Introduction to H.264 Video Compression Standard

Frank Wang | Software Engineer

ABSTRACT—In the IP camera industry, H.264 is the most popular video compression standard that provides the format for recording digital video. The H.264 standard was first published in 2003, with several revisions and updates since then. It has achieved a significant improvement in rate distortion efficiency relative to existing standards. It has made significant progress in compression efficiency and uses less capacity when data is stored or transmitted.

I - Introduction

In a typical application of H.264 such as video surveillance, video from a camera is encoded using H.264 to produce an H.264 bitstream. It is sent across a network to a decoder which reconstructs a version of the source video.

This standard exploits both temporal correlation and spatial correlation to remove the pixel redundancy in a video sequence. Furthermore, the high correlation between each syntax element is also used to predict and then represent the target syntax element. Thus, the interdependence between each syntax element is quite huge.

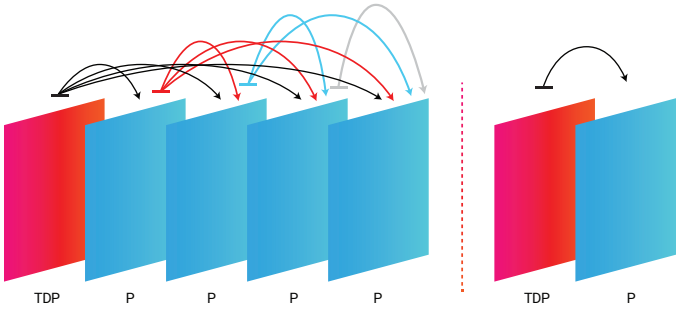
II - The H.264 Bitstream Structure

A coded picture consists of one or more slices that are made up of a series of coded macroblocks. An overview of the H.264 bitstream structure is in the following section.

A. Group of Pictures

A group of pictures (GOP) comprises a successive set of pictures which starts with a key picture [1]. In a closed-GOP coding structure, the key picture may be an instantaneous decoding refresh (IDR) picture that consists of one or more IDR slices or a special type of intra-coded slice. If an IDR picture is received at the decoder, all stored pictures that are in the decoded picture buffer (DPB) are marked "unused" immediately. Therefore, pictures preceding the IDR picture in coding order are no longer available for prediction.

In a hierarchical-P coding structure (i.e., coding order is equal to output order), P pictures are predicted from the pictures that are already encoded. This structure is designed for scenarios which require low-delay operation. A hierarchical-P coding structure example is presented in Fig.1.



B. Network Abstraction Layer Unit

An H.264 coded video sequence consists of a series of network abstraction layer units (NALUs). Each of them may include parameter sets, supplemental information, an entire coded picture, or parts of an coded picture [2]. A VCL NALU consists of a one-byte NALU header that is used to define the information within the NALU payload. There are three fields in the NALU header:

1. a 1-bit forbidden_zero_bit (F): indicates the data loss which can be used to trigger error concealment at the decoder 2. a 2-bit nal_ref_idc (NRI): signal the importance of NALU

3. a 5-bit nal_unit_type (NUT): signals the type of encapsulated byte sequence packets (EBSP) in the NALU

The most common NALUs are listed in Table I. The decoder can detect the NALU type in the NALU header to perform the different processes.

Nut	Nalu content	NRI	NALU class
1	non-IDR picture	2	VCL
5	IDR picture	3	VCL
6	Supplermental Enhancement Information	0	non-VCL
7	Sequence Parameter Set	3	non-VCL
8	Picture Parameter Set	3	non-VCL

C. Macroblock Layer

The composition of a video coding layer (VCL) NALU is presented in Fig. 2. The macroblock is the basic unit for the coding process in a picture. The region size is 16×16 pixels and contains one 16×16 luminance sample and two 8×8 chrominance samples in 4:2:0 YUV format.

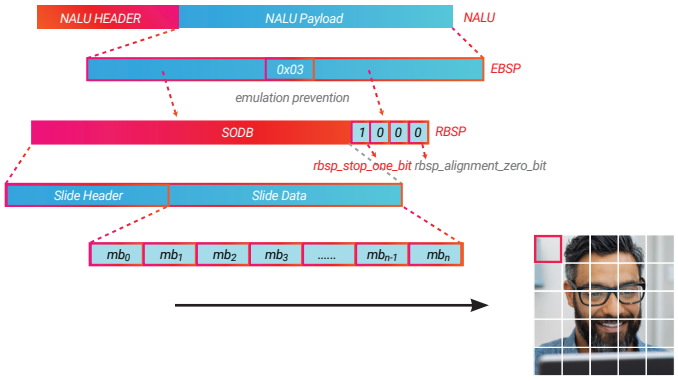


Fig.2. The composition of a VCL NALU, only one slice in a picture.

