RATE-DISTORTION-COMPLEXITY OPTIMIZATION OF AN H.264/AVC ENCODER FOR REAL-TIME VIDEOCONFERENCING ON A MOBILE DEVICE

Rahul Vanam, Jaehong Chon
Eve A. Riskin, Richard E. Ladner

University of Washington, Seattle, WA 98195

Francis M. Ciaramello, Sheila S. Hemami

Cornell University, Ithaca, NY, 14853

ABSTRACT

This paper presents an H.264 standard-compliant video encoder optimized for region-of-interest (ROI) based coding and applied to American Sign Language (ASL) videos. Three encoding parameters are developed which allow the encoder to allocate both rate and computational resources differently between the ROI and non-ROI. 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-distortion-complexity performance. These parameters can be stored in a look-up table for use by an online algorithm that selects parameters based on available computational resources. The offline training is performed both on a PC and a cell phone. The resulting parameter selections improve the encoder speed by up to 54.4% on the PC and 62.1% on the cell phone with a small decrease in intelligibility over the x264 default parameter setting.

1. INTRODUCTION

Current video cell phones, equipped with a camera and codecs, 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 a mobile videoconferencing system to operate at very low bandwidths. Furthermore, real-time capture, encoding, and transmission of digital video is difficult on mobile phones that have limited computational resources. This motivates the development of low complexity video compression algorithms.

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] 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 sign intelligibility incorporated into an H.264 rate-distortion (R-D) optimization algorithm [6]. For fixed levels of intelligibility, the ASL encoder provided bitrate reduction up to 60% over a traditional bitrate-MSE optimized video encoder. The goal of this work is to achieve as much of this gain as possible while maintaining a computational complexity appropriate for mobile devices with low processing power.

In this paper, we propose an ROI-optimized video encoder that includes three new ROI-based parameters that allow variations in encoding complexity per macroblock based on its relative importance. We use a fast offline algorithm to search the space of all possible encoder parameters, to find parameters that yield improvement in complexity (encoding speed), with only small decreases in intelligibility. This approach improves the encoder speed on the PC and cell phone platforms by an average of 45% and 52% (up to 54.4% and 62.1%), respectively, with negligible loss in intelligibility when compared to the x264 default parameter setting.

The rate-distortion-complexity (R-D-C) optimization performed in this work is evaluated specifically for ASL video, but the same procedure can be applied to any class of ROI video (e.g. videoconferencing), subject to the availability of a distortion measure that reflects the relative importance of each region. In this work, the face, hands and torso will be referred to as ROI and the background as non-ROI.

This paper is organized as follows. Section 2 describes the ROI-optimized video encoder used in this work. Section 3 describes the collection of encoding parameters available to the encoder and the additional ROI-based encoding parameters. The performance of the additional encoding parameters are evaluated in Section 4. Finally, Sections 5 and 6 describe the search technique and present the complexity versus distortion performance.

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

2. ROI-OPTIMIZED VIDEO ENCODER

The ROI-optimized encoder, applied to ASL video, is implemented within x264 [7], an open-source H.264 encoder. The R-D optimization uses an objective intelligibility measure, which is a function of the distortion in linguistically relevant regions and accurately 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. Given the region segmentation for a particular frame, the distortions affecting intelligibility are computed as a function of the weighted combination of the spatio-temporal distortions in the face, hands, and torso of the signer:

$$D_I = W_F D_F + W_H D_H + W_T D_T, \tag{1}$$

where $W_F = 1.6$, $W_H = 0.5$, and $W_H = 0.1$ [8]. Because D_I is a distortion measure it is inversely proportional with intelligibility. The varying weights control the relative importance of each type of macroblock in the ROI; a distortion in the signer's face will result in a lower intelligibility than the same level of distortion in the signer's torso.

The distortion measure D_I in Equation (1) is incorporated into a R-D optimization procedure similar to that of [9] and applied to a collection of ASL videos. A consequence of using D_I is that more rate is inherently allocated to the ROI [6]. For a given Lagrangian λ, the encoding decisions d_e that includes motion vector, mode and quantization step size (QP) is chosen such that it minimizes the joint R-D_I cost $J(X, d_e) = D_I(X, d_e) + \lambda R(X, d_e)$, where X is a particular macroblock.

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, defined for the entire frame, to quickly select a QP value for each macroblock, depending on the region types. The motion vector and mode for each macroblock are still selected according to the minimum R-D_I cost. One-pass rate control is performed at the frame-level by adjusting λ according to $\lambda(n+1) = \lambda(n) - (R_{target}/R_{actual} - 1)$, where R_{target} and R_{actual} are the target bits and actual bits for frame n [10]. For the first frame, λ is initialized using an estimate provided by x264.

