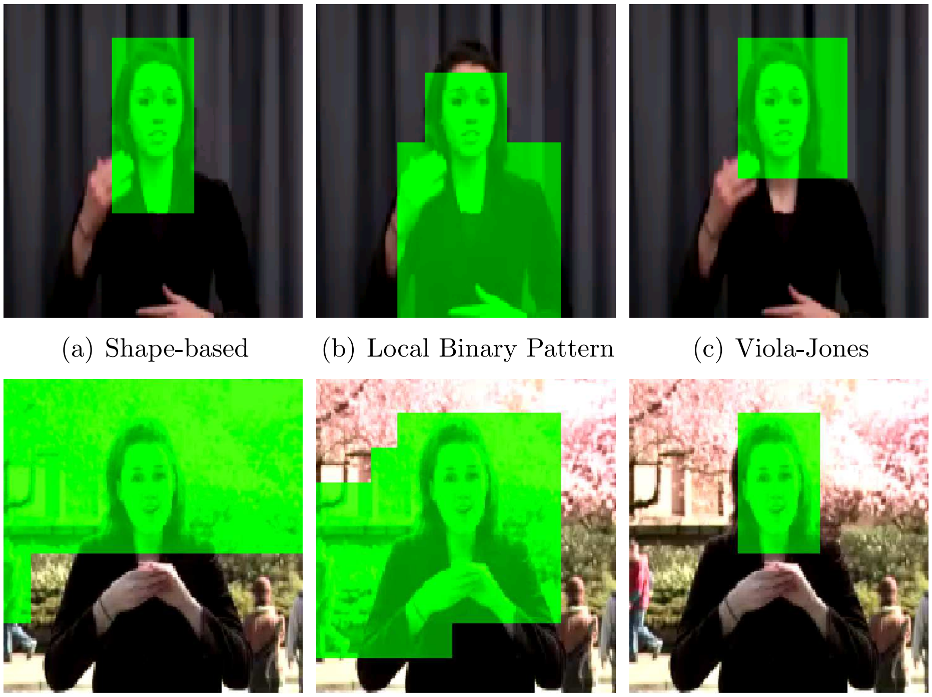
The algorithms described in Section A.2 are implemented on an HTC Apache PocketPC with an Intel PXA270 processor running at 416 MHz, with 64 MB RAM, and a 240x320 LCD display. The device runs the Windows Mobile operating system. Three test videos of American Sign Language are used for the evaluation. Two of the videos were recorded using professional video equipment and downsampled to QCIF resolution (176x144) at 10 frames per second. One of these videos was recorded indoor in a studio, the other was recorded outdoors. The third video was captured using the camera on the PocketPC while being held by the signer. It was downsampled from QVGA to a resolution of 160x120 at 15 frames per second.

One of the primary factors controlling the speed of the feature based face detection algorithms are the number of image scales included in the search space. A large number of scales ensures that faces of any size will be found, but each scale adds a significant amount of computation time. The number of scales is limited by controlling the scaling factor and the minimum/maximum expected face size in the image. In this implementation, the scaling factor was set to 1.25, the maximum face size was set to 60% of the image width, and the minimum face size was set to 15% of the image width. Also, at each image scale, the search is performed for every other pixel.

The fastest face detection method is the shape-based approach, which runs at an average of 165 msec per frame. This method is very successful when the skin detection is very accurate. For the indoor scene, the average face detection rate was 93%. However, if the skin detection yields a non-trivial amount of false alarms, the shape-based approach completely breaks down, as is the case in the outdoor scene, as illustrated in Figures A.2(a) and A.2(d).

The LBP-based classifier achieves an average detection rate of 91%, but has a very large number of false positives, as illustrated in Figures A.2(b) and A.2(e). Out of 477 frames, the LBP classifier yielded 162 false alarms. The LBP classifier was also the slowest of the three methods, running at an average of 1841 msec per frame. Finally, the Viola-Jones classifier achieves an average detection rate of 90% with only 27 false alarms and runs at 1508 msec per frame.

Of the three face detection techniques, the Viola-Jones classifier achieves the optimal trade-off between positive detections and false alarms. However, in its default implementation, it runs at fewer than 1 frame per second. The search speed can be improved by decreasing the number of image scales (i.e., increasing the scaling factor) or limiting the search space at each scale. The search space can be reduced by only evaluating candidate windows if they contain skin pixels. It can also be reduced by limiting the search to windows which were within one macroblock of a face block in the previous frame. Table A.1 demonstrates the speed improvements for each of these cases. At best, the Viola-Jones algorithm



(a) Shape-based (b) Local Binary Pattern (c) Viola-Jones

(d) Shape-based (e) Local Binary Pattern (f) Viola-Jones

Figure A.2: Typical face detection results of the three detection algorithms. Green blocks indicate the macroblock contains part of the face. The shape-based approach works very well on the indoor sequence but fails when the skin detector yields inaccurate results, as in the cherry blossoms in the background. The LBP classifier achieves high detection rates but also has many false positives. The Viola-Jones classifier accurately detects the face with the fewest false positives.

runs at approximately 2.8 frames per second.

A.4 Encoding Platform

The frame segmentation maps are used by an H.264/AVC video encoding algorithm to achieve increased compression while maintaining the quality in the region-of-interest. In order to capture and encode video sequences in real-time, the x264

