IOSR Journal of Electronics and Communication Engineering (IOSRJECE)
ISSN : 2278-2834 Volume 1, Issue 5 (May-June 2012), PP 06-10
www.iosrjournals.org

H.264 Deblocking Speed-up Using Hybrid Optimization Technique

Shoaib Kamal
Assistant Professor
Department of Electronics and Communication Engineering,
P.A College of Engineering, Mangalore

Abstract— *The proposed method defines the problem of reducing the complexity of H.264 decoding. Since deblocking accounts for a significant percentage of H.264 decoding time. Observing that branch operations are costly and that in the deblocking process there are events with significantly high probability of occurrence. The idea of Hybrid optimization to speed up the boundary strength derivation and the true-edge detection by taking advantage of the biased statistical distribution is applied. The proposed techniques can reduce the deblocking computation time, while maintaining the bit-exact output.*

Keywords- *Deblocking, H.264 Decoding, Boundary Strength, Optimization*

I. INTRODUCTION

The international video coding standard H.264/AVC [1] can produce a perceptually equivalent quality video stream at less than half of the bit rate over a wide range of bit rates and video resolutions compared to previous standards [2], [3]. Due to its superior coding efficiency, it has been adopted in several important applications including video telephony, broadcast, video streaming, HD-DVD, Bluray Disc, and video storage.

This improved compression efficiency is obtained at the expense of increased complexity. On the encoder side, a tradeoff between encoding complexity and video quality can be achieved by the use of encoder optimization techniques such as fast motion estimation techniques, fast mode decision algorithms, etc. [4]-[6]. Decoder speedup is potentially of even greater interest and importance, since the market size for H.264/AVC introduces an adaptive in-loop deblocking filter to eliminate blocking artifacts while preserving the sharpness of the content.

Broadcast television and home entertainment have been revolutionised by the advent of digital TV and DVD-video. These applications and many more were made possible by the standardisation of video compression technology. The next standard in the MPEG series, MPEG4, is enabling a new generation of internet-based video applications whilst the ITU-T H.263 standard for video compression is now widely used in videoconferencing systems.

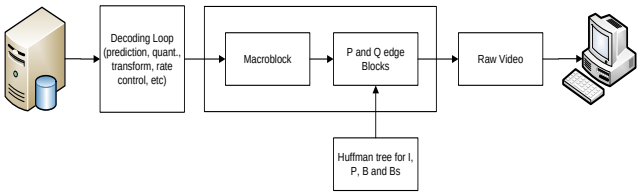

Fig.1 Top level Block Diagram

MPEG4 (Visual) and H.263 are standards that are based on video compression ("video coding") technology from circa. 1995. The groups responsible for these standards, the Motion Picture Experts Group and the Video Coding Experts Group (MPEG and VCEG) are in the final stages of developing a new standard that promises to significantly outperform MPEG4 and H.263, providing better compression of video images together with a range of features supporting high-quality, low-bitrate streaming video. The history of the new standard, "Advanced Video Coding" (AVC), goes back at least 7 years.

II. .H.264 CODEC

In common with earlier standards (such as MPEG1, MPEG2 and MPEG4), the H.264 draft standard does not explicitly define a CODEC (enCOder / DECoder pair). Rather, the standard defines the syntax of an encoded video bitstream together with the method of decoding this bitstream. In practice, however, a compliant encoder and decoder are likely to include the functional elements. Whilst the functions are likely to be necessary

for compliance, there is scope for considerable variation in the structure of the CODEC. The basic functional elements (prediction, transform, quantization, entropy encoding) are little different from previous standards (MPEG1, MPEG2, MPEG4, H.261, H.263); the important changes in H.264 occur in the details of each functional element

An input frame Fn is presented for encoding. The frame is processed in units of a macroblock (corresponding to 16x16 pixels in the original image). Each macroblock is encoded in intra or inter mode. In either case, a prediction macroblock P is formed based on a reconstructed frame. In Intra mode, P is formed from samples in the current frame n that have previously encoded, decoded and reconstructed (uF'n in the Figures; note that the unfiltered samples are used to form P). In Inter mode, P is formed by motion-compensated prediction from one or more reference frame(s).The reference frame is the previous encoded frame $F'n_{-1}$; however, the predicton for each macroblock may be formed from one or two past or future frames (in time order) that have already been encoded and reconstructed.

