

H.264/AVC Video Compression on Smartphones

M P Sharabayko, N G Markov

Tomsk Polytechnic University, 30, Lenina ave., Tomsk, 634050, Russia

sme_box@tpu.ru

Abstract. In this paper, we studied the usage of H.264/AVC video compression tools by the flagship smartphones. The results show that only a subset of tools is used, meaning that there is still a potential to achieve higher compression efficiency within the H.264/AVC standard, but the most advanced smartphones are already reaching the compression efficiency limit of H.264/AVC.

1. Introduction

Smartphones are widely spread nowadays. They all support not only video playback, but also video recording. Recorded videos are stored on the phone itself, on PC, etc., or they can be streamed to the network. In these applications, video compression efficiency plays a very important role.

Almost all smartphones today perform video compression within current industrial video compression standard H.264/AVC [1], first published in 2003. Adherence to the standard provides a possibility to recognize and playback video on any device, compliant to the H.264/AVC standard.

The standard itself offers a variety of coding tools that can be used by the encoder. It is worth mentioning that the encoder is to choose the tools it will be using for compression. Most often the reduced set of coding tools is used on low performance devices like mobile phones to decrease video coding complexity and fit within target compression speed. Even when the video compression standard is supported, it does not mean that all the available coding tools are used, and hence the provided video compression efficiency may be far from maximum achievable rates.

Meanwhile, a newer compression standard H.265/HEVC [2] was ratified by the ITU and ISO in January 2013. It provides a larger set of coding tools [3,4], making it possible to improve compression rates up to two times compared to H.264/AVC [5]. An increased set of coding tools also means higher computational complexity, mainly in video encoding, but also in playback. The usage of H.265/HEVC on mobile platforms makes sense only when there are no options to increase compression efficiency within the H.264/AVC standard. In other words, until most of H.264/AVC coding tools are used for real-time video compression on mobile devices, there is no benefit from H.265/HEVC and thus compression rates will stay within the H.264/AVC limitations.

In this paper, we study current state of video compression on flagship smartphones provided “out of the box” with the default recording software to assess how close they are to the compression ratio limit within the H.264/AVC compression standard. In Section 2, we describe key coding tools within the H.264/AVC standard that have a major impact on compression efficiency. Section 3 provides experimental results and elaboration on a coding tool used by various smartphones. Finally, our conclusions are made in Section 4.



2. Overview of AVC coding tools

2.1. General compression data-flow

H.264/AVC is a hybrid block-based video compression standard. Its compression data-flow is illustrated in Figure 1. An input video frame is initially partitioned into equal-sized blocks of 16×16 luma samples called macroblocks. A macroblock is partitioned into smaller blocks to perform prediction. There are two basic types of prediction: intra and inter. Intra-prediction works within the current video frame and is based upon the compressed and decoded data available for the block being predicted. Inter-prediction is used for motion compensation: prediction uses a similar region on a previously coded frame. The aim of the prediction process is to reduce data redundancy and, therefore, to reduce excessive information in coded bitstream [6].

