Prepared (Subject resp)		No.		
EAB/TXV Thomas Rusert		EAB-15:005754 Uen		
Approved (Document resp)	Checked	Date	Rev	Reference
		2015-03-09	PA3	

Prepared:
EAB/TXV Kenneth Andersson
EAB/TXV Thomas Rusert

Guided transcoding for ABR video distribution

Abstract

This report addresses the subject of efficient distribution of two or more different coded adaptive bit rate (ABR) video representations from a content delivery origin to one or more content delivery edges. A guided transcoding approach is introduced as a method for reducing bandwidth requirements for the video transmission between origin and edge while introducing comparably modest computational requirements at the edge. Basic idea is to send only the highest quality video representation together with some side information from origin to edge and to create the additional lower quality representations at the edge nodes by means of a transcoding operation that is guided by the side information. The side information enables the edge transcoding to be performed with significantly less complex computations than full edge transcoding.

Evaluations are performed for a configuration with seven HEVC-encoded ABR representations, including one 1080p representation, two 720p representations, two 540p representations, and two 360p representations. The results show that compared to a baseline scenario where no edge transcoding is performed, the bandwidth requirements between origin and edge can be reduced by about 37% while introducing transcoding-induced compression efficiency loss of about 9% on average (11.5%/8.9%/6.5% for 720p/540p/360p, applying for transmission of transcoded lower-quality representations). Considering an Intel Core i5-2540M “Sandy Bridge” CPU with 2.6GHz (from 2011) and an SIMD-optimized HEVC implementation, real-time transcoding of 1080p60 video to a single lower resolution representation (720p, 540p, or 360p) is estimated to fully load roughly 1-2 CPU cores. Transcoding 1080p60 to all six lower representations at the same time is estimated to fully load in the order of 5 CPU cores.

The results are specific for the selected set of ABR video representations (i.e. video resolutions and quality levels) and the selected encoding settings, and there are several trade-offs to make between origin-to-edge savings, transcoding-induced efficiency loss, and transcoding complexity.

Additional simulations are presented to verify the results for guided transcoding when using H.264. Experiments settings are based on a commercial ABR encoding profile. Guided transcoding is emulated by using a modified x264 encoder. The results are roughly in line with the results for HEVC. The complexity for guided transcoding is expected to be similar for H.264 and HEVC. The complexity gain with guided transcoding over full transcoding is expected to be less when H.264 is used as compared to HEVC.

1 Introduction

This report addresses the subject of efficient distribution of two or more different coded adaptive bit rate (ABR) video representations from a content delivery origin to one or more content delivery edges. A guided transcoding approach is introduced as a method for reducing bandwidth requirements for the video transmission between origin and edge while introducing comparably modest computational requirements at the edge. More specifically, the origin and the edge could be co-located with the origin and the edge of a content delivery network (CDN), wherein ABR clients could be adaptively requesting segments of the different representations from the CDN edge, choosing representations depending on individually available access bandwidth.

Figure 1 depicts the baseline for the comparisons made in this study. The original video is encoded and transmitted in multiple (here 3) different representations with different bit rates R_i (often referred to as “simulcast”). Each representation has an index (here 0, 1, 2), where the quality of the representations is assumed to increase with decreasing index. Thus representation 0 is the highest quality representation. The different representations could have identical video resolutions or they could differ in video resolutions, in which case downsampling operations would be performed on the source video. It can be noted that the bandwidth required between the origin and edge is the sum of bit rates from all individual representations.

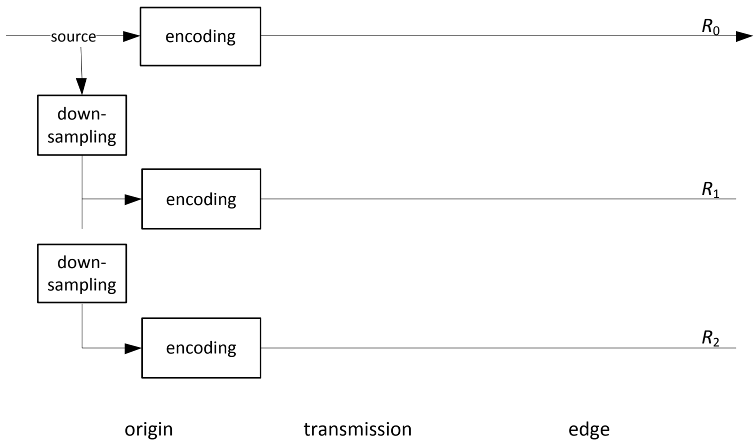

Figure 1: Baseline scenario.

Figure 2 depicts a scenario where a full transcoder is introduced at the edge in order to reduce the bandwidth requirements for the video distribution from origin to edge, while still providing the same number of representations as in the baseline scenario for subsequent consumption. Only the highest quality representation is transmitted to the edge, and the lower quality representations are generated at the edge by transcoding the highest quality representation into lower quality representations, i.e. decoding the highest quality representation, potentially downsampling the decoded video, and re-encoding it for each lower quality representation.

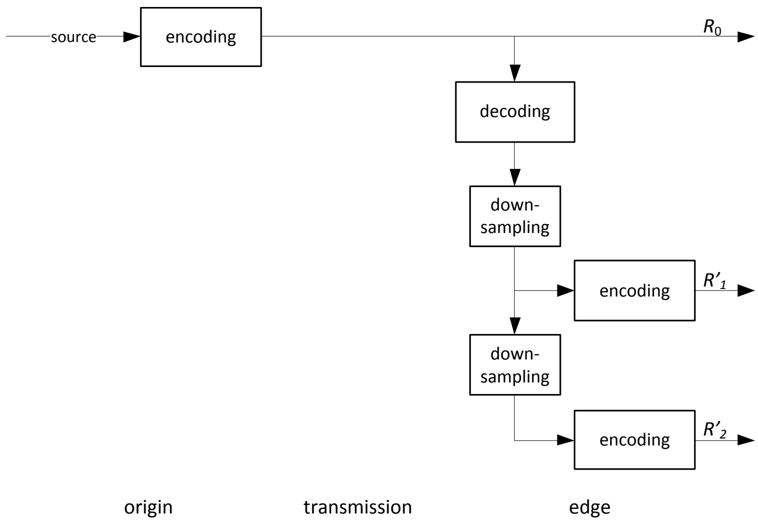

Figure 2: Full transcoding scenario.