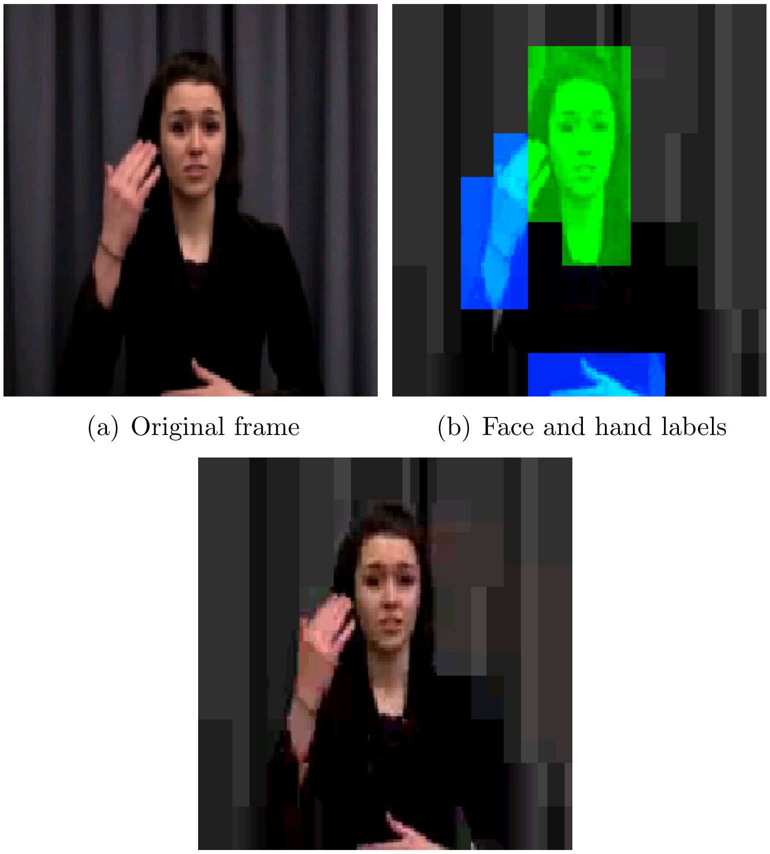
Table A.1: Improvements in speed of the Viola-Jones classifier by increasing the image scaling factor and by reducing the search space. The results are presented for the indoor video at QCIF resolution and are consistent for the other videos.

Image Scale	Search Restriction	Positive Detections	False Positives	Average Detection Time
Scale 1.25	No Restriction	87%	0	1257 msec
Scale 1.5	No Restriction	92%	0	806 msec
Scale 2.0	No Restriction	99%	0	423 msec
Scale 1.25	Face in Previous Frame	87%	0	1115 msec
Scale 1.5	Face in Previous Frame	92%	0	671 msec
Scale 2.0	Face in Previous Frame	99%	0	353 msec
Scale 1.25	Skin in Window	94%	0	975 msec
Scale 1.5	Skin in Window	95%	0	636 msec
Scale 2.0	Skin in Window	98%	0	389 msec

video encoder was ported to the mobile phone. x264 is an open-source implementation of H.264 which has been shown to be 50 times faster than the JM reference software with little reduction in performance [51]. As demonstrated in previous work, appropriately applying face and hand segmentation maps to sign language videos results in rate reductions as large as 60%, without sacrificing the overall intelligibility of the video [25]. The mobile phone can encode such videos by executing the face and hand detection algorithms prior to invoking the encoder. Figure A.3 presents a frame encoded with this region-of-interest adjustment, using the shape-based detection. The quantization parameter of the face and hand macroblocks is reduced (i.e., the quality is increased) at the expense of the rest of the frame.

A.5 Summary

This chapter analyzes low complexity methods for identifying face and hand regions in a mobile video telephony setting. Shape-based processing is the most computationally efficient method for identifying the face and hands, but cannot adequately identify these regions in the presence of skin-colored backgrounds. In these noisy environments, feature-based face detection techniques are applied to the segmentation task. The Viola-Jones algorithm achieves 90% detection rates with almost no false positives. The feature-based techniques are further optimized by restricting the search space based on the location of skin pixels in the current frame or the face in previous frames. The detection algorithms provide an H.264/AVC encoder with a macroblock-level map of the face and hands, allowing for the use of region-of-interest encoding techniques.



(a) Original frame

(b) Face and hand labels

(c) ROI encoded frame

Figure A.3: Illustration of varying region-of-interest quality. Note that the face and hands of the signer are maintained while the background is heavily distorted.

APPENDIX B

COMPARING FULL-REFERENCE QUALITY ESTIMATORS USING HEURISTIC RATE-DISTORTION OPTIMIZATION

B.1 Abstract

This chapter presents work that was performed with Paul Rademacher, an M.Eng student. While not directly related to the topic of this dissertation, it is included as an example application of several fundamentals tools used throughout the dissertation (e.g., rate-distortion optimization, genetic algorithms, and quality assessment).

Image quality estimators strive to accurately estimate the subjective quality of degraded images. This work proposes a methodology for performing rate-distortion (R-D) optimization using an arbitrary quality estimator. The proposed methodology uses a genetic algorithm to select a set of R-D optimal quantization step sizes in a JPEG-2000 encoding framework. Optimal step sizes can be found for any quality estimator that can provide a score given a compressed image. A comparison of image sets that are R-D optimal for a collection of quality estimators serves as a novel method for evaluating the performance of image quality estimators.

B.2 Introduction

Image quality estimators (QEs) strive to accurately estimate the subjective quality of a degraded image. In particular, a full-reference QE takes a reference and degraded image and produces a score that is expected to be consistent with the human rated quality of the degraded image. Because full-reference QEs have access to the reference image, they can be applied to algorithm optimization, such as developing image compression techniques that optimize more perceptually meaningful distortion models [40].

This paper presents a method for incorporating an arbitrary QE into a rate-distortion (R-D) optimization procedure. Applying a QE in this way yields images for which rate is distributed according to the implicit criteria set by the particular QE, e.g., bits are spent on what the QE deems important for quality. This methodology is particularly useful for QEs that are non-convex and not easily applied in traditional R-D optimization algorithms. The R-D optimization yields the images that lie on the R-D convex hull for a specific QE, i.e., each image has the highest possible QE score for a given rate constraint.

Image QEs are typically evaluated according to their statistical accuracy in estimating human quality ratings of degraded images [70]. In this work, comparisons between the sets of convex hull images for a collection of QEs allows for a systematic evaluation of the accuracy of a QE, without requiring expensive subjective testing. Based on the principles described in [23], the proposed R-D optimization procedure is applied to identify discrepencies among a collection of different QEs. Any discrepencies among the collection of QEs implies that one of the QEs in disagreement will be inaccurate with respect to true subjective quality.

