HRD compliant single pass H.264/AVC VBR Encoding Mode

Sourya Bhattacharyya#, Subarna Tripathi#, Emiliano Mario Piccinelli*

(Advanced System Technology labs, STMicroelectronics, India#, Italy*)

{sourya.bhattacharyya, subarna.tripathi, emiliano.piccinelli}@st.com

Abstract: The goal of Variable Bit-Rate (VBR) encoding is to maintain a constantly high visual quality within the target bit-rate, during whole encoding process, thus saving and accumulating bits during low complexity scenes and reusing those bits in higher complexity scenes. Such a goal can be quite easily achieved by off-line encoding, since it is possible to measure the whole sequence complexity distribution a priori and to use multi pass processing algorithms; however, real time encoding requires completely different approaches. This paper describes a single-pass, real-time VBR control method that achieves excellent and constant visual quality for H.264/AVC online encoding. It is also totally compliant with the Hypothetical Reference Decoder (HRD) model of H.264/AVC.

1 Introduction

In [3] a single pass, real time Variable Bit-Rate (VBR) control method is presented, which is suitable for MPEG-2 DVD recorders. Main novelty of that approach is usage of “BitBudget” concept instead of the more classic “global target bit allocation” stage. In this paper we will show that the VBR method can successfully be applied also to H.264/AVC encoding [1]. Furthermore, we will illustrate that the VBR control method herewith described is compliant with the Hypothetical Reference Decoder (HRD) model of H.264/AVC, so proposing a successful procedure to manage the HRD of H.264.

2 Basic Principles of VBR encoding

A video picture is “complex” if it contains object details or information not easily predictable from other pictures with usual spatial or temporal techniques. Image complexity is defined as $X_t = S_t \cdot Q_t$, with $S_t$ = number of bits to encode picture of type “t” (I, P or B) and $Q_t$ = related average picture quantization step. Thus, fixed the image complexity, lower the QP value, larger the amount of bits S necessary to encode the picture and better the visual image quality. Thus, $Q_t$ is an indicator of image detail and quality. According to the above definition of picture complexity, experiments [3] show that the quality of encoded sequence is related to the quantization step: two different sequences both encoded with the same constant quantizer will have same visual quality; output average bit rate is proportionately related to sequence complexity (with fixed quantizer, of course).

If the video sequence is long enough, rate control module can save bits during compression of low complexity scenes and spend later those saved bits in more complex scenes. This “complexity compensation” based rate control procedure can guarantee constant quantization (i.e. high and constant visual quality, in both easy and critical conditions) and also match the average target bitrate. Unknown distribution of pictures complexity along the sequence is the crucial problem in real-time, single-pass encoding: it is impossible to know if the amount of bits saved can balance all the possible variations in complexity that will occur up until the end of encoding. On the other hand, such goal can be “easily” achieved by more expensive multi-pass encoding systems: they can measure the whole sequence complexity distribution during one encoding step and reuse information in following encoding steps.

2.1 The budget of bits

If output average target bit rate available is $X$ Mb/sec, and we start encoding at an instantaneous output bit rate $Y$ Mb/sec ($Y < X$) for initial less complex pictures, with maintaining high output visual quality, then we are saving $(X-Y)$ Mb/sec even if the visual quality is high. In this ideal situation, the amount of bits accumulated (budget) can be used to maintain the quantizer constant even later when sequence complexity increases to generate instantaneous bitrate greater than the target average one.
Our VBR control procedure manages this process by using the BitBudget curve, as illustrated in Figure 1. The mathematical expression of BitBudget is:

\[ \text{BitBudget} (T) = \int_{0}^{T} (\text{TarBitrate} - \text{IstBitrate}) \, dt \]

where, TarBitrate and IstBitrate are respectively the target and instantaneous bit rates, whereas T is the time interval.

Figure 1: Example of BitBudget diagram with a change in complexity after 10 seconds.

Figure 1 refers situation where X = 5 and Y = 2. Up until t = 10s, the surplus is X - Y = 3 Mbit/s, which means 30 Mbit saved in total. At t = 10s the sequence complexity increases. Let’s say that instantaneous bit rate Y increases from 2 Mbit/s to 6 Mbit/s: so we have a bitrate loss = Y – X = 1 Mbit/s. Starting from the 30 Mbit previously saved, rate control can hold constant quantizer (and therefore the visual quality) for next 30 seconds (30 Mbit / 1 Mbit/s = 30 sec).