It can be noted that the bandwidth required for transmission from origin to edge is reduced at the cost of introducing computations at the edge. Furthermore, although not depicted in the figure, in order for the lower quality representations to provide the same quality level as the lower quality representations in the baseline scenario, slightly higher bit rates ($R'_1 > R_1$, $R'_2 > R_2$) are typically required due to the fact that the lower quality representations are encoded based on an already compressed video source.

The encoding part of the transcoding process is computationally very demanding, typically fully loading tens of high-end CPU cores in order to achieve good compression performance for encoding a single HD video representation in real-time or near-real-time. If multiple lower quality representations need to be created, the computational demands for all necessary encodings add up.

Figure 3 depicts the guided transcoding scenario which is subject to evaluation in this study. It extends the full transcoding scenario by introducing a specific encoder (“motion encoding”) for each lower quality representation at the origin. The “motion encoding” includes all traditional encoder building blocks such as closed-loop inter/intra prediction, transform, quantization and entropy coding, motion estimation, mode decision and rate control. The only difference from regular encoding is that in the bitstream generated at the output of the motion encoder, transform coefficients are entirely omitted. In other words, in terms of HEVC, the bitstream contains all tree block partitioning information, mode information, motion vectors, transform parameters, quantization parameters and in-loop filtering parameters (those elements are in the following collectively denoted as “motion information”) but no transform coefficient data. The motion information is transmitted to the edge transcoder, and used in a guided encoding process, in which instead of determining the motion information in a motion estimation process, the received motion information is applied as-is. Thus, “guided encoding” includes most traditional encoder building blocks such as closed-loop inter/intra prediction, transform, quantization and entropy coding, however it does not include motion estimation, mode decision and rate control, which are instead performed in the corresponding “motion encoding” process and then transmitted to the guided transcoder. Since closed-loop guided encoding is performed, no drift is induced.

Note that in order for the motion information generated in the “motion encoding” process to match the lower quality representations correctly, and to avoid drift, each motion encoder at the origin site and its associated guided encoder at the edge site must be operated in sync, which means that all operations including generation of the source video, inter/intra prediction, transform, quantization and reconstruction must be performed identically. For this reason the reconstruction of the encoded high quality video is utilized as source for motion estimation and inter-picture prediction in the “motion encoding” for the lower quality representations[1].

[1] Actually those motion estimation processes can additionally utilize the original video source in order to further slightly improve the quality of the motion information (not depicted in the figure but utilized in the investigations). Specifically, the original video source can be used when calculating the distortion term used in the (RDO-based) mode decision, while still using reconstructed pictures as references for inter prediction (so as to avoid drift).

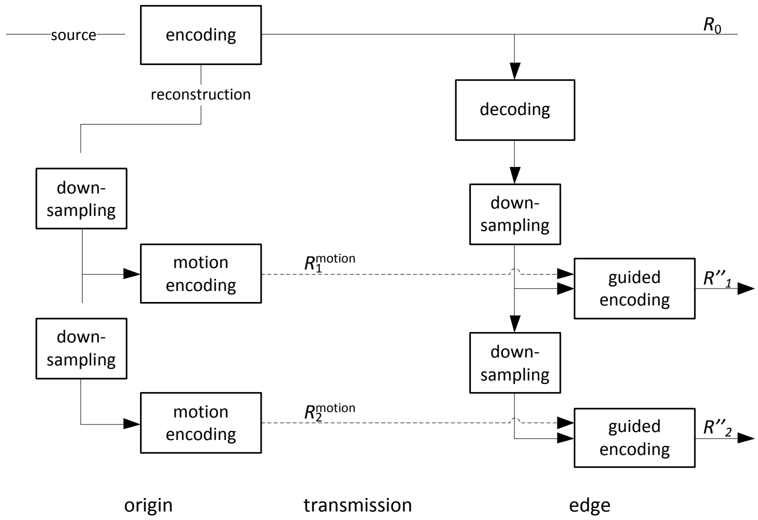

Figure 3: Guided transcoding scenario.

Compared to the full transcoding scenario, since determining the motion information is otherwise the heavy part of the transcoding process, the guided transcoding approach “shifts” computational complexity from the edge to the origin. Consequently, the complexity at the edge is significantly reduced while it is increased at the origin. Shifting the complexity in that way can be overall beneficial in a CDN scenario since there would be a single origin node but many edge nodes, meaning that compared to full transcoding, the origin complexity increase would apply once and the edge complexity decrease would apply many times.

Furthermore, compared to full transcoding, additional bandwidth is required for transmission of the additional motion information from the origin to the edge. However, since only motion information needs to be transmitted for each of the lower quality representations, the overall bandwidth requirement is lower than for the baseline scenario. Furthermore, similar as in the full transcoding scenario, in order for the lower quality representations to provide the same quality level as the lower quality representations in the baseline scenario, slightly higher bit rates ($R''_1>R_1$, $R''_2>R_2$) are typically required due to the fact that the lower quality representations are encoded based on an already compressed video source.