Inaccuracies are defined for image pairs in terms of misclassification errors such as false ties (QE rates images with equal quality, humans rate images with different quality), false differences (QE rates images with different quality, humans rate images equally), and false ordering (QE rates image A better than B, humans rate image B better than A) [12]. The proposed R-D optimization generates collec-

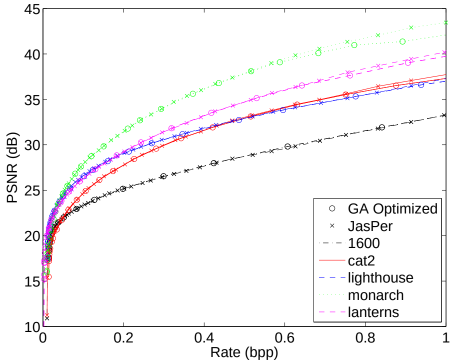


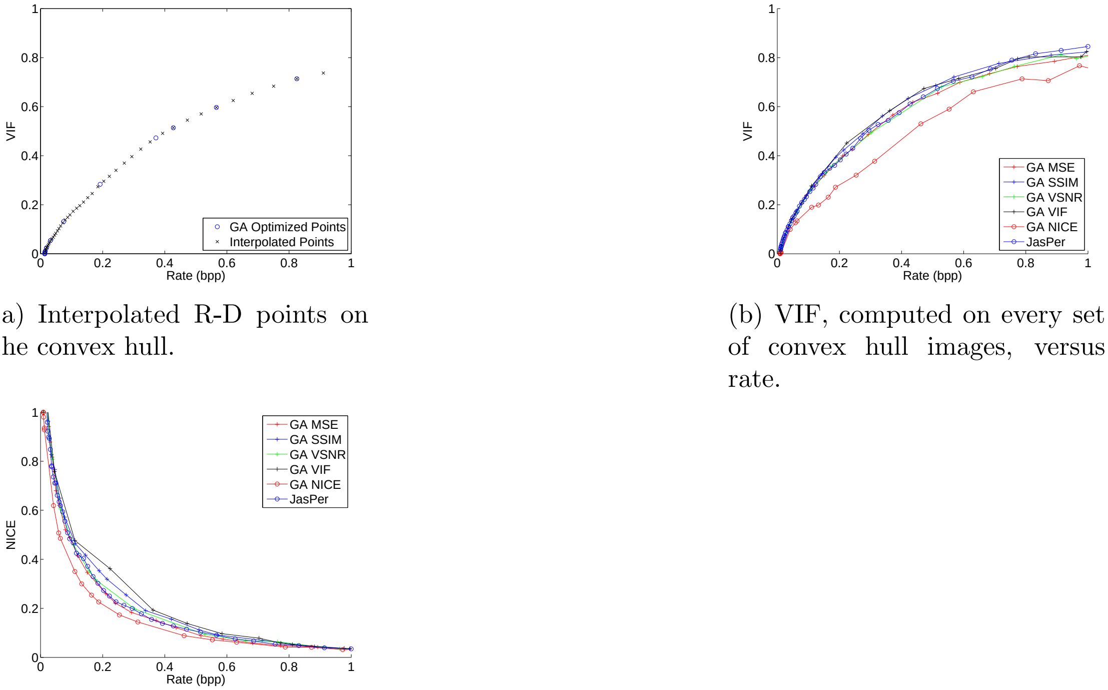
Figure B.1: A comparison of the R-D convex hull for 5 images using JasPer and the GA optimized for MSE. The nearly identical performance demonstrates that the GA properly converges to R-D optimal operating points.

tions of convex hull images for which there is significant disagreement between QEs, resulting in easily identifiable misclassification errors. Following a description of the heuristic R-D optimization in Section B.3, a collection of QEs are evaluated for inaccuracies. Section B.4 discusses a method for identifying false ties or false differences, by comparing images having a fixed QE score, while section B.5 demonstrates a method for identifying false ranks and false differences by comparing images at fixed bitrates.

B.3 Rate-Distortion optimization using a genetic algorithm

Operational R-D optimization can be formed as a Lagrangian minimization procedure, where the optimization selects a set of encoding parameters, $\vec{p}$, that satisfy the following,

$$\min_{\vec{p}} D(\vec{p}) + \lambda R(\vec{p}), \tag{B.1}$$



(a) Interpolated R-D points on the convex hull.

(b) VIF, computed on every set of convex hull images, versus rate.

(c) NICE, computed on every set of convex hull images, versus rate.

Figure B.2: Illustrations of the performance of the GA R-D optimization procedure. Novel points on the convex hull can be efficiently computed, shown in (a). In (b) and (c), the best performing set of images corresponds to the one that is optimized by the GA using the QE being computed.

where $D(\vec{p})$ and $R(\vec{p})$ are the computed distortion and rate for an image encoded with parameter $\vec{p}$. A target bitrate can be achieved by selecting the Lagrangian parameter, λ, such that $R(\vec{p}) \leq R^{target}$. The set of admissible parameters are defined by the encoding process. In the case of JPEG-2000, the encoder of choice for this work, $\vec{p}$ is typically the collection of truncation points for embedded bitstream coding techniques [78].

The goal of this work is to incorporate arbitrary distortion measures (or equivalently, QEs) into the R-D optimization framework. The embedded coding technique in JPEG-2000 (EBCOT) assumes an additive distortion measure, i.e., the total image distortion can be computed as the sum of distortions in individual code blocks, a constraint that is violated by many recent, perceptually motivated quality estimators.

As an alternative to EBCOT, JPEG-2000 allows for optimization via the quantization of wavelet transform coefficients. In this case, $\vec{p}$ corresponds to a vector of quantization stepsizes, one for each subband in the wavelet decomposition. The goal of this work is to incorporate an arbitrary QE into the minimization in Eq. (B.1). This assumption-free minimization requires a search over the space of possible parameters to identify operational points on the R-D convex hull.