3 Our proposed VBR algorithm

Based on input parameters like -
1) Average target Bit rate,
2) Maximum instantaneous Bit rate,
3) Minimum instantaneous Bit rate,
- our VBR control algorithm keeps quantization parameter (QP) rigorously uniform on the whole picture and smoothly adapts it to image content changes on a frame-by-frame basis, allowing constant quality along various Group Of Pictures (GOP) of the same scene. The BitBudget curve is measured frame after frame and divided in six zones. For every zone, a dedicated Look-Up-Table (LUT) contains the allowed quantization updates \( q_{upd} \), depending on -
1) BitsBudget slope,
2) Current zone within bit budget curve,
3) Reference quantizer \( Q_{REF}(n-1) \) which was applied to the previous frame n-1.
Reference quantizer of the current frame n is \( Q_{REF}(n) = Q_{REF}(n-1) + q_{upd} \).

This set up produces constant quality (CQ) video output in VBR control, as tested in various test sequences containing different amount of movement complexity, from extremely fast to pseudo-static. Test sequences have GOP size as 12, with I/P frame distance as 3, as in standard GOP structure.

Table 1 shows the typical behavior: CQ mode uses higher quantizers and usually keeps low their variance, ensuring more uniform visual quality, but does not always exploit whole available bitrate for short length sequences. In our approach, QP adjustment is managed at slice (a slice is entire row of macro-blocks) level.

4 Managing the HRD of H.264/AVC

4.1 Background

The Hypothetical Reference Decoder (HRD) is the virtual decoder that verifies H.264/AVC stream compliancy [2]. It consists of a Hypothetical Stream Scheduler (HSS), a Coded Picture Buffer (CPB), an instantaneous picture decoder, a Decoded Picture Buffer (DPB) and a display unit (Figure 2). Whatever rate control method applied by the encoding system, either CBR or VBR, H.264/AVC requires that the CPB buffer will never overflow and underflow.

The HRD operates as follows (Annex C of H.264/AVC standard [2]). HSS delivers data associated with Access Units (AU) that flow into CPB according to a specified arrival schedule; every AU contains exactly one picture. Data associated with each AU are removed and decoded instantaneously by the instantaneous decoder at CPB removal times. Then, each decoded picture is placed in the DPB at its CPB removal time unless it is output to the display (at its DPB output time) and it is a non-reference picture. When a picture is placed in the DPB, it is removed from the DPB at its CPB removal time or at the time when it is marked as "unused for reference". Every HRD parameter is carried in Video Usability Information (VUI) and Supplemental Enhancement Information (SEI) messages. These messages can be inserted into the H.264/AVC bit-stream itself or transmitted to the decoder through a different channel.
The most important HRD parameters are:

- **bit_rate_value**: maximum input bitrate for CPB.
- **cpb_size_value**: specifies the CPB size.
- **initial_cpb_removal_delay**: CPB delay between the arrival time of the first bit (of AU associated coded data) and the removal time of the coded data associated with the same AU, after HRD initialization.
- **cpb_removal_delay**: delay to wait, before removal of AU data associated with the picture timing SEI message, from the buffer.

### 4.2 Leaky buckets

H264/AVC offers possibility to decode the same bit stream using bit rates different from the encoding nominal one. The set of transmission rate (R), buffer size (B), and initial buffer fullness (F) parameters, from which depends the initial delay, is so called “leaky bucket” (R, B, F). An encoder can generate a bit-stream that compliances to N different leaky bucket and signals these leaky bucket through VUI-SEI messages. The decoder can choose among N leaky buckets or interpolate between two of them in order to match as much as possible the desired transmission rate (R), buffer size (B), and initial buffer fullness (F). We have used single leaky bucket model. Nevertheless, adapting our algorithm to multiple leaky buckets should be straightforward.