3. ROI-BASED COMPLEXITY ALLOCATION PARAMETERS

Implementing the ROI encoder as an extension of x264 allows for the use of all the encoding parameters available to x264, the selection of which provides a tradeoff 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 [11]. 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). We define a vector of parameter options 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 paper, the average encoding time is used as a measure of complexity.

Three additional 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, as many as 12-15 different partitions need to be analyzed for a given macroblock. Our first parameter, `nonROI-part`, restricts the partitions used by the encoder for the background blocks. 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 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) [7].

For the specific case of ASL video, the distortion in the signer's face has the largest impact on the overall intelligibility. To ensure lower distortion for the face region, we choose a motion search method for the face region having equal or higher complexity compared to the hand and torso regions. 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.

4. RESULTS: 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 ROI-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 3, without any of the ROI complexity modes enabled. For each fixed bitrate, D_I 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 ROI-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 distortion, the ROI-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.

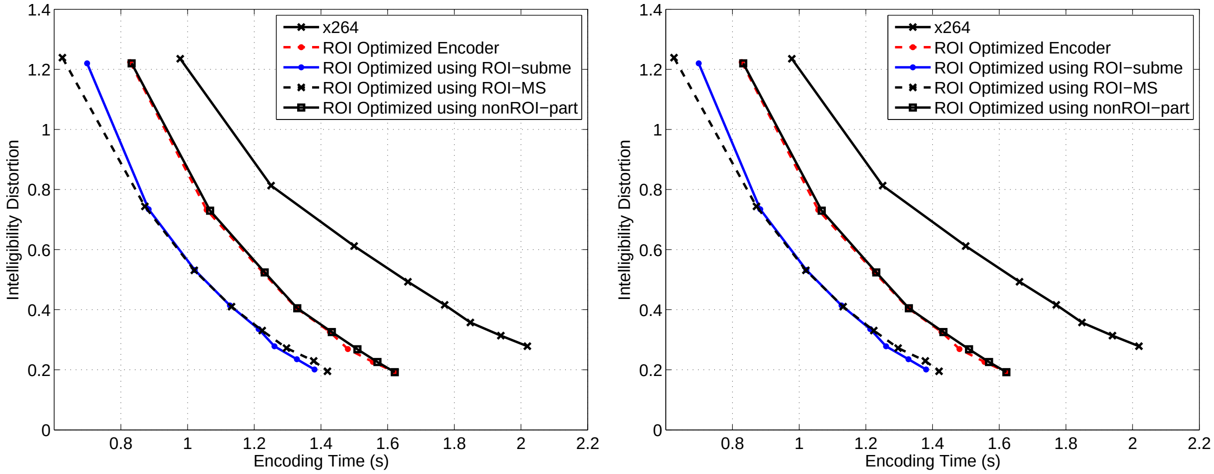
To achieve the same level of distortion, x264 must operate at a higher bitrate, because it allocates rate to the non-ROI and the ROI indiscriminately, whereas the ROI encoder allocates rate almost entirely to the ROI. The complexity gains provided by using the ROI-optimized encoder can be attributed to high distortion in the non-ROI. When using the ASL intelligibility distortion measure in Equation (1) for computing R-D cost, distortions in the non-ROI do not contribute to the distortion measure. 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 ROI-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 ROI-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 3 are all set to their highest complexity.

As illustrated in Figure 1(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 ROI-optimized encoder when using the highest complexity settings. Figure 1(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.

5. JOINT RATE-DISTORTION-COMPLEXITY OPTIMIZATION

The H.264 coding standard only specifies the operation of the decoder, leaving virtually infinite flexibility in the operation of the encoder. The set of encoding parameters dis-



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

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

Fig. 1. The R-D-C space for 5 different encoding scenarios. The x264 encoder and the ROI-optimized encoder, each running with the highest complexity settings, provide benchmark performance levels. The three ROI parameters are compared against the benchmarks.

cussed in Section 3 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. In our case, 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 on a mobile device, fast methods for choosing the appropriate set of encoding parameters must be employed.

The dominant parameter setting pruning algorithm (DPSPA) [11] 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 2. 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 rate and complexity.