The genetic algorithm (GA) is a heuristic, iterative optimization technique that efficiently searches a large space of parameter values [33]. At each iteration, the GA generates a population P of size M, where each member of the population is a particular realization of encoding parameters, i.e., $P(m) = \vec{p_m}$ and specifies quantization step sizes for each subband. Each population member has an associated cost, defined here as $C(m) = D(\vec{p_m}) + \lambda R(\vec{p_m})$, with λ fixed prior to beginning the search. Successive iterations in the GA merge population members with low cost, inject random variations into population members, and propagate the population members with the lowest cost, *evolving* toward the best population member in the search space. In this implementation, the population size is 100 and the maximum number of iterations is 500. The primary benefit of using the GA in this application is the flexibility in the choice of distortion measure $D(\vec{p_m})$. For each $\vec{p_m}$ in the population, a compressed image is generated by applying the quantization step sizes defined by $\vec{p_m}$. The distortion associated with $\vec{p_m}$ can be computed according to any model that generates a score given a compressed image (and optionally the corresponding reference image), including any full-reference or no-reference QE.

For a specified distortion measure, reference image, and λ, the GA identifies R-D optimal quantization step sizes. While the GA is very effective in its search, there is no optimality guarantee in the optimization. Despite this limitation, the GA optimization using MSE generates operational R-D points that lie on the R-D convex hull generated using JasPer [3] and the EBCOT algorithm, which has become a standard technique for operational MSE-based R-D optimization. Sweeping over a range of λ values allows the GA to generate a R-D convex hull that is optimized for MSE, as illustrated in Figure B.1 for 5 different reference images. Figure B.1 also includes the R-D convex hull generated using JasPer. Furthermore, provided that the set of λ values is sufficiently dense, novel points on the R-D convex hull can be computed using a bisection search on the quantization stepsizes associated with the nearest available points on the convex hull, illustrated in Figure B.2(a).

Given this framework for generating a set of R-D optimal quantization step sizes for an arbitrary distortion measure, the following QEs are evaluated: PSNR, SSIM [86], VSNR [15], VIF [71], NICE [65]. These QEs have publicly available implementations, facilitating their use in this application. The GA R-D optimization provides a collection of convex hull images that are optimal according to varying criteria, as illustrated by Figure B.2. By setting a fixed target value (QE or bitrate), comparisons can be made between the sets of convex hull images, identifying inaccuracies in the QEs being studied.

B.4 Comparing images for fixed QE scores

For a selected QE under test and desired target QE score, the image having the target QE score is extracted from each set of convex hull images. By design, the QE under test considers each image in this fixed-QE set to be perceptually identical (i.e., each has identical QE scores). Applying the remaining QEs to the fixed-QE set yields one of two results. In the first case, the remaining quality estimators are inconsistent with the QE under test and score the fixed-QE set as having varying perceptual quality. In the second case, the remaining quality estimators also exhibit fixed scores consistent with the QE under test. In the event of inconsistencies, if the images are perceptually different, there will be false ties in the fixed QE under test. If the images are perceptually equivalent, there will be false differences in the conflicting QE.

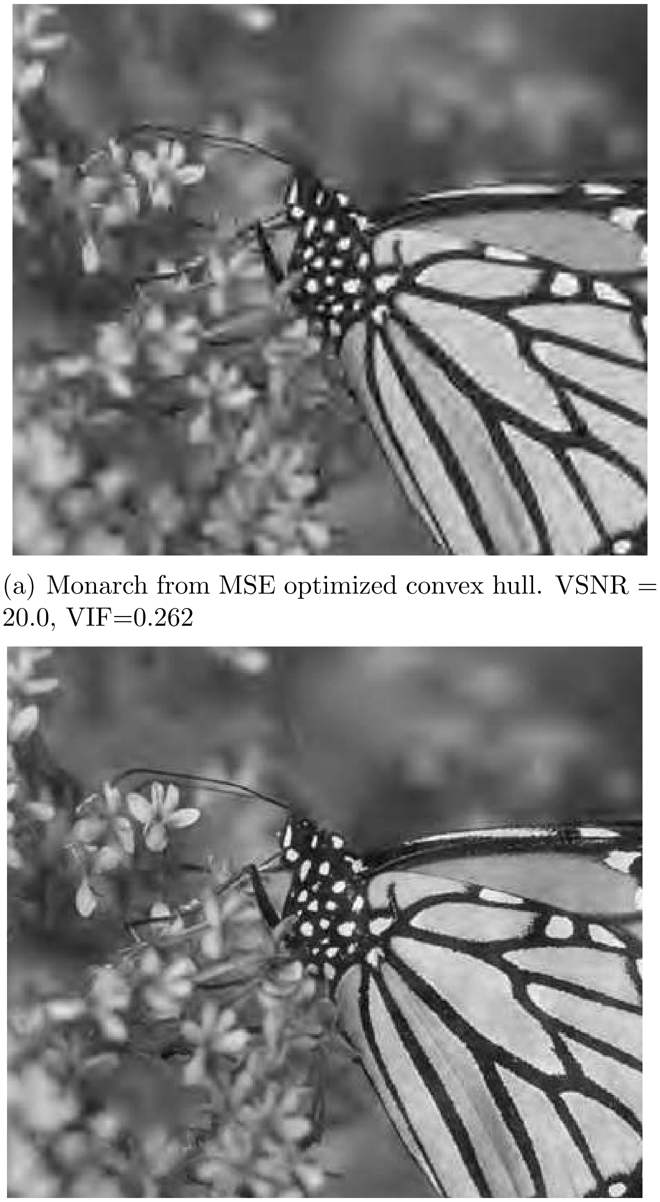
As an illustrative example, the *monarch* images having VSNR = 20 are extracted from each of the 6 sets. According to VSNR, each of these 6 images has the same perceptual quality. The set of QEs is applied to the fixed-VSNR images, yielding the scores provided in Table B.1. Also reported in Table B.1 is the coefficient of variation, computed as $c_v = \frac{\sigma}{\mu} \times 100$, which allows for the effective comparison of the variations in the quality scores when the means of the scores are significantly different. Both PSNR and SSIM have small coefficients of variation, indicating that these quality estimators are in agreement with VSNR. However, both VIF ($c_v = 21.9\%$ and NICE ($c_v = 30.8\%$) exhibit widely varying scores on the fixed-VSNR images. In this case, either VSNR, PSNR, and SSIM are creating a false tie or VIF and NICE are creating a false difference.