Note that the guided transcoding operation could be performed “on demand”, i.e. at the time when a lower quality representation is requested from the edge. In that case, assuming a pre-populated cache is associated with the edge, only the highest quality representation and the lower quality motion information would be stored. Then, when a lower quality representation is requested for the first time, the guided transcoding process could be performed for that representation, and the resulting transcoding bit stream could be stored for later usage. Alternatively, the guided transcoding could be performed for all lower quality representations at the time when the cache is pre-populated.

Further note that in a live scenario, any sort of edge transcoding (guided or unguided) will likely introduce additional end-to-end delay. Assuming segment-based processing, the additional delay is expected to correspond to at least one segment duration (caused by the need to fully receive the high quality source segment, and transcode it afterwards). Sub-segment based processing may reduce the additional delays.

The guided transcoding principle can be applied for any video codec. Even a mix of different video codecs could be used, i.e. some or all lower quality representations could be encoded using different video codecs than the highest quality representation. In this study, HEVC is used as video codec. The following sections focus on evaluating the reduction of bandwidth requirements between origin and edge, as well as on the computational complexity of the guided transcoding process.

2 Compression efficiency tests

2.1 Test settings

Compression efficiency tests have been performed using the HEVC reference software HM as well as a custom guided transcoding implementation based on HM and the SHVC reference software SHM.

Five 1080p test sequences from the common test set used in SHVC standardization in JCT-VC have been utilized as follows:

- Kimono@24Hz
- ParkScene@24Hz
- Cactus@50Hz
- BasketBallDrive@50Hz
- BQTerrace@60Hz

For each sequence, four test encodings were performed using a random-access coding configuration based on common conditions in SHVC standardization, with four different quantizer settings, QP = 34,30,26,22.

For each of the test encodings, six lower quality representations were produced, resulting in a total of seven quality representations with resolution and quantizer settings as follows:

- 1080p, QP
- 720p, QP
- 720p, QP+2
- 540p, QP
- 540p, QP+2
- 360p, QP
- 360p, QP+2

In order to generate the lower quality representations, downsampling was performed with a 7-bit approximation of the 8-bit filters coefficients in the SHM downsampling tool[2]. The filters have 12 taps[3].

2.2 Test results

Table 1 depicts the average bit rates for the seven different sequences after normal HEVC encoding (i.e. corresponding to the baseline scenario) according to common standardization test conditions, averaged over QP=34,30,26,22 for each sequence.

In order to obtain numbers for guided transcoding, a custom HM-based motion encoder has been utilized to generate motion information for each lower quality representation, obtaining a motion information bit rate R_i^{motion} for each test case. In a first experiment, the "motion encoder" used was employing HM-like RDO, optimizing the RD cost considering both motion information and transform coefficients (even though transform coefficients are not included in the bitstream), i.e. the RD cost corresponding to the resulting bitstream <u>after</u> guided transcoding. The resulting bit stream has been utilized in the guided transcoding process, obtaining a resulting transcoded bit rate, R_i'', and a PSNR value for the decoded transcoded bit stream, PSNR''.

[2] The implementation is based on SHM-4.0 which deploys 7-bit upsampling filtering to allow 16 bit accumulators.

[3] Replacing the 12-tap filters with 8-tap filters for luma and 4-tap filters for chroma has shown to produce similar results as using 12-tap filters when evaluating the performance against originals that have been downsampled using 12-tap filters. Reducing the number of filter coefficients reduces the number of multiplication and additions and can thus reduce the impact of downsampling to the total complexity of guided transcoding.

Table 2 depicts the average bit rate reduction compared to normal HEVC encoding when encoding only motion information for the lower quality representations. In order to obtain the numbers, the PSNR values for the decoded transcoded bit streams (relative to the original, potentially downsampled, source video), denoted as PSNR'', have been obtained, and a corresponding bit rate R has been obtained by interpolating the (R, PSNR) results from the normal HEVC encodings using a 3rd order polynomial interpolation and matching the PSNR value. Then with R_i^{motion} the bit rate of the motion information bit stream, the bit rate reduction for each representation has been determined as $(R_i\text{-}R_i^{motion})/R_i$. As can be seen from the table, the amount of bit rate spent on the motion information is around 50% of the corresponding full HEVC bit stream. The total bit rate reduction depicted in the table is determined as $(R^{HEVC}\text{-}R^{GT})/R^{HEVC}$, where R^{HEVC} corresponds on the total bit rate required for transmitting all representations in the baseline scenario, and R^{GT} corresponds to the total bit rate required for transmitting the highest quality representation and the motion information for all lower quality representations in the guided transcoding scenario. The average total bit rate reduction is 32.6%[4].

Table 2 also depicts the theoretical maximum bit rate reduction, which is 61.8% on average. That number corresponds to the case where no side information is sent from the origin to the edge, i.e. only the highest quality representation is transmitted, which corresponds to the full transcoding scenario in Section 1.

Table 3 depicts the bit rate overhead required for the lower quality representations after transcoding, compared to normal HEVC encoding. The numbers are obtained by comparing the bit rates R_i'' after guided transcoding, and the PSNR'' after decoding the transcoded video, and determining a corresponding bit rate R for normal HEVC encoding for the same PSNR value, using interpolation in the same way as above. The average bit rate overhead for the transcoded representations is 8.0%.

[4] Note that since the total bit rate reduction calculation includes all bits transmitted from origin to edge, including the bits for the highest quality representation and the bits for the motion information related to the lower quality representations, and the bit rate reduction compared to the baseline scenario applies only to the information associated with the lower quality representations, the bit rate reduction generally increases when the number of lower quality representations is increased.