### 4.3 Our HRD management proposal

In addition to minimal quantizer update (for constant quality encoding), based on bit budget curve and reference quantizer, we have implemented a timing model for management of AU removal times from CPB and monitoring HRD buffer fullness in a frame by frame manner. Monitoring AU arrival and removal times makes it possible to satisfy HRD constraints even in low target bit rate conditions, without violating constant quality encoding procedure.

Our timing model synchronizes instantaneous encoder buffer fullness with that of HRD at each AU’s arrival or removal. HRD buffer fullness is initialized zero at time zero, and updated based on arrival and removal times of incoming AUs. HRD buffer fullness is incremented at full or partial arrival of an AU in CPB. When an AU is removed instantaneously from CPB at its removal time, HRD buffer fullness is decremented by corresponding AU size.

We use circular linked list queue to monitor and store information of access units that have
arrived fully or partially but yet to be removed from CPB. Based on queue information, we derive instantaneous HRD buffer fullness. Also to adjust removal time of each AU for satisfying buffer constraints, we set initial\_cpb\_removal\_delay in such a way so that decoding should ideally start when the CPB fullness reaches half of cpb\_buffer\_size. 

\[
\text{initial\_cpb\_removal\_delay} = \frac{\text{cpb\_buffer\_size}}{2}\text{bit\_rate}
\]

if ( numAU == 0 ) {
    cpb\_removal\_delay = 0;
} else if (pict\_structure != H264\_FRAME) {
    cpb\_removal\_delay += (AU\_bit\_used / (Avg\_Bit\_Pict / 2)) + 1;}
else {
    cpb\_removal\_delay += (AU\_bit\_used / Avg\_Bit\_Pict) * 2 + 2;}

Figure 4: Pseudo-code for removal delay adjustment

From Figure 4, CPB removal delay in Picture Timing SEI message is reset at each buffering period, incremented by 2 for frame picture and by 1 for field picture. Field clock unit used is equal to 1 / (2 \* frame rate).

We adjust current AU's cpb removal delay value in such a way so that AU removal time will always be greater than its final arrival time, preventing HRD buffer underflow occurrence. Above mentioned procedure suggests that frame encoded with more number of bits than its average allocated will be removed from the CPB later. Thus removal time difference for 2 frames can be more than 1/(frame rate) i.e. the perceptual frame-rate can become less than specified input frame rate value. So, duration of whole sequence can become larger than the specified input (equal to total number of frames divided by frame rate). But for standard bitrate and for standard resolution sequence, where average bit per picture is close to actual bits to encode the picture, the visual experience is good and not at all annoying and at the same time it always satisfies HRD constraints.

5 Results and Conclusion

Table 1, Figure 3, and Figure 5 show the CQ results (the specified maximum and minimum instantaneous bitrates are 50% more and 30% less than the average target bitrate).

![Figure 5: QP range for sequence Demoiselle (H.264 baseline, total frames 100, and 25 fps) with average bitrate 1 Mbps, maximum and minimum instantaneous bitrate specified are 50% more than average bitrate and 30% less than the average bitrate.](image)

Our proposed method thus generates constant quality HRD compliant streams being within the bound of the average target bitrate.

<table>
<thead>
<tr>
<th>Seq</th>
<th>Total frames</th>
<th>QP range</th>
<th>Target bitrate (Kbps)</th>
<th>Actual bitrate (Kbps)</th>
<th>Deviation (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>films</td>
<td>908</td>
<td>26-29</td>
<td>1000</td>
<td>903.2</td>
<td>-9.68</td>
</tr>
<tr>
<td>foreman</td>
<td>250</td>
<td>27-27</td>
<td>1000</td>
<td>1047.8</td>
<td>4.78</td>
</tr>
<tr>
<td>philips</td>
<td>100</td>
<td>26-30</td>
<td>1000</td>
<td>1104.5</td>
<td>10.45</td>
</tr>
<tr>
<td>starwars</td>
<td>100</td>
<td>27-27</td>
<td>1000</td>
<td>1137.4</td>
<td>13.74</td>
</tr>
<tr>
<td>rotate</td>
<td>395</td>
<td>23-27</td>
<td>1000</td>
<td>526.6</td>
<td>-47.33</td>
</tr>
</tbody>
</table>

Table 1: QP range used

6 References