Transcoding or Not? -- A Study of Quantization Configuration for H.264-to-HEVC Transcoding

Jing-Ya Huang[1], Mie-Fen Lin[1], and Pao-chi Chang[1]
[1]Department of Communication Engineering
National Central University
Zhungli, Taiwan
{jingyahuang, mflin, pcchang}@vaplab.ce.ncu.edu.tw

***Abstract*—In the past several decades, many video coding standards with higher coding efficiency were developed. It raises an interesting question: Does transcoding benefit a previous coded video stream? This work uses H.264-HEVC transcoding as the study object. The experimental results show that the quality parameter (QP) plays the key role. With low H.264 QP and high HEVC QP, transcoding can always reduce the bitrate with similar PSNR. The bitrate saving can be over 50%, depending on the texture complexity. On the contrary, high H.264 QP with low HEVC QP can only cost tremendous bitrate with little PSNR improvement.**

Keywords—H.264/AVC; High Efficiency Video Coding (HEVC); transcoding; Quality Parameter (QP).

I. Introduction

It is the trend that higher resolutions and better quality are highly demanded in all kinds of video applications, from high definition video (HD) on mobile devices to ultra high definition (UHD) 4K digital television. In the past several decades, many video coding standards with higher coding efficiency were developed to accommodate the increased huge video data.

High efficiency video coding (HEVC) is currently the most efficient video coding standard [1]. Compared with the widely used H.264/AVC [2], HEVC can achieve the same quality with only 50% bitrate. However, H.264/AVC is still the mainstream and it is not expected to be totally replaced in the near future. Therefore, both standards will co-exist and efficient transcoding from H.264 to HEVC is an important issue [3][4].

An interesting question regarding the transcoding is that "Does transcoding benefit a previous coded video stream in terms of rate-distortion performance?" In other words, for an already H.264 coded stream, is it necessary to transcode it to get an HEVC bitstream? This study tried to answer the question and find efficient quantization parameter (QP) configuration for H.264-to-HEVC transcoding. This study set up an experiment environment. Video sequences with different characteristics were coded with various QPs.

II. TRANSCODING SCENARIOS

Two Scenarios, transcoding with the same frame size and transcoding with upsampling, were considered in this study.

A. Transcoding only

This scenario assumes that the display frame size is the same as the original captured frame size. Starting from an H.264 encoded bit stream, it can simply be decoded by H.264 decoder to show the video. If the RD performance is promising, the sequence can further be encoded by HEVC, then the video can be viewed by decoding the HEVC sequence.

B. Transcoding with upsampling

This scenario assumes that the display frame size is larger than the original captured frame size. For instance, a full HD (1920x1080) video might be displayed on a 4K display. In addition to the above transcoding process, upsampling the HEVC decoded sequences to get a higher resolution is needed.

III. EXPERIMENTAL ENVIRONMENT

Figure 1 shows the complete setup for this study. The 1920x1080 video sequences were first cut 8 pixels in height to fit the macroblock size and resulted in 1920x1072 sequences for simulation. Different parts of the architecture were used in each scenario.

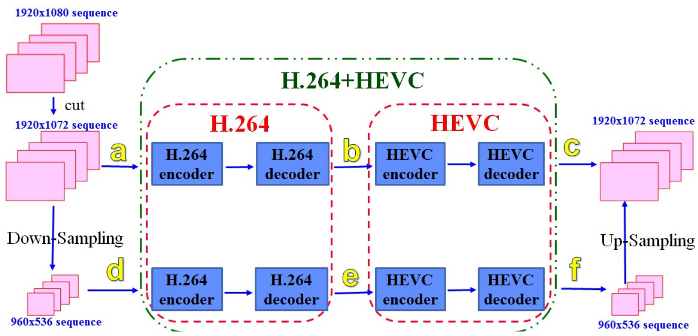

Fig. 1. Complete architecture of H.264-to-HEVC transcoding

For Part 1 transcoding only, the video sequence **a** was H.264 encoded/decoded to get the sequence **b**, which was then processed by HEVC encoder/decoder. The output **c** was compared with the original sequence **a** to calculate the rate-PSNR performance.

Part 2 simulates the performance of transcoding and upsampling, shown as in Fig. 2. The sequence **a** was downsampled to be 960x536 video sequence **d**, which simulated a 2K sequence. The same process as in Part 1 was repeated, i.e., H.264 encoding/decoding and HEVC encoding/decoding. Finally, the output sequence **f** was upsampled to the original resolution **g** to simulate a 4K sequence.

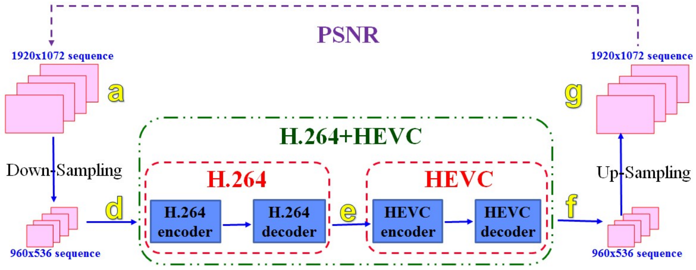

Fig. 2. Downsampling/upsampling transcoding process

IV. EXPERIMENTAL RESULTS AND DISCUSSIONS

The experiments were performed on JM19.0 for H.264 and HM16.9 for HEVC. Experimental conditions, encoder parameters, and test sequences are listed in Table I.