The reason for the overhead associated with the transcoded video is twofold. Firstly, the lower quality representations are encoded based on an already compressed video source. Secondly, according to test conditions in SHVC standardization, the normal HEVC encodings are made using RDOQ (Rate Distortion Optimized Quantization), however RDOQ is turned off for the motion encoding in the guided transcoding experiments in order to keep the complexity of the guided transcoding process low[5]. The impact of re-encoding compressed content is going down as the difference between source and target quality increases. This is reflected by the results in Table 3, where the overhead goes down as the quality of the target video goes down, i.e. the overhead is 10.6%, 8.0% and 5.5% on average for 720p, 540p, and 360p, respectively. The impact of RDOQ, based on results not shown in this report, is expected to be in the range of few percent (up to around 4-5%).

In a second experiment, the RD cost function in the RDO of the motion encoders was modified to consider both, the RD cost after guided transcoding and the bit rate cost for transmission of motion information from origin to edge. Specifically, a new cost function $D+\lambda_1 R''+ \lambda_2 R^{motion}$ was employed, where $D+\lambda_1 R''$corresponds to the cost function used in the first experiment (with D the distortion, R'' the bit rate after transcoding, including both motion information and transform coefficients, and λ_1 the Lagrangian parameter), R^{motion} the bit rate for the motion information only, and $\lambda_2=\lambda_1/2$. Goal of the approach is to increase the bit rate savings for the transmission from origin to edge at the cost of slight additional bit rate overhead required for the lower quality representations after transcoding. $\lambda_2=\lambda_1/2$ was experimentally found to provide a reasonable trade-off. Table 4 and Table 5 depict results for origin-to-edge savings and lower quality representation overhead, respectively. The average total bit rate reduction is 37.0% (up from 32.6% in Table 2) and the average bit rate overhead for the transcoded representations is 9.0% (up from 8.0% in Table 3). Again, in Table 5, it can be noted that the overhead goes down as the quality of the target video goes down, i.e. the overhead is 11.5%, 8.9% and 6.5% on average for 720p, 540p, and 360p, respectively.

[5] Note that unlike in a previous version of this report, the reported results were obtained by using the HEVC SDH (Sign Data Hiding) tool, which is part of the common conditions in SHVC standardization, in the guided transcoding process. The assumption made here is that the additional encoder complexity associated with SDH is negligible in the guided transcoding process.

Table 1: Average bitrates [kbps] for HEVC encoding at QP 22, 26, 30, 34.

Sequence and quantizer setting	1080p QP	720p QP	720p QP+2	540p QP	540p QP+2	360p QP	360p QP+2
Kimono	2391	1392	1035	1002	745	544	406
ParkScene	3706	1877	1382	1196	888	486	366
Cactus	7730	3349	2487	2219	1674	1050	806
BasketBallDrive	7612	3438	2578	2345	1776	1170	894
BQTerrace	13643	2908	1938	1643	1168	565	418
Average	**7016**	**2592**	**1884**	**1681**	**1250**	**763**	**578**

Table 2: Average total bitrate reduction for “motion information only” versus simulcast HEVC encoding.

Sequence	720p	720p QP+2	540p	540p QP+2	360p	360p QP+2	Total	Max (theoretical)
Kimono	57%	57%	57%	56%	59%	56%	38,98%	68,27%
ParkScene	53%	52%	57%	54%	58%	56%	34,20%	63,24%
Cactus	52%	50%	54%	51%	55%	52%	32,59%	63,42%
BasketBallDrive	49%	47%	51%	49%	52%	48%	31,30%	64,23%
BQTerrace	53%	52%	55%	53%	55%	53%	25,74%	49,56%
Average	**53%**	**52%**	**55%**	**53%**	**56%**	**53%**	**32,57%**	**61,75%**

Table 3: Average bitrate cost of transcoded representations versus HEVC.

Sequence	Cost 720p QP	Cost 720p QP+2	Cost 540p QP	Cost 540p QP+2	Cost 360p QP	Cost 360p QP+2	Average
Kimono	12,87%	9,29%	10,22%	7,94%	6,47%	5,46%	8,71%
ParkScene	10,63%	7,71%	7,57%	5,83%	5,39%	4,31%	6,91%
Cactus	11,83%	8,65%	8,57%	6,73%	6,08%	5,20%	7,84%
BasketBallDrive	13,32%	9,75%	9,88%	7,77%	6,70%	5,68%	8,85%
BQTerrace	12,90%	8,89%	8,96%	6,37%	5,55%	4,08%	7,79%
Average	**12,31%**	**8,86%**	**9,04%**	**6,93%**	**6,04%**	**4,94%**	**8,02%**

Table 4: Average bitrate reduction for "motion information only" versus simulcast HEVC encoding, using alternative RDO cost function in the motion encoding.

Sequence	720p	720p QP+2	540p	540p QP+2	360p	360p QP+2	Total	Max (theoretical)
Kimono	64%	65%	65%	64%	66%	64%	44,13%	68,27%
ParkScene	61%	60%	63%	62%	65%	63%	39,00%	63,24%
Cactus	59%	57%	61%	58%	62%	59%	37,11%	63,42%
BasketBallDrive	56%	55%	58%	56%	59%	55%	35,91%	64,23%
BQTerrace	59%	59%	61%	59%	61%	59%	28,77%	49,56%
Average	**60%**	**59%**	**62%**	**60%**	**63%**	**60%**	**36,99%**	**61,75%**

Table 5: Average bitrate cost of transcoded representations versus HEVC, using alternative RDO cost function in the motion encoding.