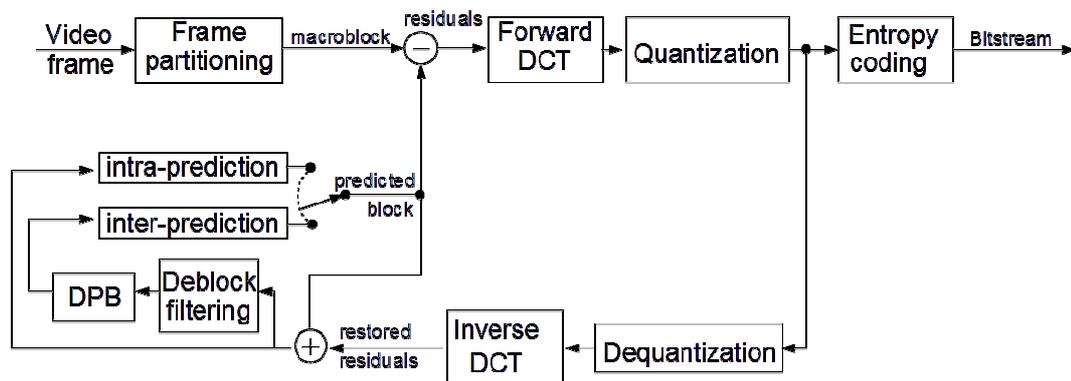


Figure 1. The data-flow of the hybrid block-based video encoder

Once the prediction is done, it is subtracted from the original data to get residuals. Residuals are subjected to Forward Discrete-Cosine Transform (DCT). DCT translates spatial residual information into a frequency domain. Thus the remaining spatial redundancy of this information is partly reduced. Quantization is applied to the transformed matrix to lose inconsiderable information. The remaining data and the steps applied are subjected to entropy coding, which forms a compressed bit-stream [6].

For inter- and intra-prediction purposes, the compressed data should be reconstructed in the encoder. The only data loss takes place after integer DCT and quantization. Dequantization and inverse DCT are performed to restore residuals. The restored residuals and predicted values are summed up to get reconstructed pixel values. These reconstructed values are used for intra-prediction within the current video frame. An additional frame post-processing stage called deblock filtering is optionally applied to eliminate image blockiness. The final restored and post-processed video frame is stored in Decoded Picture Buffer (DPB) for inter prediction of further frames.

2.2. Entropy coding

The video coding process produces a number of values that must be encoded to form a compressed bitstream. These values include: transformed and quantized residuals, information about the structure of the compressed data and the compression tools used during encoding, other supplemental information. These values are presented as a sequence of binary codes using variable length coding and/or arithmetic coding [7].

The H.264/AVC video compression standard provides two entropy coders: Context-Adaptive Variable-Length Coding (CAVLC) and Context-Adaptive Binary Arithmetic Coding (CABAC). The type of the entropy coder to be used is fully decided by the video encoder. CABAC provides about 10% higher compression rates at the same level of image distortion. The drawback is that CABAC is more computationally expensive in encoding as well as in decoding. Mobile video compression systems often use CAVLC instead to reduce computational load.

2.3. Deblock filtering

H.264/AVC provides an optional feature of image post-processing called deblock filtering. Deblock filtering works on borders of block partitions to reduce blockiness artefacts that appear after coarse quantization. Filtering improves not only the visual quality, but also compression efficiency of P and B frames, because blockiness decreases the efficiency of inter-frame motion compensation. At the same time, deblock filtering involves additional computing operations in the encoder, and in the decoder. Low performance systems sometimes do not use this feature to save computing resources.

2.4. Spatial intra prediction modes

There are ten spatial intra prediction modes available in H.264/AVC. Any luma block of size 16×16 pixels can only be predicted by one of four modes, while prediction of 4×4 and 8×8 luma blocks can be performed with any of 9 prediction modes (see Table 1). A video encoder can use any available intra prediction mode. Low performance compression systems tend to reduce intra prediction modes set to choose from thus decreasing computational complexity at the cost of less efficient compression.

Table 1. Available spatial intra prediction modes depending on the intra prediction block size

Intra prediction mode	Block size		
	4×4	8×8	16×16
Plane	no	No	yes
Vertical	yes	Yes	yes
Horizontal	yes	Yes	yes
DC	yes	Yes	yes
Diagonal Down Left	yes	Yes	no
Diagonal Down Right	yes	Yes	no
Vertical Right	yes	Yes	no
Horizontal Down	yes	Yes	no
Vertical Left	yes	Yes	no
Horizontal Up	yes	Yes	no

2.5. Intra partitioning

H.264/AVC describes three partitions for a luma component of an intra coded 16×16 macroblock: sixteen sub-blocks of 4×4 pixels; four sub-blocks of 8×8 pixels; only one block of 16×16 pixels. The video encoder has to choose a proper partitioning for each macroblock. Smaller blocks provide closer prediction, but more supplemental bits have to be coded. The goal is to have a prediction close enough, but with less bits involved. Also smaller intra partitioning requires smaller transform blocks. For example, 4×4 partitioning involves 4×4 transform blocks, which also leads to more residual bits.

2.6. Temporal inter prediction

Temporal inter prediction, also called motion compensation, provides a huge opportunity to decrease the amount of information for entropy coding. Static regions and moving objects can be predicted close enough to significantly reduce the bit size of a coded block.