The macroblock information is presented by a lot of syntax elements that form the string of data bits (SODB). Because a SODB may or may not be byte-aligned, it is necessary to transfer the SODB into the byte-aligned raw byte sequence payload (RBSP) for the specification requirements. In order to prevent zero_byte (i.e., 0x00) and start_code_prefix_one_3bytes (i.e., 0x000001) from occurring in the NALU payload, an emulation prevention (i.e., 0x03) is inserted into the bitstream when 0x0000 is happening.

Figure 3 illustrates a general H.264/AVC encoder architecture; the related encoding process is described as follows: a raw picture is divided into many non-overlapping macroblocks, and then a prediction macroblock is generated by intra or inter prediction and subtracted from the target macroblock to form a residual macroblock. Then, the residual macroblock is transformed and quantized according to the selected quantization parameter (QP). Finally, the residual macroblock is coded into the bitstream by entropy coding. There are two entropy coding modes allowed in H.264/AVC; one is context-based adaptive variable length coding (CAVLC), and the other is context-based adaptive binary arithmetic coding (CABAC). Only the CAVLC entropy coding tool is supported in the baseline profile.

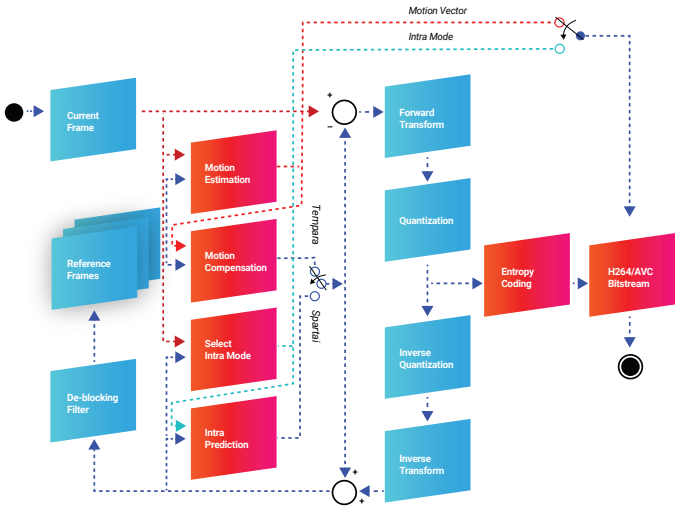


Fig.3. The H.264/AVC encoder architecture

1. Intra Prediction

An intra macroblock (I-macroblock) is coded using spatial correlation. The slice that has the same slice group number can act as the predicted reference for intra prediction in the I picture.

Figure 4 exhibits a set of intra 4×4 prediction modes that are available in H.264, including a DC and 8 angular predictions to make up different combinations of prediction. These prediction modes are suitable for the complex region.

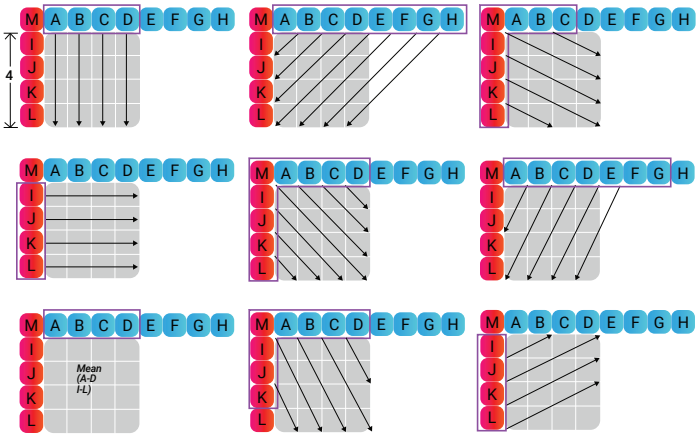


Fig.4. Intra 4×4 prediction modes.

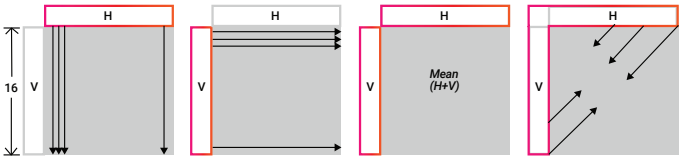


Fig.5. Intra 16×16 prediction modes.