Sequence	Cost 720p QP	Cost 720p QP+2	Cost 540p QP	Cost 540p QP+2	Cost 360p QP	Cost 360p QP+2	Average
Kimono	14,25%	10,96%	11,46%	9,35%	7,78%	6,84%	10,11%
ParkScene	10,96%	8,02%	8,00%	6,30%	5,99%	4,90%	7,36%
Cactus	12,52%	9,52%	9,39%	7,52%	7,08%	6,23%	8,71%
BasketBallDrive	14,63%	11,22%	11,25%	9,23%	8,30%	7,33%	10,33%
BQTerrace	13,28%	9,36%	9,44%	6,75%	6,00%	4,48%	8,22%
Average	**13,13%**	**9,82%**	**9,91%**	**7,83%**	**7,03%**	**5,96%**	**8,95%**

3 Estimation of transcoding complexity

The guided transcoding operation for a picture consists of the following steps, see Figure 3:

1. Decoding picture of the highest quality representation and storing it in the decoded picture buffer (DPB) for the highest quality representation.

2. For a given target resolution, downsampling of the decoded highest quality picture to the target resolution (using identical downsampling method as used when encoding at the origin).

3. Decoding the motion information associated with the target quality and target picture.

4. For each block in the target picture, performing the following steps a-c. Storing the resulting picture in the DPB for the target quality.

    a. Applying the motion information for motion-compensated prediction using reference pictures in the DPB for the target quality.

    b. Obtaining a prediction residual (using downsampled highest quality picture as source).

    c. Encoding the prediction residual together with the motion information (using identical quantization method as the encoder).

We estimate the computational complexity in units of decodings of the highest quality representation (DHU).

By definition, step 1 has a complexity of 1 DHU. With C_1 denoting the normalized complexity of step 1, C_1=1.

Table 6 depicts complexity estimates for steps 2-4. The motivation for the numbers is as follows.

Step 2 consist of pixel-wise filtering and is estimated to have similar complexity as the motion compensation process (8-tap filter for luma and 4-tap filter for chroma). In [1] motion compensation is estimated to contribute to around 50% of the total decoding time. The fraction of motion compensation on the total complexity is reduced as the bitrate goes up since then the portion of residual decoding is increased, thus 50% may be seen as an upper limit. Using similar filter length for downsampling as for motion compensation, it is assumed that downsampling is 50% as complex as decoding[6]. Using this assumption, the complexity for 720p is C_2=50%*1/2.25=0.222 (2.25 is the number of pixels for 1080p divided by the number of pixels for 720p), and accordingly, C_2=50%*1/4=0.125 for 540p, and C_2=50%*1/9=0.0556 for 360p.

In step 3, we estimate the decoding complexity for the side information to be similar as the complexity of full decoding for the target resolution[7].

In step 4, we estimate the encoding complexity to be similar as the complexity for decoding of the target resolution, thus the complexity is the same as in step 3.

[6] Note that reducing the number of filter coefficients by applying a bi-linear filter instead of an 8-tap filter could reduce the complexity of the downsampling to something like 50%/4=12.5% of the decoding time..

[7] Since about 20% of the decoding time is due to entropy decoding according to [1] and considering that most entropy decoding is due to decoding of residual, the actual complexity for decoding the side information should be less than in the estimation, i.e. C_3 in Table 6 can be seen as an upper limit.

Table 6: Complexity estimates for the guided transcoding steps.

	720p	540p	360p
C_1	1	1	1
C_2	0.222	0.125	0.0556
C_3	0.444	0.25	0.111
C_4	0.444	0.25	0.111
C_{total}	**2.11**	**1.625**	**1.2776**

The decoding of the highest quality representation can be reused when transcoding into several representations at the same time. The total complexity for transcoding into six representations with two bitrates for each target resolution is about $C_{total}=1+0.222+2*2*0.444+0.125+2*2*0.25+0.0556+2*2*0.111=4.623$.

The actual time for decoding the highest quality bit stream (i.e. the time associated with $C_1=1$) is estimated by applying a single threaded OpenHEVC decoder (a SIMD optimized HEVC decoder) on the highest quality bit streams produced above[8]. The experiment is conducted on a PC laptop with Intel Core i5-2540M “Sandy Bridge” with clock speed of 2.6GHz (CPU released in 2011) using a 64-bit Windows 7 and 4GB RAM. Results are depicted in Table 7.

[8] Command line: hevc_ffmpeg_w64.exe –i bitstream

Table 7: Measurement of decoding time using OpenHEVC. Speed numbers are given as multiples of real-time, i.e. a speed number of 2.5 indicates that decoding of a 10sec clip required 4s.

Sequence	Bitrate [kbps] QP26	Speed QP26	Bitrate [kbps] QP30	Speed QP30	Bitrate [kbps] QP34	Speed QP34
Kimono@24Hz	2527	2,5	1435	2,9	811	3,1
ParkScene@24Hz	3898	2,1	2115	2,8	1130	2,8
Cactus@50Hz	6868	1,2	3591	1,7	2036	1,6
BasketBallDrive@50Hz	7142	1,1	3774	1,3	2153	1,6
BQTerrace@60Hz	9930	0,9	3439	1,0	1579	1,6
Average	**6073**	**1,6**	**2871**	**1,9**	**1542**	**2,1**

From the results in the table, it can be deduced that with the given configuration, it is possible to decode a 1080p60 video on a single CPU core roughly in real-time. Thus considering C_{total}=4.623, real-time guided transcoding from 1080p60 to six lower quality representations as depicted above would load about 4.6 CPU cores.

4 Discussion of results

The results are specific for the selected set of ABR video representations (i.e. video resolutions and quality levels) and the selected encoding settings, and there are several trade-offs to make between origin-to-edge savings, transcoding-induced efficiency loss, and transcoding complexity.

The bandwidth savings between origin and edge can be expected to increase as the number of ABR video representations increases. Furthermore, the transcoding-induced bitrate overhead is expected to go down as the difference between highest quality video and target video increases.

Potential options for optimizing the overall performance include the following:

- Introducing RDOQ in the guided transcoding process is expected to reduce the transcoding-induced bitrate overhead by few percent, at the cost of higher transcoding complexity.

