JOINT RATE-INTELLIGIBILITY-COMPLEXITY OPTIMIZATION OF AN H.264 VIDEO ENCODER FOR AMERICAN SIGN LANGUAGE

Rahul Vanam[†], *Eve A. Riskin*[†], *Richard E. Ladner*[§]

[†]Dept. of Electrical Engineering,
[§]Dept. of Computer Science and Engineering,
University of Washington, Seattle, WA 98195
{rahulv, riskin}@ee.washington.edu, ladner@cs.washington.edu

Francis M. Ciaramello, Sheila S. Hemami

School of Electrical and
Computer Engineering,
Cornell University, Ithaca, NY, 14853
fmc3@cornell.edu, hemami@ece.cornell.edu

ABSTRACT

This paper presents an H.264 standard-compliant video encoder optimized for region-of-interest (ROI) based coding tuned to American Sign Language (ASL) videos. Encoding modes are developed which allow the encoder to allocate both rate and computational resources differently across the ROIs. An objective measure of intelligibility is included in an encoder parameter optimization by modifying a fast offline distortion-complexity optimization algorithm, resulting in parameter selections that demonstrate excellent rate-intelligibility-complexity performance. These parameters can be stored in a look-up table for use by an online algorithm which selects parameters based on available computational resources. The resulting parameter selections improve the encoder speed by up to 21.2% with a small decrease in intelligibility over the x264 default parameter settings.

1. INTRODUCTION

Cell phone technology has become ubiquitous due to its convenience and mobility. Current video cell phones are equipped with a camera and codecs, and have the potential for use in real-time mobile videoconferencing. However, the availability of high bandwidth 3G networks is limited to few cities in the United States, ultimately requiring such a system to operate at very low bandwidths. Furthermore, real-time capture, encoding, and transmission of digital video is difficult on devices with limited computational resources, such as mobile phones. This motivates the need for low complexity video compression algorithms which can provide video that is useful to the end user. In the past, the perceptual quality of videoconferencing has been improved by reducing distortions in the user's face [1, 2]. Region-of-interest (ROI) based video compression can be extended to American Sign Language (ASL) video. For ASL video, an observer is tracking the signer's face and hands and evaluating distortions only in those regions. This is supported by both the linguistic structure of sign language [3] (e.g. how information is conveyed) and by eye-tracking experiments [4]. Because of this unique structure, several specialized algorithms have been proposed for encoding sign language video [4, 5, 6].

In the author's previous work, an ASL optimized video encoder was developed using an objective measure of intelligibility incorporated into an H.264 rate-distortion (R-D) optimization algorithm [6]. A performance bound for the system is obtained using the Viterbi algorithm to search over all possible quantization parameters and encoding modes. For fixed levels of intelligibility, bitrate can be reduced by as much as 60% over an R-D optimization algorithm which measures distortion as MSE. The goal of this work is to achieve as much of this gain as possible while maintaining a computational complexity appropriate for low mobile devices with low processing power.

Traditionally, ROI-optimized encoders achieve bitrate savings by allocating rate only to the most relevant regions. In this work, we extend this concept and also allocate more computational resources to these important regions. Two additional encoding options are presented which allow variations in encoding complexity based on the relative importance of each macroblock. A fast offline algorithm is then used to search the space of possible encoding parameters available in H.264, including the proposed ROI-tuned options, to find parameters that give us improvement in encoding speed with only small decreases in intelligibility. The results demonstrate that appropriate parameter selections improve the encoder speed by up to 21.2% with a small decrease in intelligibility when compared to the x264 default parameter settings.

2. SIGN LANGUAGE INTELLIGIBILITY-OPTIMIZED VIDEO ENCODER

The ASL optimized encoder is implemented within x264 [7], an open-source H.264 encoder. The rate-distortion optimization uses an objective intelligibility measure, which is a function of the distortion in linguistically relevant regions and ac-

This work is supported by the National Science Foundation under grant numbers CCF-0514353, CCF-0514357 and IIS-0811884.

curately predicts an observer's subjective intelligibility rating [3]. Each frame of the input sequence is segmented into the signer's face, hands, torso, and background, using color-based skin detection and morphological processing. This segmentation operates in real-time on a mobile device [8]. Given the region segmentation for a particular frame, the distortions affecting intelligibility are computed as the weighted combination of the mean squared error (MSE) in the face, hands, and torso of the signer:

$$D = W_F MSE_F + W_H MSE_H + W_T MSE_T, \quad (1)$$

where $W_F = 1.6$, $W_H = 0.5$, and $W_H = 0.1$. Because of the varying weights, a particular MSE in the signer's face will result in a higher total distortion than the same MSE in the signer's torso.