Figure 5 exhibits a set of 4 intra 16×16 prediction modes. The pixels that are highlighted in red line are used to form the prediction for the current block. A bigger prediction block size tends to give less precision, but fewer bits are required to code the prediction mode.

Figure 6 shows an example for intra 4×4 macroblock from the Elecard StreamEye software. The decoded macroblock is the combination of the predicted macroblock and the residual macroblock. The highlighted red block is coded by the intra 4×4 vertical prediction mode because the original texture has the vertical texture; the encoder attempts to select the best mode during rate distortion optimized (RDO) mode selection. The best coding mode of a macroblock is based on the trade-off between the bitrate and distortion cost that is controlled by the QP. A smaller QP tends to select the mode that gives the least distortion, allowing a higher bitrate.

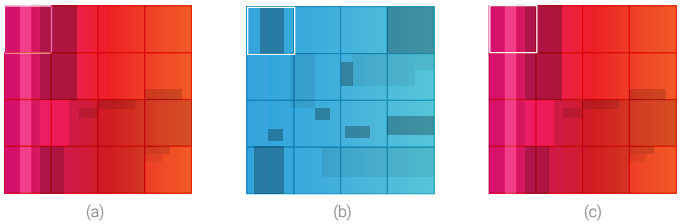


Fig.6. An example for intra 4×4 macroblock. (a) Predicted macroblock. (b) Residual macroblock. (c) Reconstructed macroblock.

2. Inter Prediction

An inter macroblock (P-macroblock) is coded using the temporal correlation. The prediction block is formed from previously coded pictures stored in the DPB. There are many partition sizes for selection in H.264/AVC. The ultimate goal is to find a trade-off between the total bits that are used for coding the motion vector and the residual data and the distortion cost. As can be seen in Figure 7, the offset between the co-located block and the best match block (i.e., the car tire) in the reference frame is the so-called motion vector [3].

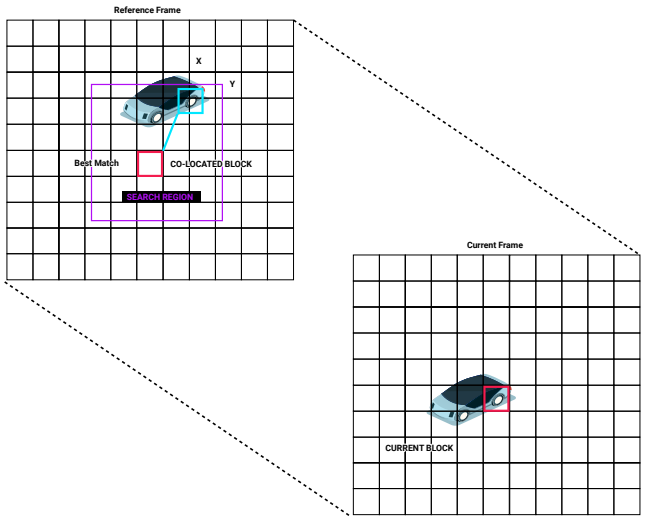


Fig.7. Finding a suitable prediction block.

3. Residual Data

The output of the prediction process is a residual block, created by subtracting the prediction block from the current block, and a set of parameters signaling the intra prediction type or describing how the motion block was estimated. Figure 8 depicts the residual blocks in an intra 4×4 macroblock for YUV 4:2:0 sampling; the number labeled in the block is the transmission order.

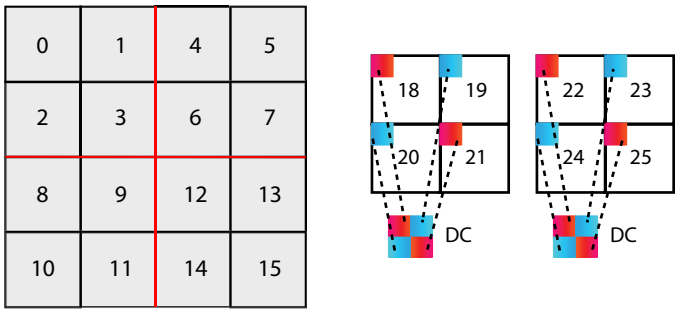


Fig.8. The residual blocks in an intra 4×4 macroblock, 4:2:0 sampling.