To determine the presence of a false difference or a false tie in the set of fixed-VSNR images without access to subjective quality ratings, the conflicting QE

Opt.	Computed QE - Fixed-VSNR				
Via	PSNR	SSIM	VSNR	VIF	NICE
NICE	29.7^{1}	0.921^{1}	20.0^{1}	0.442^{1}	0.123^{1}
VIF	28.4^{2}	0.885^{2}	20.0^{2}	0.306^{2}	0.450^{3}
JasPer	28.3^{3}	0.869^{5}	20.0^{3}	0.260^{5}	0.451^{4}
MSE	28.3^{4}	0.871^{4}	20.0^{4}	0.262^{4}	0.441^{2}
SSIM	28.2^{5}	0.877^{3}	19.8^{6}	0.274^{3}	0.464^{6}
VSNR	28.1^{6}	0.869^{6}	19.8^{5}	0.257^{6}	0.458^{5}
c_v	2.2%	2.1	1.7%	21.9%	30.8%

Table B.1: Scores for each tested QE on images constrained to have fixed QE score of VSNR = 20 for the monarch image. Each row represents a different set of convex hull images optimized via the GA. JasPer is the convex hull sets for this image coder. The final row provides the coefficient of variation, which is a normalized measure of the variation in the scores. The superscript numerals correspond to the rankings, as defined by the QE in the column heading.

(VIF) is selected as a proxy for true subjective quality. If the images are perceptually different, as suggested by VIF, then VSNR must be considered inaccurate in this case. The two images with the largest ΔVIF, each having VSNR = 20.0, correspond to the MSE-optimized image (VIF = 0.262) and the NICE-optimized image (VIF = 0.442). This image pair, provided in Figure B.3, clearly demonstrates a perceptual difference between the images; VIF correctly ranks this image pair while VSNR exhibits a false tie. In particular, note the amount of blurring in the high frequency regions of the flower petals and lines on the butterfly in Figure B.3(a). The image in Figure B.3(b) maintains the fidelity of the high frequency regions. The VSNR score is heavily impacted by the quantization of middle to low frequency subbands, which appears as slight contrast changes on the flat regions of the butterfly wings.



(a) Monarch from MSE optimized convex hull. VSNR = 20.0, VIF=0.262

(b) Monarch from NICE optimized convex hull. VSNR = 20.0, VIF=0.442

Figure B.3: An image pair having a fixed VSNR score and maximally different VIF scores. VIF more accurately reflects the perceived quality of this image pair.

B.5 Comparing images at fixed bitrates

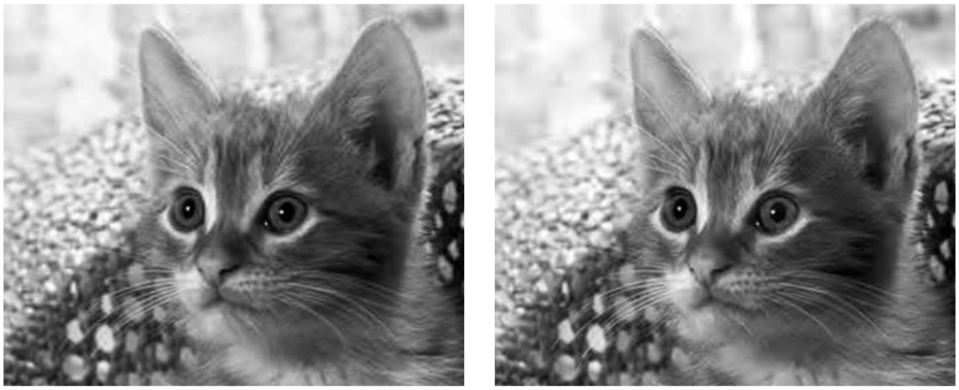
At a fixed bitrate, each quality estimator ranks the image from its own convex hull as having the highest quality. Comparing images across the convex hull sets for a fixed target bitrate yields a collection of fixed-rate images on which all the QEs disagree about the relative quality of the images. For example, Table B.1 provides the QE scores for the *cat2* image at 0.6 bpp. As expected, the QE scores are best when computed on the fixed-rate image from the set for which the QE was optimized. The image that is rated by humans as having the highest quality determines which QE that is most accurate, since that QE properly identifies the highest quality image. If the images are deemed perceptually equivalent, then each of the QEs exhibit false differences, the severity of which depends on the variations in the QE scores over the fixed-rate image collection. Determining the true subjective rankings requires a perceptual image quality experiment, a task which is left for future work. However, one illustrative examples is provided in Figure B.4, where the image pair corresponds to the VIF-optimized and NICE-optimized images, each at 0.6 bpp. Perceptual differences between this pair of images are very difficult to identify, suggesting that, in this case, VIF and NICE exhibit false differences.

B.6 Conclusion and Future Work

This work presented a method for identifying rate-distortion optimal quantization step sizes given an arbitrary distortion criterion. Several image quality estimators (QEs) were applied in this framework. By comparing image pairs that have a fixed target QE value or fixed bitrate, inaccuracies in QEs, such as false ties in VSNR and

Optimized	Computed QE				
Via	PSNR	SSIM	VSNR	VIF	NICE
JasPer	34.2 ^{1}	0.928 ^{3}	29.0 ^{2}	0.593 ^{4}	0.226 ^{3}
MSE	34.1 ^{2}	0.928 ^{4}	28.9 ^{3}	0.587 ^{5}	0.215 ^{2}
VSNR	33.8 ^{3}	0.929 ^{2}	29.0 ^{1}	0.598 ^{3}	0.248 ^{4}
SSIM	33.6 ^{4}	0.929 ^{1}	28.7 ^{4}	0.600 ^{2}	0.251 ^{5}
VIF	33.2 ^{5}	0.926 ^{5}	28.3 ^{5}	0.616 ^{1}	0.273 ^{6}
NICE	28.9 ^{6}	0.906 ^{6}	21.6 ^{6}	0.478 ^{6}	0.193 ^{1}
c_v	5.7%	0.9%	9.8%	8.2%	20.1%

Table B.2: Scores for each tested QE constrained to have fixed bitrates of 0.6 bpp for the cat image. Each row represents a different set of convex hull images optimized via the GA. JasPer is the convex hull sets for the image coder. The final row provides the coefficient of variation, which is a normalized measure of the variation in the scores. The superscript numerals correspond to the rankings, as defined by the QE in the column heading.