The ASL-tuned distortion measure in (1) is incorporated into a rate-distortion (R-D) optimization procedure similar to that of [9] and applied to a collection of ASL videos. For a given Lagrangian λ, the parameter p that includes motion vector, mode and quantization step size (QP) is chosen such that it minimizes the joint R-D cost $J(X,p) = D(X,p) + \lambda R(X,p)$, where X is a particular macroblock. A consequence of using the distortion measure in 1 is that more rate is inherently allocated to the important regions (i.e., face, hands, and torso). The work presented in [6] identified a functional relationship between λ and the resulting optimal QPs. Ultimately, this allows for fast encoding by using a single parameter λ to quickly select a QP value for each of the region types. The motion vector and mode for each macroblock are still selected according to the minimum R-D cost. Rate control is performed at the frame-level by adjusting the Lagrangian parameter λ, according to $\lambda(n+1) = \lambda(n) - R_{target}/R_{actual}$, where R_{target} and R_{actual} are the target bits and actual bits for frame n.

3. ROI-BASED COMPLEXITY ALLOCATION ENCODER OPTIONS

Four variable encoding parameters are varied to achieve different points in rate-distortion-complexity: 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. A maximum of 16 reference frames can be specified using `ref`. Ten different `part` options specify the partition size from 4×4 and above for intra (I), predictive (P) and bi-predictive (B) macroblocks [10]. 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).

Two additional encoding parameters are added to the x264 encoder that allow the encoding complexity to vary on a per-block basis, depending on the type of region being encoded (e.g. face, hand, or background). The first parameter (`backgrd-part`) restricts the partition search performed by the encoder in background blocks. In H.264, as many as 12-15 different modes need to be analyzed for a given macroblock. Since distortions in background macroblocks do not contribute to the overall distortion measure in Equation (1), 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 face and hand blocks and one for background blocks. When using both the `backgrd-part` and `part` parameters, the encoder uses `backgrd-part` option for background macroblocks and `part` option for the face and hands macroblocks. For ease of integration into the pre-existing encoder structures, the `backgrd-part` has the same 10 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.

In motion-compensated video coding, searching for optimal motion vectors comprises a significant portion of the total encoding time. To speed up the motion search, a parameter (`ROI-ME`) is included that specifies a potentially different motion search method for the face, hands, torso and background macroblocks. This approach was demonstrated to improve the encoding speed of the x264 encoder by up to 12% for ASL videos [11]. The space of possible x264 motion search methods are listed in increasing order of complexity: diamond (DIA), hexagon (HEX) and uneven multihexagon (UMH) search. The background macroblocks use only the DIA search, while the torso region uses equal or lower complexity search compared to the face and hand regions. The `ROI-ME` includes the following 8 options $(1, \ldots, 8)$ corresponding to the motion search in (face, torso, background) regions: (HEX, UMH, DIA), (UMH, HEX, DIA), (HEX, HEX, DIA), (UMH, DIA, DIA), (HEX, DIA, DIA), (DIA, DIA, DIA), (UMH, UMH, DIA) and (UMH, UMH, UMH).

For each of the encoding parameters, the options are indexed in order of increasing complexity. For example, a value of `part` = 10 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 reduce the overall rate-distortion performance.

4. JOINT RATE-INTELLIGIBILITY-COMPLEXITY OPTIMIZATION

The set of encoding options discussed in Section 3 made available to the encoder determine the achievable bitrate, distortion, and complexity. A vector of parameter options is referred to as *parameter settings*. The x264 default parameter setting is the vector (`subme` = 5, `part` = 8, `ref` = 1, `trellis`= 1, `backgrd-part`=0, `ROI-ME`=0). These values are also listed in Table 2. This parameter vector corresponds to high complexity sub-pixel motion estimation; all possible 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. An ideal video encoder will select the parameter setting which results in a compressed video that meets the target rate and complexity constraints in an optimal way. An exhaustive search over all possible parameter settings requires 358400 encodings per video ($7 \times 16 \times 10 \times 4 \times 10 \times 8$). Because it is impossible to perform an exhaustive search of this rate-distortion-complexity space in real-time on a mobile device, fast and accurate methods for choosing the appropriate set of encoding parameters must be employed.