- Using simulcast instead of guided transcoding for representations (or segments of representations, or pictures) where the overhead is high would reduce the average overhead and the transcoding complexity at the cost of reduced origin-to-edge bandwidth savings.

- Using simulcast instead of guided transcoding for representations (or segments of representations, or pictures) where the origin-to-edge bandwidth savings are low could reduce the transcoding complexity without strongly affecting the overall origin-to-edge savings.
- The RD cost function could potentially be further optimized to trade origin-to-edge savings against transcoding overhead.

5 Additional results for H.264

5.1 Introduction

Additional simulations are presented to verify the results for guided transcoding when using H.264. The experimental settings are based on a commercial ABR encoding profile. Guided transcoding is emulated by using a modified x264 encoder.

5.2 Experimental settings

- Encoding profile based on commercial encoding profile provided by Fabrix (hereafter referred to as “Fabrix profile”), with some modifications as follows
- Test sequences as specified in Section 2.1
- Resolutions 1080p, 720p, 540p, 360p (instead of 1080p, 720p, 486p, 360p in Fabrix profile), one representation per resolution (i.e. 4 representations in total)
- Frame rates 24Hz, 50Hz and 60Hz (instead of 30Hz in Fabrix profile)
- Intra period 2s, hierarchical B picture prediction with three B pictures
- H.264 High profile for 1080p, 720p, H.264 Main profile for 540p, 360p
- x264 (fetched from http://git.videolan.org, 2014-12-20), preset “slow”
- Trellis quantization disabled for guided transcoding, enabled for normal encoding (baseline scenario)
- Constant quality setting (--crf) with quality points determined as follows
  - Coding with quantizer setting to match bit rate for 1080p with bit rate from Fabrix profile, calling the resulting quantizer setting QP
  - For 720p, 540p and 360p, coding with four different quantizer settings QP, QP+1, QP+2 and QP+3; averaging the results for the four settings

Formatted: Heading 2, Don't adjust space between Latin and Asian text

Formatted: List Bullet 2

Formatted: List Bullet 2

Formatted: List Bullet 2, Outline numbered + Level: 2 + Numbering Style: Bullet + Aligned at: 5,14 cm + Tab after: 5,77 cm + Indent at: 5,74 cm

5.3 Compression efficiency results

Table 8 depicts the average bitrates obtained for x264 encoding, as well as the reference bitrates from the Fabrix profile. As can be seen, all bitrates except for the 720p bitrates correspond roughly to the reference bitrates. Reason for the deviation for 720p is that when the reference bitrate is used, the resulting quality for 720p is very similar to the quality for 1080p, which makes transcoding from 1080p to 720p very inefficient, i.e. the loss due to encoding of an already compressed video is very high. That problem would equally apply for both guided transcoding and full transcoding. Therefore a lower target quality was selected.

Table 8: Average bitrates [kbps] for normal x264 encoding.

	1080p	720p	540p	360p
Kimono24Hz	7218	2938	1880	328
ParkScene24Hz	6475	2557	1521	336
Cactus50Hz	6545	2785	1738	316
BasketBDrive50Hz	6227	3210	1778	350
BQTerrace60Hz	7404	2086	1277	317
Average	**6774**	**2715**	**1639**	**329**
Fabrix profile	**7000**	**4800**	**1800**	**300**

The performance for guided transcoding is emulated by disabling transform coefficient coding in x264 (using a software modification), and coding 720p, 540p and 360p based on the decoded 1080p video. For implementation simplicity, the reference used for distortion calculation in the encoder is the decoded 1080p video as well, not the uncompressed video as in the case of the HEVC experiments (the impact on the result is expected to be small, i.e. very few percent).

Table 9 shows the average bitrate reduction when only motion information is coded for each of the lower resolutions. It also shows the total bitrate reduction compared to the baseline scenario, considering the bitrates for 1080p video as well as the motion information for 720p, 540p and 360p. It also shows the maximum theoretical gain which corresponds to the case of full transcoding.

Compared to the results for HEVC, the bitrate reduction is similar for the cases of 720p and 540p. The bitrate reduction for 360p is lower than in the case of HEVC. The total bitrate reduction is lower which is primarily due to the fact that fewer representations are used in the H.264 experiments (4 instead of 7).

Formatted: Heading 2, Don't adjust space between Latin and Asian text

Formatted: Body Text

Formatted: Caption, Indent: Left: 4,5 cm, Space Before: 18 pt, Keep with next, Keep lines together

Formatted: Keep with next

Formatted Table

Formatted: Centered, Keep with next

Formatted: Keep with next

Formatted: Centered, Keep with next

Formatted: Keep with next

Formatted: Centered, Keep with next

Formatted: Keep with next

Formatted: Centered, Keep with next

Formatted: Keep with next

Formatted: Centered, Keep with next

Formatted: Keep with next

Formatted: Centered, Keep with next

Formatted: Font: Bold

Formatted: Keep with next

Formatted: Centered, Keep with next

Formatted: Font: Bold

Formatted: Centered

Formatted: Body Text

Table 9: Average total bitrate reduction for "motion information only" versus simulcast H.264 encoding.

	720p	540p	360p	Total	Max (theoretical)
Kimono	54%	53%	34%	21,79%	41,40%
ParkScene	62%	61%	46%	24,35%	40,29%
Cactus	49%	46%	26%	19,64%	42,30%
BasketBDrive	38%	34%	19%	16,40%	45,96%
BQTerrace	39%	41%	32%	12,95%	33,01%
Average	**48%**	**47%**	**31%**	**19,03%**	**40,59%**

Table 10 depicts the average bitrate cost after transcoding, compared to normal x264 encoding. Numbers are obtained by determining bitrates after decoding, and relating them to bitrates necessary to achieve the same PSNR using normal x264 encoding, matching PSNR values in the same way as described in Section 2.2. PSNR values are measured with respect to the uncompressed videos.