(a) Cat2 VIF optimized: VIF = 0.616, NICE=0.273

(b) Cat2 NICE optimized: VIF = 0.478, NICE=0.193

Figure B.4: An image pair that elicits a false difference in both VIF and NICE. Note that NICE is a distortion measure (larger values indicate lower utility). Consequently, NICE and VIF yield opposite (and false) rankings for these image pairs.

false differences in VIF and NICE, were identified. The proposed methodology will be extended to a wider variety of image quality estimators and coders, including JPEG. Furthermore, subjective image quality experiments will be performed using image pairs generating from this optimization.

BIBLIOGRAPHY

[1] Open source computer vision library.

[2] x264 (rev. 736). http://developers.videolan.org/x264.html.

[3] M. Adams and R. Ward. Jasper: a portable flexible open-source software tool kit for image coding/processing. In *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*, volume 5, May 2004.

[4] D. Agrafiotis, N. Canagarajah, D. R. Bull, J. Kyle, H. Seers, and M. Dye. A perceptually optimised video coding system for sign language communication at low bit rates. *Sig. Proc.: Image Comm.*, (21):531–549, 2006.

[5] G. Awad, J. Han, and A. Sutherland. A unified system for segmentation and tracking of face and hands in sign language recognition. In *International Conf. Pattern Recognition (ICPR 2006)*, volume 1, pages 239–242, 2006.

[6] C. Baker and C. A. Padden. Focusing on the nonmanual components of American Sign Language. *Understanding language through sign language research*, pages 27–57, 1978.

[7] M. Barkowsky, J. Bialkowski, B. Eskofier, R. Bitto, and A. Kaup. Temporal trajectory aware video quality measure. *IEEE Selected Topics in Signal Processing*, 3(2):266–279, 2009.

[8] R. Battison. *Lexical Borrowing in American Sign Language.* Linstok Press, Inc, Silver Spring, MD, 1978.

[9] M. H. Birnbaum, editor. *Psychological Experiments on the Internet.* Academic Press, San Diego, CA, 2000.

[10] R. A. Bradley and M. E. Terry. The rank analysis of incomplete block designs i: The method of paired comparisons. *Biometrika*, 39:324–345, 1952.

[11] D. H. Brainard. The psychophysics toolbox. *Spatial Vision*, 10:433–436, 1997.

[12] M. H. Brill, J. Lubin, P. Costa, S. Wolf, and J. Pearson. Accuracy and cross-calibration of video-quality metrics: new methods from ATIS/T1A1. *Signal Processing: Image Communication*, 19:101–107, Feb. 2004.

[13] A. Cavender, R. E. Ladner, and E. A. Riskin. MobileASL: Intelligibility of sign language video as constrained by mobile phone technology. In *Proc. International ACM SIGACCESS Conference on Computers and Accessibility (ASSETS 2006)*, pages 71–78, 2006.

[14] D. Chai and K. N. Ngan. Face segmentation using skin color map in videophone applications.

[15] D. Chandler and S. Hemami. VSNR: a Wavelet-Based visual Signal-to-Noise ratio for natural images. *IEEE Trans. Image Proc.*, 16(9):2284–2298, 2007.

[16] M. A. Changizi, A. Hsieh, R. Nijhawan, R. Kanai, and S. Shimojo. Perceiving the present and a systematization of illusions. *Cognitive Science: A Multidisciplinary Journal*, 32(3):459, 2008.

[17] N. Cherniavsky, A. Cavender, E. Riskin, and R. Ladner. Variable Frame Rate for Low Power Mobile Sign Language Communication. In *Proc. International ACM SIGACCESS Conference on Computers and Accessibility (ASSETS 2007)*, pages 163–170, 2007.

[18] N. Cherniavsky, J. Chon, J. Wobbrock, E. Riskin, and R. Ladner. Activity Analysis Enabling Real-Time Video Communication on Mobile Phones for Deaf Users. In *Proc. ACM Symposium on User Interface Software and Technology (UIST 2009)*, 2009.

[19] J. Chon, N. Cherniavsky, E. Riskin, and R. Ladner. Enabling access through real-time sign language communication over cell phones. *43rd Annual Asilomar Conference on Signals, Systems, and Computers*, 19(1), 2009.

[20] F. Ciaramello and S. Hemami. The influence of space and time varying distortions on objective intelligibility estimators for region-of-interest video. In *Proc. IEEE Int. Conf. Image Proc.*, 2010.

[21] F. Ciaramello and S. Hemami. An objective intelligibility measure for assessment and compression of American Sign Language video. *IEEE Trans. Image Proc.*, accepted for publication.

[22] F. Ciaramello, J. Ko, and S. Hemami. Quality versus intelligibility: Evaluating the coding trade-offs for American Sign Language video. *Proc. Information Sciences and Systems (CISS)*, Mar. 2010.

[23] F. M. Ciaramello and A. R. Reibman. Supplemental subjective testing to

evaluate the performance of image and video quality estimators. In *Human Vision and Electronic Imaging XVI*, January 2011.

[24] F. Ciaramello and S. Hemami. 'Can you see me now?' An objective metric for predicting intelligibility of compressed American Sign Language video. In *Proc. SPIE Human Vision and Electronic Imaging*, volume 6492, January 2007.

[25] F. Ciaramello and S. Hemami. Complexity constrained rate-distortion optimization of sign language video using an objective intelligibility metric. In *Proc. SPIE Visual Communication and Image Processing*, volume 6822, January 2008.

[26] F. Ciaramello and S. Hemami. Quantifying the effect of disruptions to temporal coherence on the intelligibility of compressed American Sign Language video. In *Proc. SPIE: Human Vision and Electronic Imaging*, volume 7240, February 2009.

[27] F. Ciaramello and S. Hemami. Quality versus intelligibility: Studying human preferences for American Sign Language video,. In *Proc. SPIE: Human Vision and Electronic Imaging*, volume 7865, January 2011.

[28] F. Ciaramello, R. Vanam, J. Chon, S. Hemami, E. Riskin, and R. Ladner. Rate-distortion-complexity optimization of an H.264/AVC encoder for real-time videoconferencing on a mobile device. In *Wkshp. on Video Proc. and Quality Metrics*, 2010.