The dominant parameter setting pruning algorithm (DPSPA) [10] is applied to determine 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 distortion-complexity convex hull. For a fixed bitrate, DPSPA provides a collection of parameter settings which correspond to operating points lying approximately on the distortion-complexity (D-C) convex hull, as illustrated in Figure 1. 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 effectively provides a look-up table that specifies the parameter settings to use such that distortion is minimized given both a rate and complexity constraint.

5. EXPERIMENTAL RESULTS: APPLYING DPSPA TO ASL VIDEO SET

The DPSPA algorithm is applied to a set of nine training ASL videos and six test ASL videos each having 320×240 frame resolution, 200 frames and a frame rate of 15 fps. The results are reported for fixed values of λ. Fixing λ for a collection of sequences effectively fixes the QP value in each frame. This results in slight variations in rate and is an approximation of a constant rate scenario. Results are currently being generated at fixed values of bitrate using the rate control algorithm described in Section 2 and will be presented at the workshop. These experiments are conducted on a Windows XP PC having a 2.01 GHz AMD processor.

The DPSPA algorithm is executed for values of $\lambda = \{0.5, 1, 5, 40, 150\}$ corresponding to average bitrate of {215.5, 155.8, 74.2, 28.8, 16.1} kb/s for the default parameter setting on test videos. The optimal parameter settings computed by DPSPA are applied to the test set of ASL videos to obtain the maximum and average encoding speed improvement, the relative gain in bitrate and the loss in intelligibility of DPSPA parameter setting over the default parameter setting. As demonstrated in Table 1, the DPSPA parameter settings provide average speed improvements of approximately 15% with little decrease in intelligibility. A difference of approximately 1.5 dB corresponds to a statistical change in subjective intelligibility score [6]. Therefore, the average decreases in intelligibility shown in Table 1 will not significantly reduce the perceived intelligibility.

The reductions in complexity result from choosing an optimal combination of encoding parameters, including the use of the additional coding modes tuned for region-based coders. Table 2 lists the encoder parameter settings which have been applied to the test videos. While the default parameter setting

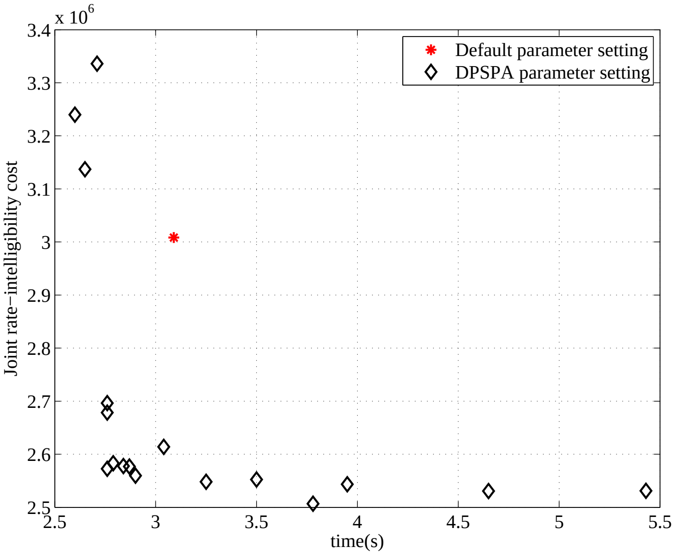

Fig. 1. Joint R-D_I cost vs. encoding time for an ASL training video. DPSPA provides an approximation of the convex hull of the R-D-C space.

Table 1. Relative performance of DPSPA parameter setting over the default parameter setting for ASL test videos. A negative bitrate gain implies lower bitrate for DPSPA. Tests were performed on a Windows XP PC with 2.01 GHz AMD processor.

λ	Avg. rate gain	Max speed gain	Avg. speed gain	Avg. Intell. loss (dB)
0.6	-2.4%	15.7%	14.4%	0.08
1.1	-0.47%	19.2%	16.8%	0
5	-26.8%	21.2%	13.7%	0.16
46	0.49%	20.1%	15%	0.67
150	-0.66%	18%	13.7%	0.22

Table 2. The default parameter setting (fixed) and DPSPA parameter settings for different values of λ.