The decoder receives a compressed bitstream from the NAL. The data elements are entropy decoded and reordered to produce a set of quantized coefficients X. These are rescaled and inverse transformed to give Dn' (this identical to the Dn' shown in the Encoder). Using the header information decoded from the bitstream, the decoder creates a prediction macroblock P, identical to the original prediction P formed in the encoder. P is added to Dn' to produce uF'n which this is filtered to create the decoded macroblock F'n.

III. H.264 DEBLOCKING PROCESS

H.264 introduces a content adaptive in-loop filter to eliminate blocking artifacts while retaining true edges. Filtering is applied to each boundary of 4 × 4 luma and chroma block in a 16× 16 macroblock (MB). The deblocking process mainly consists of two parts, boundary strength (BS) derivation and edge filtering. BS determines the strength of the applied filter. No filtering is needed for boundaries with a BS value of 0, whereas the strongest filtering is performed for boundaries with a BS value of 4. For boundaries with nonzero BS values true edge detection is applied first, and only those determined as artificial edges induced by the compression are filtered. It presents the BS derivation process defined in the H.264 specification and implemented in the reference software. The basic idea behind the process is to apply a stronger filter at places likely to have more significant blocking distortion, such as the boundary of an intra-coded macroblock orthe boundary of a 4×4 block that contains coded coefficients. After the BS is derived, an analysis on the gradients of a set of eight samples across the boundary is performed to check whether the filtering should be skipped to preserve image sharpness. The error created by quantization is usually smaller than a threshold for a specific quantization parameter (QP). Therefore, for edges with nonzero BS values a pair of quantization-dependent parameters α and β are used in the content-activity check to determine whether each set of samples should be filtered, as illustrated in Fig. 3. Filtering on the set of eight samples takes place only if the three conditions $|p0 - q0| < \alpha$, $|p0 - p1| < \beta$, and $|q0 - q1| < \beta$ all hold. In Fig. 3, strong filtering refers to BS = 4, and normal filtering refers to BS = 1, 2, and 3.

Edge filtering is then applied for each set of eight luma pixels across the boundary by using one low-pass filter based on the derived BS value. Related chroma components are also filtered accordingly.

IV. DESCRIPTION OF RECONSTRUCTION FILTERS

A filter is applied to every decoded macroblock in order to reduce blocking distortion. The deblocking filter is applied after the inverse transform in the encoder (before reconstructing and storing the macroblock for future predictions) and in the decoder (before reconstructing and displaying the macroblock). The filter has two benefits: (1) block edges are smoothed, improving the appearance of decoded images (particularly at higher compression ratios) and (2) the filtered macroblock is used for motion-compensated prediction of further frames in the encoder, resulting in a smaller residual after prediction. (Note: intra-coded macroblocks are filtered, but intra prediction is carried out using unfiltered reconstructed macroblocks to form the prediction). Picture edges are not filtered.

Filtering is applied to vertical or horizontal edges of 4x4 blocks in a macroblock, in the following order:

- Filter 4 vertical boundaries of the luma component (in order a,b,c,d)
- Filter 4 horizontal boundaries of the luma component (in order e,f,g,h)
- Filter 2 vertical boundaries of each chroma component (i,j)
- Filter 2 horizontal boundaries of each chroma component (k,l)

Each filtering operation affects up to three pixels on either side of the boundary.It shows 4 pixels on either side of a vertical or horizontal boundary in adjacent blocks p and q (p0,p1,p2,p3 and q0,q1,q2,q3). Depending on the current quantizer, the coding modes of neighbouring blocks and the gradient of image

samples across the boundary, several outcomes are possible, ranging from (a) no pixels are filtered to (b) p0, p1, p2, q0, q1, q2 are filtered to produce output pixels P0, P1, P2, Q0, Q1 and Q2

V. BOUNDARY STRENGTH DERIVATION

The choice of filtering outcome depends on the boundary strength and on the gradient of image samples across the boundary. The boundary strength parameter Bs is chosen according to the following rules:

- p or q is intra coded and boundary is a macroblock boundary Bs=4 (strongest filtering)
- p or q is intra coded and boundary is not a macroblock boundary Bs=3
- neither p or q is intra coded; p or q contain coded coefficients Bs=2
- neither p or q is intra coded; neither p or q contain coded coefficients; p and q have different reference frames or a different number of reference frames **or** different motion vector values Bs=1
- neither p or q is intra coded; neither p or q contain coded coefficients; p and q have same reference frame and identical motion vectors Bs=0 (no filtering)

The filter is "stronger" at places where there is likely to be significant blocking distortion, such as the boundary of an intra coded macroblock or a boundary between blocks that contain coded coefficients.
frequency response of a linear-phase finite impulse response (FIR) filter using multiple copies of the same filter [5].

BS DERIVATION PARAMETER DEFINITIONS

Parameter	Defintion
C_{IS}SliceBoundary	= 1 if the boundary is a slice boundary, 0 otherwise
C_{IS}8×8TransformInnerBoundary	= 1 if the boundary is inside an 8 × 8 transform block, 0 otherwise
C_{IS}Intra*MB*	= 1 if P or Q is intra-coded, 0 otherwise
C_{IS}*MB*Boundary	= 1 if the boundary is a MB boundary, 0 otherwise
C_{CBP}Condition	= 1 if either of the two CBPs of P and Q equals to 1, 0 otherwise
C_{MV}Condition	= 1 if the two reference frames of P and Q are different or any component of the two motion vectors has difference more than one pixel, 0 otherwise

Table.1 BS Derivation Parameter Definitions

VI. FILTER DECISION

A group of samples from the set (p2,p1,p0,q0,q1,q2) is filtered only if:
(a) Bs > 0 and
(b) |p0-q0|, |p1-p0| and |q1-q0| are each **less** than a threshold □ □ or □ □ (□ □ and □ □ are defined in the standard [1]).

The thresholds □ □ and □ □ increase with the average quantizer parameter QP of the two blocks p and q. The purpose of the filter decision is to "switch off" the filter when there is a significant change (gradient) across the block boundary in the original image. The definition of a significant change depends on QP. When QP is small, anything other than a very small gradient across the boundary is likely to be due to image features (rather than blocking effects) that should be preserved and so the thresholds □ □ and □ □ are low. When QP is larger, blocking distortion is likely to be more significant and □ □ , □ □ are higher so that more filtering takes place.

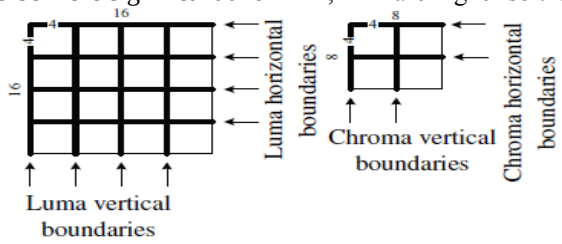

Fig. 2 Boundaries in a macro block to be filtered

VII. PROPOSED TECHNIQUE:HYBRID OPTIMIZATION

A. Huffman Optimization Technique

The existing technique i.e., the basic H.264 deblocking part makes the use of more conditional statements thereby leading to increased computational timing and more power consumption (Fig. 3)

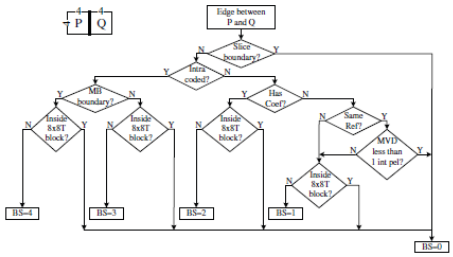

Fig. 3 Normal deblocking (reference)

The proposed technique employed to reduce the computational timing is Huffman Tree Coding method which is employed to reduce the computational timing more than that of the reference model.