[29] S. Daly, K. Matthews, and J. Ribas-Corbera. Face-based visually-optimized image sequence coding. In *Proc. IEEE Int. Conf. Image Proc.*, pages 443–447 vol.3, 1998.

[30] K. Emmorey, R. Thompson, and R. Colvin. Eye gaze during comprehension of American Sign Language by native and beginning signers. *J. Deaf Stud. Deaf Educ.*, 14(2):237–243, April 2009.

[31] R. A. Foulds. Biomechanical and perceptual constraints on the bandwidth requirements of sign language. *IEEE Trans. Neural Systems and Rehabilitation Engineering*, 12(1):65–72, March 2004.

[32] J. Fritsch, S. Lang, A. Kleinehagenbrock, G. Fink, and G. Sagerer. Improving adaptive skin color segmentation by incorporating results from face detection. In *Robot and Human Interactive Communication, 2002. Proceedings. 11th IEEE International Workshop on*, pages 337–343, 2002.

[33] D. E. Goldberg. *Genetic Algorithms in Search, Optimization, and Machine Learning.* Addison-Wesley Professional, 1 edition, January 1989.

[34] F. Grosjean. A study of timing in a manual and a spoken language: American Sign Language and English. *Journal of Psycholinguistic Research*, 8(4):379–405, July 1979.

[35] GSMA. General packet radio service, 2010.

[36] N. Habili, C. C. Lim, and A. Moini. Segmentation of the face and hands in sign language video sequences using color and motion cues. *IEEE Trans. Ckts. Syst. for Video Tech.*, 14(8):1086–1096, August 2004.

[37] A. Hadid, M. Pietikainen, and T. Ahonen. A discriminative feature space for detecting and recognizing faces. volume 2, pages II–797–II–804 Vol.2, 2004.

[38] J. C. Handley. Comparative analysis of bradley-terry and thurstone-mosteller paired comparison models for image quality assessment. In *PICS 2001: Image Processing, Image Quality, Image Capture, Systems Conference*, pages 108–112, 2001.

[39] J. Harkins, A. Wolff, E. Korres, R. Foulds, and S. Galuska. Intelligibility experiments with a feature extraction system designed to simulate a low-bandwidth video telephone for deaf people. In *Proc. RESNA Annual Conference*, volume 14, pages 38–40, 1991.

[40] S. S. Hemami and A. R. Reibman. No-reference image and video quality estimation: Applications and human-motivated design. *Signal Processing: Image Communication*, 25(7):469–481, August 2010.

[41] S. Hooper, C. Miller, S. Rose, and G. Veletsianos. The effects of digital video quality on learner comprehension in an American Sign Language assessment environment. *Sign Language Studies*, 8(1):42–58, 2007.

[42] ITU. *P.910: Subjective video quality assessment methods for multimedia applications.* September 1999.

[43] H. Kamphuis, H. Frowein, E. Rikken, and J. Spoor. Mobile videotelephony for deaf people: the effect of video quality on the use of text telephony. In *Vehicular Technology Conference, 1999. VTC 1999 - Fall. IEEE VTS 50th*, volume 2, pages 1082–1085 vol.2, 1999.

[44] M. Kleiner, D. Brainard, and D. Pelli. What's new in psychtoolbox-3? *Perception*, 36(ECVP Abstract Supplement), 2007.

[45] P. Kortum, W. S. Geisler, B. E. Rogowitz, and J. P. Allebach. Implementation of a foveated image coding system for image bandwidth reduction. In *Proc. SPIE: Human Vision and Electronic Imaging*, volume 2657, pages 350–360, San Jose, CA, USA, April 1996.

[46] P. Letellier, M. Nadler, and J.-F. Abramatic. The telesign project. *Proceedings of the IEEE*, 73(4):813 – 827, 1985.

[47] S. K. Liddell. Nonmanual signals and relative clauses in American Sign Language. *Understanding language through sign language research*, pages 59–90, 1978.

[48] S. K. Liddell and R. E. Johnson. American Sign Language: The phonological base. *Sign Language Studies*, 64:195–278, 1989.

[49] C.-W. Lin, Y.-J. Chang, and Y.-C. Chen. Low-complexity face-assisted video coding. In *Proc. IEEE Int. Conf. Image Proc.*, volume 2, pages 207–210 vol.2, 2000.

[50] M. D. Manoranjan and J. A. Robinson. Practical low-cost visual communication using binary images for deaf sign language. *IEEE Trans. Rehabilitation Engineering*, 8(1):81–88, 2000.

[51] L. Merritt and R. Vanam. Improved rate control and motion estimation for H.264 encoder. In *Proc. IEEE Int. Conf. Image Proc.*, volume 5, 2007.

[52] N. Moroney. Unconstrained web-based color naming experiment. In *Proc. SPIE Color Imaging: Device-Dependent Color, Color Hardcopy and Graphic Arts*, 2003.

[53] L. J. Muir and I. E. G. Richardson. Perception of sign language and its application to visual communications for deaf people. *J. Deaf Stud. Deaf Educ.*, 10(4):390–401, October 2005.

[54] K. Nakazono, Y. Nagashima, and A. Ichikawa. Digital encoding applied to sign language video. *IEICE Trans. Information & Systems*, E89-D(6), June 2006.

[55] A. Ninassi, O. L. Meur, P. L. Callet, and D. Barba. Considering temporal

variations of spatial visual distortions in video quality assessment. *IEEE Selected Topics in Signal Processing*, 3(2):253–265, 2009.

[56] S. C. Ong and S. Ranganath. Automatic Sign Language Analysis: A Survey and the Future beyond Lexical Meaning. *IEEE Trans. Pattern Analysis and Machine Intelligency*, 27(6):873–891, 2005.

[57] A. Ortega and K. Ramchandran. Forward-adaptive quantization with optimal overhead cost for image and video coding with applications to mpeg video coders. In *Proc. IS&T/SPIE Digital Video Compression*, February 1995.

[58] D. Parish, G. Sperling, and M. Landy. Intelligent temporal subsampling of american sign language using event boundaries. volume 16, pages 282–294, 1990.