The cost is caused by two reasons, (1) the encoding is based on an already compressed video, (2) the transcoder does not use Trellis quantization whereas the baseline encoder does.

The results are roughly similar to the results obtained for HEVC, although the cost is somewhat higher in case of 720p, which is partly due to the fact that the distortion calculation used for motion encoding uses the compressed video as reference (as mentioned earlier).

Table 10: Average bitrate cost of transcoded representations versus H.264.

	720p	540p	360p
Kimono	22,58%	10,84%	0,64%
ParkScene	17,08%	9,61%	1,88%
Cactus	13,99%	6,56%	-1,20%
BasketBDrive	17,99%	5,82%	-0,83%
BQTerrace	21,07%	11,59%	3,10%
Average	**18,54%**	**8,89%**	**0,72%**

Formatted: Caption, Indent: Left: 4,5 cm, Space Before: 18 pt, Keep with next, Keep lines together
Formatted: Keep with next
Formatted Table
Formatted: Font color: Black
Formatted: Centered, Keep with next
Formatted: Keep with next
Formatted: Centered, Keep with next
Formatted: Keep with next
Formatted: Centered, Keep with next
Formatted: Keep with next
Formatted: Centered, Keep with next
Formatted: Keep with next
Formatted: Centered, Keep with next
Formatted: Keep with next
Formatted: Centered, Keep with next
Formatted: Font: Bold
Formatted: Centered
Formatted: Font: Bold, Not Highlight
Formatted: Font: Bold
Formatted: Font: Bold, Not Highlight
Formatted: Font: Bold
Formatted: Body Text
Formatted: Caption, Indent: Left: 4,5 cm, Space Before: 18 pt, Keep with next, Keep lines together
Formatted: Keep with next
Formatted: Keep with next
Formatted: Keep with next
Formatted: Keep with next
Formatted: Keep with next
Formatted: Keep with next
Formatted: Font: Bold

5.4 Computational complexity

The decoding complexity for H.264 is believed to be in the same range as the decoding complexity for HEVC when the video quality is similar. Therefore the complexity for guided transcoding is expected to be similar for H.264 and HEVC, i.e. the numbers in Section 3 should roughly apply for both codecs.

While the decoding complexity is believed to be similar for H.264 and HEVC, the encoding complexity for highest quality encoding is typically assumed to be significantly higher for HEVC than for H.264. In other words, HEVC exhibits a stronger imbalance between encoding and decoding complexity. Since relative to full transcoding, guided transcoding replaces a full encoding operation with a guided encoding operation that has similar complexity as decoding, the complexity gain with guided transcoding over full transcoding is expected to be less when H.264 is used as compared to HEVC.

56 Summary

This report analyses the use of guided transcoding for origin-to-edge distribution of HEVC-encoded ABR video, evaluating bandwidth requirements for the video transmission as well as computational complexity requirements incurred at the edge. Evaluations are performed for a configuration with seven ABR representations, including one 1080p representation, two 720p representations, two 540p representations, and two 360p representations. The results show that compared to a baseline scenario where no edge transcoding is performed, the bandwidth requirements between origin and edge can be reduced by about 37% while introducing transcoding-induced compression efficiency loss of about 9% on average (11.5%/8.9%/6.5% for 720p/540p/360p, applying for transmission of transcoded lower-quality representations). Considering an Intel Core i5-2540M “Sandy Bridge” CPU with 2.6GHz (from 2011) and an SIMD-optimized HEVC implementation, real-time transcoding of 1080p60 video to a single lower resolution representation (720p, 540p, or 360p) is estimated to fully load roughly 1-2 CPU cores. Transcoding 1080p60 to all six lower representations at the same time is estimated to fully load in the order of 5 CPU cores.

The results are specific for the selected set of ABR video representations (i.e. video resolutions and quality levels) and the selected encoding settings, and there are several trade-offs to make between origin-to-edge savings, transcoding-induced efficiency loss, and transcoding complexity.

Additional simulations are presented to verify the results for guided transcoding when using H.264. Experiments settings are based on a commercial ABR encoding profile. Guided transcoding is emulated by using a modified x264 encoder. The results are roughly in line with the results for HEVC. The complexity for guided transcoding is expected to be similar for H.264 and HEVC. The complexity gain with guided transcoding over full transcoding is expected to be less when H.264 is used as compared to HEVC.

Formatted: Heading 2, Don't adjust space between Latin and Asian text

Formatted: Body Text

~~6~~7 References

[1] “HEVC Complexity and Implementation Analysis “, IEEE Transcactions on CSVT, VOL. 22, NO. 12, DECEMBER 2012.

~~7~~8 Annex: RD plots

In the figures below, “HEVC” refers to regular HEVC encoding as in the baseline scenario; “SI”, or “side information”, refers to the motion information transmitted for the lower quality representations in the guided transcoding scenario; “Transcoded” characterizes the resulting bit stream after guided transcoding.

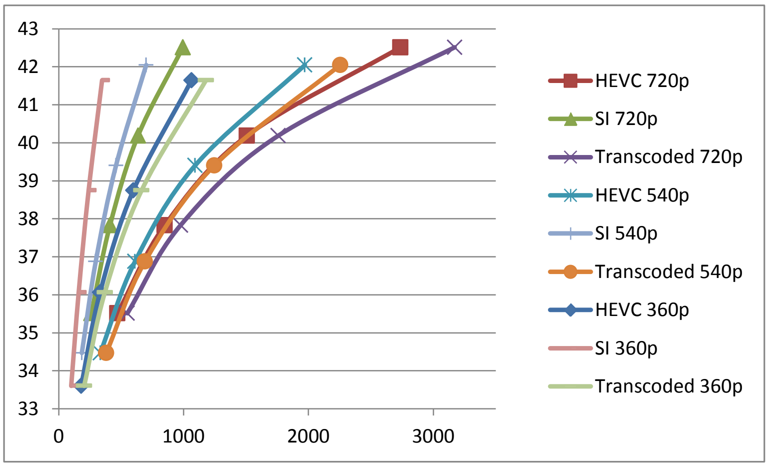

Figure 4: Bitrates and PSNR for HEVC and guided transcoding for Kimono at QP 22, 26, 30, 34.