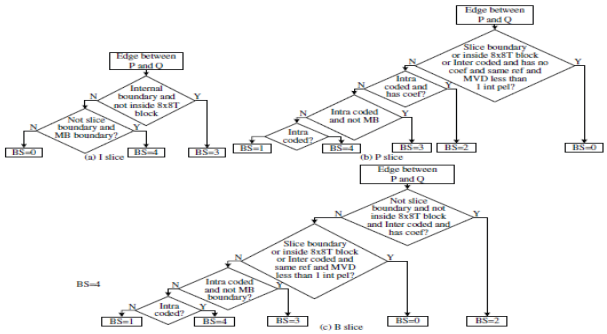

Fig. 4 Huffman Optimized computational timing due to less conditional block

In Huffman optimization technique, the number of conditional statements involved to carry out the deblocking in general method is reduced but without affecting the output. The result obtained after applying Huffman Optimization proved that the deblocking process is fastened upto seven times as compared to the base method.

B. Replacing Branch Operations with Combination Logics and Arithmetic Operations

It is highly desirable to replace one branch operation with a small amount of logic and arithmetic operations, since branch operations usually cause pipeline stalls and obstruct the of instruction-level parallelism, thus degrading the performance of parallel processors. The P-slice BS derivation process is used here as an example to explain the replacement of a branch operation by combination logics.

Note that the objective of the sequential operations in Fig. 3 is to derive a strength value for the boundary. Therefore, it can be regarded as a combination logic module where the input parameters are the MB position, MB transform type, MB type, edge position, coded block pattern, reference frame indices, and motion vectors. The output of this module is the BS value ranging from 0 to 4.

In Fig. 4(b), all the required input parameters are involved in the first step to check whether the BS value is 0 or not. Consequently, this can be formulated using a combination of logic and arithmetic operations to replace the branch operations. By extending this idea, the entire BS derivation process in Fig. 2 can be replaced with the pseudo code below

$BS = (!\ (C_{IS}\text{SliceBoundary}||C_{IS}\text{8×8TransformInnerBoundary}))*$
$(C_{IS}\text{Intra}_{MB} * \ (3 +!\ C_{ISMB}\text{Boundary}) + (!\ C_{IS}\text{Intra}_{MB}\ ) *$
$(C_{CBP}\text{Condition} * \ 2 + (!\ C_{CBP}\text{Condition}) * \ C_{MV}\text{Condition}))$

C. Hybrid Scheme for H.264 Conforming High-Speed Deblocking

Based on these techniques, a hybrid scheme composed of logic, arithmetic, and branch operations was proposed. Significant computation saving was achieved while preserving conformance to the standard. The proposed scheme was shown to reduce the deblocking computational load by more than seven times in comparison with simplified H.264 reference software. The proposed hybrid approach can be incorporated into any H.264 decoder to save computation, thus leading to less power consumption and higher decoding speed. In particular, larger gains are achieved on most modern processors where the branch operations are much more expensive than logic and arithmetic operations.

VIII. CONCLUSION

The in-loop deblocking process is identified as the most computationally expensive part of the H.264 decoder. The reference deblocking process given in the H.264 specification and reference software is not computationally optimal. The data collected from several video bit streams demonstrate biased statistical distributions. We proposed a statistics-dependent decoding technique to achieve encoder/platform-independent decoder speedup. Huffman tree structures for the boundary strength derivations were introduced and analyzed. Similarly, statistics-dependent decoding was applied to true-boundary detection. We also proposed the use of combination logics to replace the branch operations to further reduce complexity.

REFERENCES

[1] Advanced Video Coding for Generic Audiovisual Services, ITU-T Rec. & ISO/IEC 14496-10 AVC, version 4, 2005.
[2] Lin and T. Chang, "Fast block type decision algorithm for intra prediction in H.264 frext," in Proc. IEEE Int. Conf. Image Process., vol. 1. Genoa, Italy, Sep. 2005, pp. 585–588.
[3] G. N. Rao, "Real-time software implementation of H.264 baseline profile video encoder for mobile and handheld devices," in Proc. IEEE Int. Conf. Acoustics, Speech Signal Process., vol. 5. Toulouse, France, May 2006, pp. 457–460.
[4] K. Ugur1, "Generating H.264/AVC compliant bitstreams for lightweight decoding operation suitable for mobile multimedia systems," in Proc. IEEE Int. Conf. Acoustics, Speech Signal Process., vol. 2. Toulouse, France, May 2006, pp. 33–36.