There are two types of inter predicted frames in H.264/AVC: P-frame and B-frame. P-frames (predicted) are allowed to have blocks predicted from one of the previously coded frames. B-frames (bidirectionally predicted) are allowed to have blocks predicted from two previously coded frames. Prediction from two motion regions is a pixel-by-pixel weighted average of two predictions. B-frames are used to get additional rate savings. On the other hand, additional memory is required to store at least two previously coded frames, and, furthermore, additional compression delay is introduced.

2.7. Inter partitioning

H.264/AVC supports inter prediction block sizes ranging from 16×16 to 4×4 luma samples. The luma component of each macroblock (16×16 samples) can have one of four possible partitionings (16×16,

16×8, 8×16 or 8×8), illustrated in Figure 2. If 8×8 macroblock partitioning is chosen, each of four 8×8 macroblock partitions can have one of four possible sub-partitions (8×8, 8×4, 4×8 or 4×4), illustrated in Figure 3. These partitions and sub-partitions give rise to a large number of possible combinations within each macroblock [8].

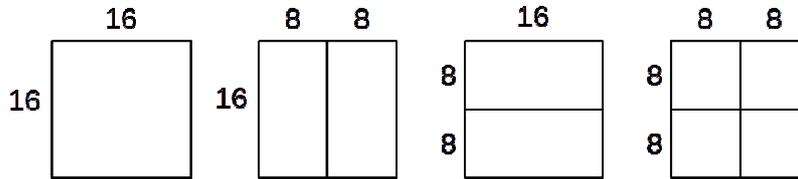


Figure 2. Macroblock partitions: 16×16, 8×16, 16×8, 8×8

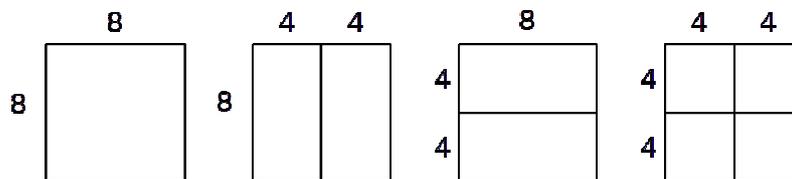


Figure 3. Macroblock sub-partitions: 8×8, 4×8, 8×4, 4×4

Block partitioning involves some decision algorithms that usually become more computationally expensive with the increase of inter partitions involved. Each partition in an inter-coded macroblock is predicted from an area of the same size in a reference picture. The offset between the two areas (the motion vector) has ¼-pixel resolution (for the luma component). The luma and chroma samples at sub-pixel positions do not exist in the reference picture and have to be interpolated from nearby image samples. Sub-pixel motion compensation can provide significantly better compression performance than integer-pixel compensation, at the expense of increased complexity [7].

A search for the best motion vector is called motion estimation. It is a job of the encoder to come up with a motion estimation algorithm. And, it is the encoder that decides a search region and whether to use ½ and ¼-pixel interpolation or not, depending on the computational resources available.

2.8. Transform sizes

The encoder subtracts predicted pixels of a macroblock from its actual pixels to form a residual. A block of residual samples is transformed using a 4×4 or 8×8 integer transform. The transform outputs a set of coefficients, each of which is a weighting value for a standard basis pattern [7].

2.9. Adaptive Quantization

The output of the transform is quantized, i.e. each coefficient is divided by an integer value. Quantization reduces the precision of the transform coefficients according to a quantization parameter (QP) [4]. Most of the time all macroblocks within a video frame have the same QP, but H.264/AVC provides an opportunity to select QP on a macroblock basis. If a frame consists of macroblocks with different QP values, then it means that an adaptive quantization technique is applied. This technique makes it possible to save more details where they are needed, and loose more information where it is less relevant. However, it involves QP decision logic that complicates encoder, and often adaptive quantization is not used.

3. Test conditions and results

Experiments are held upon video samples, found on the GSMarena website [9], that were taken directly from smartphones without any processing. These samples are used to analyse H.264/AVC coding tools employed in video compression. Two widely used video resolutions and frame rates are studied: 3840×2160 pixels (4K Ultra HD) and 1920×1080 pixels (Full HD) both at 30 frames per second.