Parameter name	Default	λ_{DPSPA}				
		0.6	1.1	5	46	150
`subme`	5	3	3	3	2	2
`ref`	1	3	1	1	1	1
`part`	8	6	8	8	3	3
`trellis`	1	0	0	2	2	2
`backgrd-part`	–	8	7	3	6	7
`ROI-ME`	–	6	5	6	6	6

uses HEX search for the entire frame, the DPSPA parameter setting exploits the `ROI-ME` option, using either (HEX, DIA, DIA) or (DIA, DIA, DIA) for (face, torso, background), each of which have lower search complexity than the default setting. DPSPA often chooses lower complexity `subme` and `backgrnd-part` compared to the default `subme`=5 and P8×8, I8×8, I4×4 for background macroblocks. Since DPSPA generates parameter settings that trade-off joint rate-intelligibility cost with encoding time, it does not always pick parameters having lower complexity than the default setting. For example in Table 2, for $\lambda_{DPSPA} = 0.6$, DPSPA picks three reference frames instead of one reference frame. The additional computation cost is mediated by the corresponding reduction in distortion.

6. CONCLUSION AND FUTURE WORK

This paper presents an ASL encoder based on the H.264 standard in which both rate and complexity can be allocated to the region-of-interest. The proposed encoder includes two new parameters that specify the partition size for the background macroblocks and ROI-based motion search complexity. DPSPA, a fast offline algorithm, is used to choose parameter settings that have excellent rate-intelligibility-complexity performance. These parameter settings can be stored in a look-up table that can be used by an online algorithm which chooses parameter settings based on the available computational resources and bandwidth. When compared to the default parameter settings, the DPSPA parameter settings gives up to 21.2% improvement in encoding speed with a small decrease in intelligibility.

DPSPA quickly provides a collection of parameter settings which are nearly optimal in terms of rate, distortion, and complexity. Real-time video encoding on a mobile device imposes constraints on both bitrate and encoding speed. Bitrate is a function of the available network bandwidth, which will vary depending on network load and geographical location. The encoding complexity depends on the maximum processing power of a given device and on the remaining battery life. Given these constraints on bitrate and encoding speed, the offline training results from DPSPA can be stored in a look-up table on the mobile device, allowing extremely fast selection of optimal encoding parameters based on the current network and processing resources. The ASL optimized encoder is currently being ported to an HTC TyTN II cell phone, being integrated into the MobileASL application in order to validate the results computed by DPSPA in the offline training. [12].

7. REFERENCES

[1] D. Chai and K N. Ngan, "Face segmentation using skin color map in videophone applications," in *IEEE Trans. Circuits and Systems for Video Technology*, 1999, vol. 9, pp. 551–564.

[2] S. Daly, K. Matthews, and J. Ribas-Corbera, "Face-based visually-optimized image sequence coding," in *Proc. IEEE International Conference on Image Processing (ICIP'98)*, 1998, pp. 443–447 vol.3.

[3] F. M. Ciaramello and S.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 '09*, 2009, vol. 7240.

[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," in *Signal Processing: Image Commun.*, 2006, number 21, pp. 531–549.

[5] K. Nakazono, Y. Nagashima, and A. Ichikawa, "Digital encoding applied to sign language video," in *IEICE Trans. Inf. & Sys.*, June 2006, vol. E89-D.

[6] F. M. Ciaramello and S. S. Hemami, "Complexity constrained rate-distortion optimization of sign language video using an objective intelligibility metric," in *Proc. SPIE, Visual Communication and Image Processing '08*, Jan. 2008, vol. 6822.

[7] "x264," http://developers.videolan.org/x264.html.

[8] F. M. Ciaramello and S.S. Hemami, "Real-time face and hand detection for videoconferencing on a mobile device," in *Workshop on Video Processing and Quality Metrics for Consumer Electronics*, Scottsdale, AZ, January 2009.

[9] 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. of IS&T/SPIE Digital Video Compression '95*, February 1995.

[10] 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.

[11] R. Vanam, E. A. Riskin, R. E. Ladner, and S. S. Hemami, "Fast parameter setting selection algorithms for distortion-complexity optimization of h.264 encoder," *IEEE TCSVT (in preparation)*.

[12] Eve Riskin, Sheila Hemami, and Richard Ladner, *The MobileASL Project*, http://www.cs.washington.edu/research/MobileASL/index.html.