TABLE I. EXPERIMENTAL CONDITIONS

Configuration file (JM19.0)	encoder_JM_Intra_HE, encoder_JM_LB_HE, encoder_JM_LP_HE, encoder_JM_RA_B_HE
Configuration file (HM16.9)	encoder_intra_main, encoder_lowdelay_main, encoder_lowdelay_P_main, encoder_randomaccess_main
Test Sequence	Kimono1, ParkScene, BasketballDrive, BQTerrace, Cactus
Resolution	1920 x 1080 and 960 x 536
QP	15, 19, 23, 27, 31, 35
Frame	20

In Part 1, Fig. 3 shows experimental results, represented by the rate-distortion (RD) performance of BasketballDrive sequence encoded by Low Delay P (LD_P) configuration. The red line shows the R-D performance by H.264 encoding/decoding only with QP 15, 19, 23, 27, 31, and 35. Other lines represent the JM-HM transcoding results. Each line shows the results of a specific QP in JM with various QPs in HM.

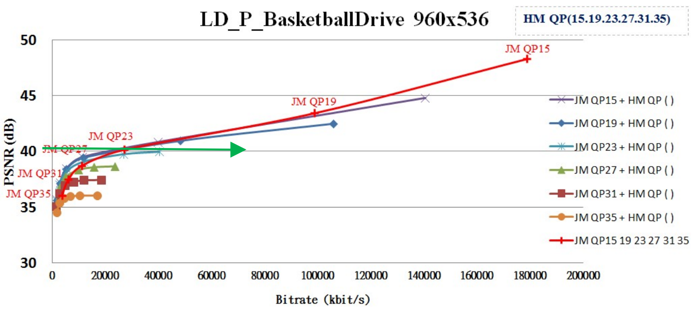

Fig. 3. R-D curves for BasketballDrive sequence under LD_P configuration

All points on the left side of the red line exhibit the benefit of transcoding since lower rates are needed for the same PSNRs. All tested sequences have rooms for RD improvement by transcoding. However, different sequences experience different transcoding gains.

TABLE II shows two examples that HM in the transcoding uses the same QP in HM, e.g., 27 and 31, as the QP used in JM only. The results show that JM-HM transcoding can save over 50% bitrate compared with JM only, and the PSNR reduction is almost not noticeable.

TABLE II. BITRATE SAVINGS OF BASKETBALLDRIVE UNDER LD_P

Low_Delay_P	JM QP( )	JM QP( ) +HM QP ( )		
QP	27	15+27	19+27	23+27
PSNR (db)	38.71	38.4337	38.3823	38.1736
Bitrate (kbit/s)	11189.4	5304.88	5300.54	5231.12
Bitrate Saving (%)		-53%	-53%	-53%
QP	31	15+31	19+31	23+31
PSNR (db)	37.458	37.1592	37.1282	36.9926
Bitrate (kbit/s)	6070.8	2951.16	2950.02	2903.74
Bitrate Saving (%)		-51%	-51%	-52%

In Part 2, transcoding with upsampling is employed to fit larger display size. TABLE III and IV show the results for Kimono and BQTerrace sequences after JM-HM transcoding (represented by PSNR) and after upsampling (represented by up PSNR). Similar to Part I, the PSNR decreases as QP in HM increases. The quality degradation from upsampling is highly affected by the video characteristics. The gap between PSNR and up PSNR is small for low texture complexity sequences, such as Kimono. On the contrary, sequences with high texture complexity, such as BQTerrace, exhibits significant quality loss. Therefore, upsampling for complex sequences is not recommended.

TABLE III. PSNRS AFTER TRANSCODING AND AFTER UPSAMPLING FOR KIMONO UNDER ALL-INTRA CONFIGURATION

QP	15x15	15x19	15x23	15x27	15x31	15x35
up PSNR	41.8319	41.2078	40.2869	38.9051	37.1263	34.9636
PSNR(db)	46.9002	45.446	43.4317	41.0403	38.4812	35.7691
(kbit/s)	20105.4144	11656.9536	7781.1552	5157.4944	3402.816	2113.8048

TABLE IV. PSNRS AFTER TRANSCODING AND AFTER UPSAMPLING FOR BQTERRACE UNDER ALL-INTRA CONFIGURATION

QP	15x15	15x19	15x23	15x27	15x31	15x35
up PSNR	28.2076	28.1514	28.0164	27.7604	27.299	26.5105
PSNR(db)	46.1024	43.9894	41.2939	38.298	35.1894	32.1232
(kbit/s)	79439.832	57582.504	41450.016	29081.688	19678.92	12408.72

V. CONCLUSION

With similar QP, transcoding can always reduce the bitrate. With low H.264 QP and high HEVC QP, transcoding can always reduce the bitrate with similar PSNR. The bitrate saving can be over 50%, depending on the texture complexity. On the contrary, high H.264 QP with low HEVC QP can only cost tremendous bitrate with little PSNR improvement.

REFERENCES

[1] G. J. Sullivan, J.-R. Ohm, W.-J. Han, and T. Wiegand, "Overview of the High Efficiency Video Coding (HEVC) Standard", IEEE Trans. Circuits and Systems for Video Technology, Vol. 22, No. 12, pp. 1649-1668, Dec. 2012.

[2] Advanced Video Coding, ISO/IEC 14496-10, ITU-T Rec. H.264, Version 13, Mar. 2011.

[3] M. Budagavi, A. Fuldseth, G. Bjøntegaard, V. Sze, M. Sadafale, “Core Transform Design in the High Efficiency Video Coding (HEVC) Standard,” IEEE Journal of Selected Topics in Signal Processing, December 2013.

[4] J. T. Fang, Z. Y. Chen, T. L. Liao, and P. C. Chang, “ A Fast PU Mode Decision Algorithm for H.264/AVC to HEVC Transcoding,” in Proc. Fourth International conference on Computer Science & Information Technology (CCSIT), Sydney, Australia, Feb. 2014, vol. 4, no. 2, pp. 215-225.