The flagship smartphones that support 4K Ultra HD video recording were taken for experiments. These smartphones are to provide the most advanced video compression, as they have several of the most advanced mobile CPUs, GPUs and chipsets.

Table 2. Usage of general coding tools

	CABAC		AQ		ME		Bitrate	
	2560	1080	2560	1080	2560	1080	2560	1080
Apple iPhone 6s	✓	✓	✓	✓	190x94	190x94	50	17
Samsung Galaxy S7	✓	✓	✓	✓	128x64	128x64	50	17
Microsoft Lumia 950XL	✓	✓	✗	✗	94x46	94x46	50	18
Sony Xperia Z5 Premium	✓	✓	✗	✗	94x30	94x46	55	20
LG G5	✓	✓	✗	✗	157x29	160x64	48	17
Huawei Nexus 6P	✗	✗	✗	✗	94x46	94x46	42	17
HTC 10	✓	✗	✗	✗	157x29	160x64	55	20
Motorola Moto Z Force	✓	✓	✗	✗	157x29	160x64	51	17
Xiaomi Mi 5	✗	✗	✗	✗	157x29	160x64	42	15

Table 2 shows the usage of general coding tools: an entropy encoder (CABAC or CAVLC), adaptive quantization (AQ), motion estimation (ME) region in integer pixel samples, and a video bitrate. All the studied smartphones apply deblock filtering, so it is not included in the table. Also B-frames are not used by all of the studied smartphones. Almost similar bitrates at the target resolutions and frame rates are provided. Generally speaking, the higher the bitrate is, the less compression artifacts are introduced to the recorder video. At the same time, the higher bitrate provides higher computational load on the entropy coder. From the results in Table 3 we may suggest that Xiaomi Mi 5 uses a lower target bitrate to decrease computational load and provide the target frame rate in real time.

Most of the devices use CABAC as an entropy coder, except for Huawei Nexus 6P and Xiaomi Mi 5. At the same time HTC 10 uses CABAC only for 4K Ultra HD resolution. Adaptive quantization is applied only by Apple iPhone 6s and Samsung Galaxy S7.

Motion estimation regions, shown in Table 2, are taken from the maximum motion vector used throughout the stream (in integer pixels). The results may not be precise, but we believe they are very close to actual borders. The first thing to be noted is the rectangular form of the region: all the devices seem to optimize their motion search to save computing resources, suggesting that the most of the motion is to happen in a horizontal direction, parallel to the ground, which makes sense in real life recordings. The largest ME region is used by Apple iPhone 6s, the second largest - is in LG G5.

Table 3. Usage of video coding tools for 3840x2160 resolution at 30 frames per second

Smartphone	Intra blocks			TB		Inter blocks						
	4x4	8x8	16x16	4x4	8x8	4x4	4x8	8x4	8x8	8x16	16x8	16x16
Apple iPhone 6s	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Samsung Galaxy S7	✓	✓	✓	✓	✓	✗	✗	✗	✓	✓	✓	✓
Microsoft Lumia 950XL	✗	✗	✓	✓	✗	✗	✗	✗	✓	✓	✓	✓
Sony Xperia Z5 Premium	✗	✗	✓	✓	✓	✗	✗	✗	✓	✓	✓	✓
LG G5	✗	✓	✓	✓	✓	✗	✗	✗	✓	✓	✓	✓
Huawei Nexus 6P	✗	✗	✓	✓	✗	✗	✗	✗	✓	✓	✓	✓
HTC 10	✗	✓	✓	✓	✓	✗	✗	✗	✓	✓	✓	✓
Motorola Moto Z Force	✗	✓	✓	✓	✓	✗	✗	✗	✓	✓	✓	✓
Xiaomi Mi 5	✗	✗	✓	✓	✗	✗	✗	✗	✓	✓	✓	✓

Table 3 presents the usage of various block sizes for intra and inter prediction and for transform coding by mobile phones for recording 30 video frames per second at a resolution of 3840×2160 pixels. The results show that only Apple iPhone 6s uses all the block sizes provided by the AVC standard. Samsung Galaxy S7 also uses all the block sizes available except for 4×4, 4×8 and 8×4 inter prediction. All the rest studied mobile phones do not use these inter prediction block sizes as well, but additionally they omit 4×4 intra prediction blocks. Some mobile phones also do not use 8×8 intra prediction and 8×8 transform blocks.