[59] D. G. Pelli. The videotoolbox software for visual psychophysics: Transforming numbers into movies. *Spatial Vision*, 10:437–442, 1997.

[60] P. J. Phillips, H. Moon, S. A. Rizvi, and P. J. Rauss. The FERET evaluation methodology for face-recognition algorithms. volume 22, pages 1090–1104, 2000.

[61] S. Phung, S. Bouzerdoum, A., and S. Chai, D. Skin segmentation using color pixel classification: analysis and comparison. *IEEE Tran. Pattern Analysis and Machine Intelligence*, 27(1):148–154, January 2005.

[62] H. Poizner, U. Bellugi, and V. Lutes-Driscoll. Perception of american sign language in dynamic point-light displays. volume 7, pages 430–440, 1981.

[63] I. E. G. Richardson. *H.264 and MPEG-4 Video Compression.* John Wiley & Sons Ltd, West Sussex PO19 IUID, England, 2003.

[64] E. Riskin, S. Hemami, R. Ladner, and J. Wobbrock. The MobileASL project.

[65] D. Rouse and S. Hemami. Natural image utility assessment using image contours. In *Proc. IEEE Int. Conf. Image Proc.*, 2009.

[66] D. M. Saxe and R. A. Foulds. Robust region of interest coding for improved sign language telecommunication. *IEEE Trans. Information Technology in Biomedicine*, 6(4):310–316, December 2002.

[67] R. Schumeyer, E. Heredia, and K. Barner. Region of interest priority coding for sign language videoconferencing. In *IEEE Multimedia Signal Processing Workshop*, pages 531–536, 1997.

[68] G. M. Schuster and A. K. Katsaggelos. Fast and efficient mode and quantizer selection in the rate distortion sense for H.263. In R. Ansari and M. J. Smith, editors, *Proc. SPIE Vol. 2727,Visual Communications and Image Processing*, volume 2727, pages 784–795, February 1996.

[69] K. Seshadrinathan and A. C. Bovik. An information theoretic video quality metric based on motion models. In *Wkshp. on Video Proc. and Quality Metrics*, January 2007.

[70] H. R. Sheikh, M. F. Sabir, and A. C. Bovik. A statistical evaluation of recent full reference image quality assessment algorithms. *IEEE Trans. Image Proc.*, 15(11):3440–3451, Nov. 2006.

[71] H. Sheikh and A. Bovik. Image information and visual quality. *IEEE Trans. Image Proc.*, 15(2), 2006.

[72] Y. Shoham and A. Gersho. Efficient bit allocation for an arbitrary set of quantizers. *IEEE Trans. Acoustics, Speech and Signal Processing*, 36(9):1445–1453, September 1988.

[73] P. Siple. Visual constraints for sign language communication. *Sign Language Studies*, 19:95–110, 1978.

[74] G. Sperling, M. Landy, Y. Cohen, and M. Pavel. Intelligible encoding of ASL image sequences at extremely low information rates. *Computer Vision, Graphics, and Image Processing*, 31:335–391, 1985.

[75] C. Strobl, F. Wickelmaier, and E. Karls. Accounting for individual differences in bradley-terry models by means of recursive partitioning. Technical Report Number 54, University of Munich, 2009.

[76] T. Supalla and E. Newport. How many seats in a chair? The derivation of nouns and verbs in American Sign Language. In P. Siple, editor, *Understanding language through sign language research*, New York, NY, 1978. Academic Press.

[77] V. Tartter and K. Knowlton. Perception of sign language from an array of 27 moving spots. volume 289, pages 676–678, 1981.

[78] D. Taubman. High performance scalable image compression with ebcot. *IEEE Trans. Image Proc.*, 9(7):1158–1170, July 2000.

[79] R. D. Tweney, G. W. Heiman, and H. W. Hoemann. Psychological processing of sign language: Effects of visual disruption on sign intelligibility. *Journal of Experimental Psychology: General*, 106(3):255–268, 1977.

[80] C. Valli and C. Lucas. *Linguistics of American Sign Language: An Introduction.* Gallaudet University Press, Washington, DC, 2000.

[81] A. M. van Dijk and J. Martens. Subjective quality assessment of compressed images. *Signal Processing*, 58(3):235–252, May 1997.

[82] R. Vanam, E. A. Riskin, and R. E. Ladner. H.264/MPEG-4 AVC encoder parameter selection algorithms for complexity distortion tradeoff. In *Proc. of DCC*, Mar. 2009.

[83] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *Proc. Computer Vision and Pattern Recognition*, 2001.

[84] VQEG. Final report from the video quality experts group on the validation of objective models of video quality assessment. June 2000.

[85] VQEG. Final report from the video quality experts group on the validation of objective models of multimedia quality assessment, phase I. September 2008.

[86] Z. Wang, A. Bovik, H. Sheikh, and E. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Trans. Image Proc.*, 13(4):600–612, 2004.

[87] Z. Wang, L. Lu, and A. C. Bovik. Video quality assessment based on structural distortion measurement. *Sig. Proc.: Image Comm.*, 19(2):121–132, February 2004.

[88] T. Wiegand, M. Lightsone, D. Mukherjee, T. G. Camplbell, and S. Mitra. Rate-distortion optimized mode selection for very low bit rate video coding and the emerging H.263 standard. *IEEE Trans. Ckts. Syst. for Video Tech.*, 6(2), April 1996.

[89] T. Wiegand, G. Sullivan, G. Bjontegaard, and A. Luthra. Overview of the H.264/AVC video coding standard. *IEEE Trans. Ckts. Syst. for Video Tech.*, 13(7):560–576, Jul. 2003.

[90] T. Wiegand, H. Schwarz, A. Joch, F. Kossentini, and G. J. Sullivan. Rate-constrained coder control and comparison of video coding standards. *IEEE Trans. Ckts. Syst. for Video Tech.*, 13(7), July 2003.

[91] M.-H. Yang, N. Ahuja, and M. Tabb. Extraction of 2d motion trajectories and its application to hand gesture recognition. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 24(8):1061–1074, August 2002.

[92] M. Yuen and H. R. Wu. A survey of hybrid MC/DPCM/DCT video coding distortions. *Signal Process.*, 70(3):247–278, 1998.