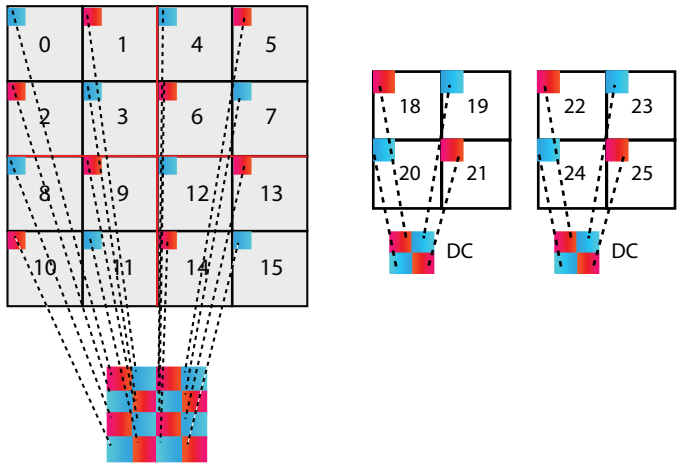


Fig.9. The residual blocks in an intra 16×16 macroblock, 4:2:0 sampling.

The residual data have the most important influence on the bitrate of a bitstream. One way of controlling bitrate is simply to try and enforce a constant number of bits per coded frame, by measuring the output bitrate and feeding it back to control QP. Increasing QP reduces coded bitrate and decreasing QP increases coded bitrate.

D.Profiles

The H.264 standard specifies many syntax and decoding algorithms, which cover a wide range of potential video scenarios. A profile places limits on the algorithmic capabilities required

of an H.264 decoder. Each Profile is intended to be useful to a class of applications. Hence, a decoder conforming to the main profile of H.264 only needs to support the tools contained within the main profile. For example, the baseline profile may be useful for low-delay or conversational applications such as video conferencing, with relatively low computational requirements. The main profile may be suitable for basic television applications such as standard definition TV services. The high profiles add tools to the main profile which can improve compression efficiency especially for higher spatial resolution services such as high definition TV.

Table II lists the coding tools supported by the baseline, extended, main and high profiles. We can choose a suitable profile for the special scenario.

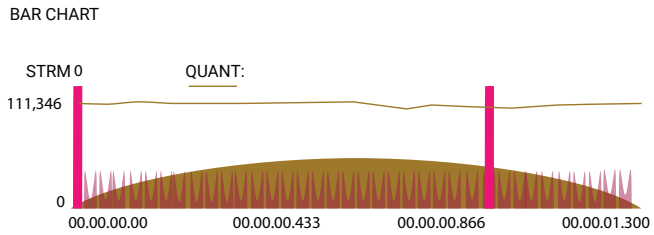
Coding tools that support in particular profiles ↵

Feature	Baseline	Extended	Main	High
Flexible macrobook ordering	√ ↵	√ ↵	↵	↵
Arbitrary slice ordering	√ ↵	√ ↵	↵	↵
Data participating	↵	√ ↵	↵	↵
B slices	↵	√ ↵	√ ↵	√ ↵
CABAC entropy coding	↵	↵	√ ↵	√ ↵
8x8 vs. 4x3 transform adaptivity	↵	↵	↵	√ ↵

Table II

III - Analysis Of Practical Stream

As can be seen in Figure 10, we capture a practical stream from our IP camera and analyze the stream by the Elecard's StreamEye analyzer software. In this case, a main profile sequence that starts with an IDR slice (red bar) followed by P slices (blue bar). It is obvious that the bitrate of the IDR picture is greater than the bitrate of the P picture. Besides, the entropy coding method is CABAC and the frame rate is 30.



name	value
strean type	AVC/H.204
profile	Main
level	3.1
chroma format	4:20
resolution	1920 x 1080
frame rate	30
coding mode	CABAC

IV - Conclusion

In this paper, we offered an overview of the H.264 video compression standard. It has significantly improved in rate-distortion efficiency relative to existing standards. It describes the significant progress on compression efficiency that uses less capacity when it is stored or transmitted. There are two attractive video compression standards, H.265 and AV1. However, there is still a long way to go to meets the needs of manufacturer and customer.

References

M. Wien, High Efficiency Video Coding: Coding Tools and Specification, Berlin: Springer-Verlag, 2015.

2. I. E. Richardson, The H.264 Advanced Video Compression Standard, 2nd ed., New York: Wiley, 2010.

3. Y. Wang, J. Ostermann, and Y. Q. Zhang, Video Processing and Communications, Prentice Hall, 2001.

EnGenius Technologies | 1580 Scenic Ave. Costa Mesa, CA 92626

Email: partners@engeniustech.com | Website: engeniustech.com

Version: 1.00 03/2020