Table 4 presents the usage of various block sizes for recording 30 frames per second at 1920×1080 pixels. Microsoft Lumia 950XL, Sony Xperia Z5 Premium and Huawei Nexus 6P use 4×4 and 16×16 intra prediction blocks only, but 4×4 intra prediction modes Vertical Right and Vertical Left are omitted. Therefore only 8 out of 10 intra prediction modes are used. HTC10 and Xiaomi Mi 5 also omit 8×8 intra prediction, but use all the available 10 intra prediction modes. It is interesting to emphasize that HTC 10 employs 8×8 intra prediction for 3840×2160, and 8×8 intra prediction for the 1920×1080 video resolution. Also, HTC 10 utilizes CABAC only for 4K video, while using CAVLC for Full HD. It is worth mentioning that LG G5 uses 8×8 transform blocks only for intra macroblocks.

Table 4. The usage of video coding tools for 1920×1080 resolution at 30 frames per second

Smartphone	Intra blocks			TB		Inter blocks						
	4×4	8×8	16×16	4×4	8×8	4×4	4×8	8×4	8×8	8×16	16×8	16×16
Apple iPhone 6s	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Samsung Galaxy S7	✓	✓	✓	✓	✓	✗	✗	✗	✓	✓	✓	✓
Microsoft Lumia 950XL	✓	✗	✓	✓	✗	✗	✗	✗	✓	✓	✓	✓
Sony Xperia Z5 Premium	✓	✗	✓	✓	✓	✗	✗	✗	✓	✓	✓	✓
LG G5	✓	✓	✓	✓	✓	✗	✗	✗	✓	✓	✓	✓
Huawei Nexus 6P	✓	✗	✓	✓	✗	✗	✗	✗	✓	✓	✓	✓
HTC 10	✓	✗	✓	✓	✗	✗	✗	✗	✓	✓	✓	✓
Motorola Moto Z Force	✓	✓	✓	✓	✓	✗	✗	✗	✓	✓	✓	✓
Xiaomi Mi 5	✓	✗	✓	✓	✗	✗	✗	✗	✓	✓	✓	✓

4. Conclusion

In this paper we studied the usage of the H.264/AVC video compression tool in video recording with flagship smartphones. The results show that the most advanced smartphones use almost all the available coding tools, except for B-frames and only one reference frame in DPB. The rest of the studied smartphones do not apply a lot of coding tools. We may conclude that it will take some time for the phones to reach the compression efficiency limit within AVC, therefore adoption of H.265/HEVC is already possible, at least below 4K UltraHD resolutions [10].

5. References

- [1] Recommendation H.264: Advanced video coding for generic audiovisual services 2014 *ITU-T*
- [2] Recommendation H.265: High Efficiency Video coding 2015 *ITU-T*
- [3] Sullivan G, Ohm J, Woo-Jin H and Wiegand T 2012 Overview of the High Efficiency Video Coding (HEVC) *IEEE Transactions on Standard Circuits and Systems for Video Technology* **22** 1649-1668
- [4] Bordes P, Clare G, Henry F and Raulet M 2010 An overview of the emerging HEVC standard. *International Symposium on Signal, Image, Video and Communications* (Gold Coast, Australia)
- [5] Sharabayko M and Markov N. 2016 Contemporary video compression standards: H.265/HEVC, VP9, VP10, Daala *Proc. Int. Siberian Conf. on Control and Communications (SIBCON) (Moscow)* **1** 1–4
- [6] Sharabayko M, Ponomarev O and Chernyak R Intra Compression Efficiency in VP9 and HEVC 2013 *Applied Mathematical Sciences* **7** (137) 6803-6824

- [7] Richardson I 2010 *The H.264 Advanced Video Compression Standard, 2nd Edition*
- [8] Richardson I 2011 *White Paper: H.264/ AVC Inter Prediction Vcodex*
- [9] <http://www.gsmarena.com>
- [10] Bross B, George V, Alvarez-Mesa M, Mayer T, Chi C, Brandenburg J, Schierl T, Marpe D, Juurlink B 2013 HEVC performance and complexity for 4K video *IEEE Third International Conference on Consumer Electronics (Berlin)* pp 44-47