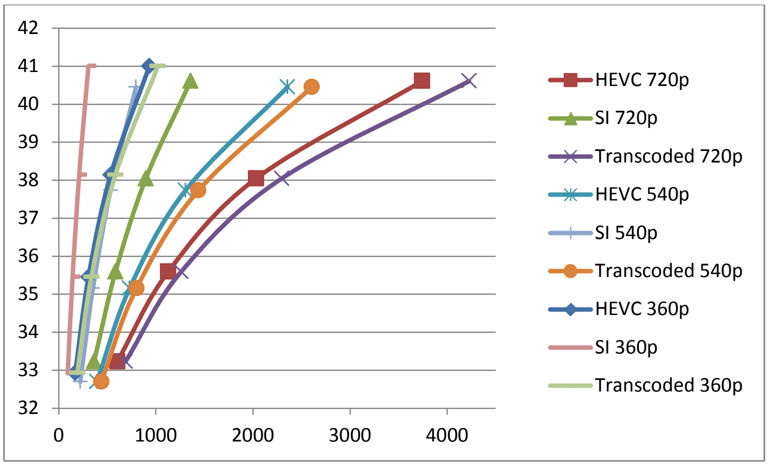

Figure 5: Bitrates and PSNR for HEVC and guided transcoding for ParkScene for QP 22, 26, 30, 34.

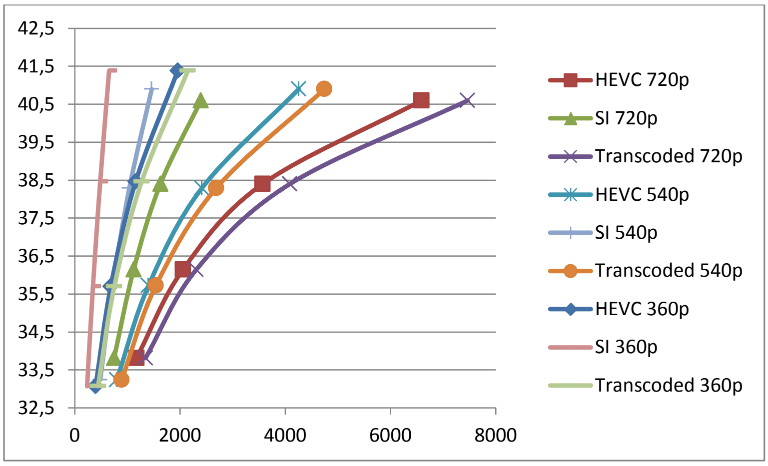

Figure 6 Bitrates and PSNR for HEVC and guided transcoding for Cactus for QP 22, 26, 30, 34.

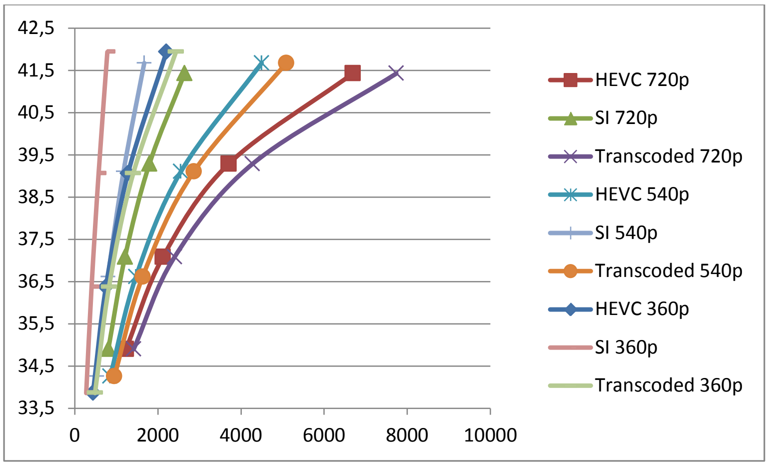

Figure 7: Bitrates and PSNR for HEVC and guided transcoding for BasketballDrive for QP 22, 26, 30, 34.

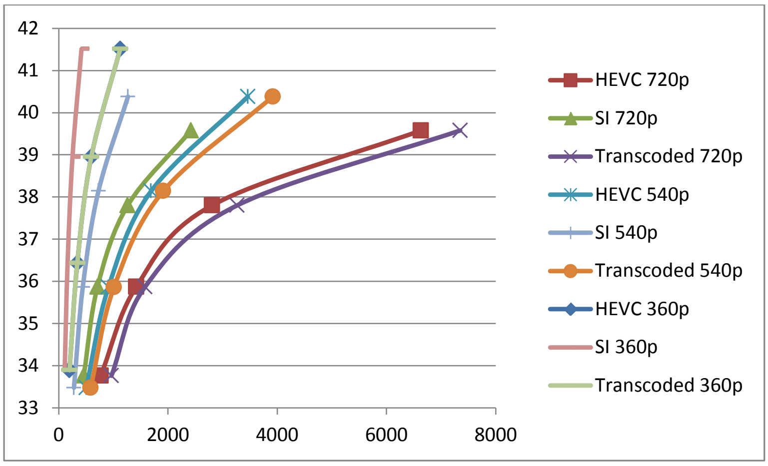

Figure 8: Bitrates and PSNR for HEVC and guided transcoding for BQTerrace for QP 22, 26, 30, 34.