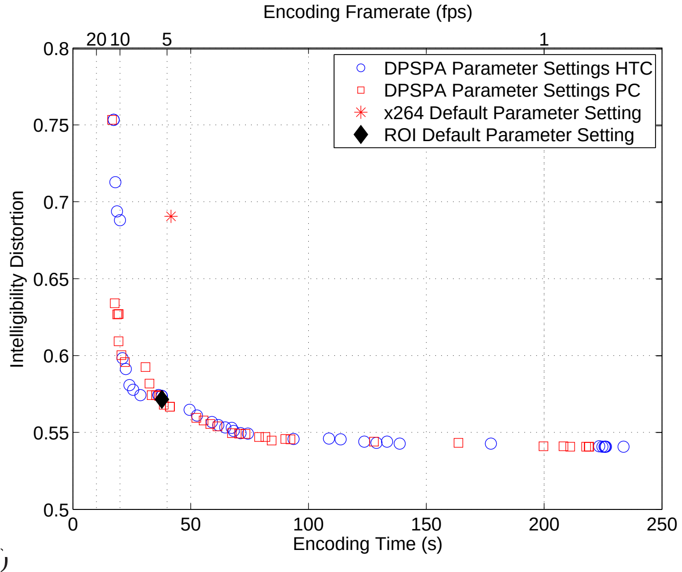


Fig. 2. Intelligibility distortion vs. encoding time (lower y-axis) and the corresponding encoding frame rate (upper y-axis) for the outdoor ASL training set at 30 kbps, running on the HTC TyTN II cell phone. DPSPA parameter settings obtained on either the PC or the cell phone have similar performance on the cell phone.

6. RESULTS: APPLYING DPSPA TO ASL VIDEOS

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 400 MHz Qualcomm MSM7200 ARMv6 processor.

The x264 default parameter setting is the vector (`subme` = 5, `ref` = 1, `part` = (P8 $\times$ 8,B8 $\times$ 8, I8 $\times$ 8,I4 $\times$ 4), `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 $D_I(p)$ and $C(p)$ correspond to the intelligibility distortion and encoding time of a parameter setting p. We define the change in intelligibility as $\Delta D_I = D_I(default) - D_I(DPSPA)$ and speed gain = $\frac{(C(default)-C(DPSPA))}{C(default)} \times 100$.

As demonstrated in Tables 1 and 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 [8]. Therefore, the average decreases in intelligibility shown in Tables 1 and 2 will not significantly reduce the perceived intelligibility.

Tables 1 and 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 ROI-optimized encoder can use very coarse, low-complexity parameter options. The overall speed improvement of the ROI-optimized encoder depends on the relative level of activity in the non-ROI.

Tables 1 and 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. We apply this default parameter setting to our ROI-optimized encoder and call it 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 2. 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 10fps, the nominal limit for full ASL conversations. [12]

Table 1. Intelligibility distortion difference (ΔD_I) 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 ΔD_I indicates a higher intelligibility distortion for DPSPA.

Bitrate	Indoor		Outdoor		Indoor & Outdoor	
(kbps)	ΔD_I	speed gain	ΔD_I	speed gain	ΔD_I	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%

Table 2. Intelligibility distortion difference (ΔD_I) 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)	ΔD_I	speed gain	ΔD_I	speed gain	ΔD_I	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%

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 2 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.

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 tradeoff 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.

We compare the x264 default parameter setting with the DPSPA parameter setting having comparable D_I 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 entroby 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. CONCLUSION

This paper presents a region-of-interest (ROI) encoder based on the H.264 standard in which both rate and complexity can be allocated to the ROI. The proposed encoder includes three new parameters that specify the level of sub-pixel motion estimation in the ROI (`ROI-subme`), the partition size restrictions for non-ROI macroblocks (`nonROI-part`), and the ROI-based motion search complexity (`ROI-MS`). This ROI-optimized encoder is evaluated for American Sign Language video. The `ROI-subme` and `ROI-MS` parameters reduce the encoding speed by 16% without affecting the R-D performance.

DPSPA, a fast offline algorithm, is used to choose parameter settings that have excellent rate-intelligibility-complexity performance. These settings can be stored in a look-up table that can be used by an online algorithm which chooses parameter settings based on the currently available computational resources and bandwidth. When compared to the x264 default parameter settings, the DPSPA parameter settings gives up to 54.4% improvement in encoding speed on a PC and 62.1% improvement in encoding speed on a cell phone with a small decrease in intelligibility. Additionally, computing the look-up table on the PC platform and implementing it on the cell phone results in the same relative performance as when the look-up table is computed on the cell phone platform. Performing the training and developing a look-up table can be executed much more rapidly on a PC and simply deployed to the cell phone.

8. 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 Communication*, 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. Ciaramello and S. S. Hemami, "An objective intelligibility measure for assessment and compression of American Sign Language video," *(in preparation)*.

[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] T. Wiegand, M. Lightsone, D. Mukherjee, T. George Camplbell, and S. Mitra, "Rate-distortion optimized mode selection for very low bit rate video coding and the emerging H.263 standard," in *IEEE Trans. Circuits and Systems for Video Technology*, April 1996, vol. 6.

[11] 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.

[12] 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 *Proceedings of RESNA Annual Conference*, 1991, vol. 14, pp. 38–40.