RC22382 (W0203-080) March 25, 2002
Electrical Engineering

IBM Research Report

One-pass VBR Video Encoding Algorithm

Krishna Ratakonda, Ashish Jagmohan

IBM Research Division
Thomas J. Watson Research Center
P.O. Box 218
Yorktown Heights, NY 10598

Research Division
Almaden - Austin - Beijing - Delhi - Haifa - India - T. J. Watson - Tokyo - Zurich

One-Pass VBR Encoding Algorithm

Ashish Jagmohan, Krishna Ratakonda

15th August 2001

Introduction

In CBR encoding the short-term average bitrate for the output bitstream is constrained by the presence of a fixed size buffer at the decoder (called the vbv buffer). The deviation of the actual bitrate from the average bitrate is bounded by the necessity of avoiding underflow or overflow of the vbv buffer. CBR encoding is used whenever the said decoder buffer constraints apply, for example, in real time videoconferencing.

In VBR encoding the corresponding constraint is on the total bitrate used for the entire bitstream, rather than the short-term average bitrate.[1] This is a looser constraint bounding, as it does, the long-term average bitrate of the output rather than the average short-term bitrate. Since the only constraint in the VBR case is the constraint on the total bitrate, VBR encoding is equivalent to CBR encoding using a very large vbv buffer. Typical video sequences are non-stationery, with subsequence complexity varying widely over the stream. Thus different subsequences of successive frames should be ideally coded at different short-term average bitrates, so as to produce an output sequence of almost constant quality. VBR encoding is inherently better-suited to this scenario than CBR encoding. This is because it avoids the tight control over short-term average bitrate required in CBR video encoding. Thus VBR encoding can potentially produce a better average quality for the same long-term bitrate as compared to CBR encoding.

VBR encoding can be done using one or more encoding passes over the input video bitstream. When multiple passes are used (usually two), the final output bitstream is produced by the last encoding pass. The other encoding passes are used to determine the statistics of the input video bitstream. These statistics include the average inter-pixel difference and frame mean of the video frames and the complexity of the video frames (defined as the product of the quantization scale used to encode the frame and the number of bits used to encode the frame). In the final encoding pass, the quantization scales of the video frames are selected, on the basis of the statistics gathered in previous passes, such that the almost-constant quality condition mentioned above is satisfied. For MPEG encoding, this is approximately equivalent to maintaining an almost constant quantization scale over the frames of the video. The main advantage of using multiple passes is that the knowledge of statistics garnered can be used quite effectively to ensure this. The main disadvantages of using multiple encoding passes are the added complexity (in terms of both time and computation) and the inability to perform such an encoding in a real-time manner, since access to the entire video stream is needed, before the final encoding can begin. Thus multiple pass VBR encoding is mostly useful when there are no time and complexity constraints, for eg. in encoding DVDs.

One-pass VBR encoding offers a compromise between low complexity, low quality CBR encoding and high complexity, high quality multiple pass VBR encoding. In one-pass VBR encoding the statistical information is gathered in the same pass in which the output video bitstream is generated. In other words, the statistics of the video frames already encoded are used to decide the quantization scale of future frames. The encoding quality of one-pass VBR is not as good as that of multiple-pass VBR - the non-stationarity of video implies that basing your encoding decisions

[1] It should be noted that the total bitrate constraint only applies in the encoding applications considered in this report. Other encoding applications, such as VBR video transmission over ATM networks may have additional constraints imposed, for example, leaky-bucket constraints on the maximum burst rate and the short-term rate.

on past statistics is imperfect. However, the encoding quality of one-pass VBR is better than that of CBR because of the looser buffer constraints mentioned before. The complexity tradeoff of CBR, one-pass VBR and multiple-pass VBR is the other way around. One-pass VBR is ideal for applications where the short-term average bitrate is not constrained and where the demands of low-complexity and/or real-time encoding imply that that multiple pass VBR cannot be used. An example is recording digital video onto a hard-disk off a broadcast video stream. Since the read speed from a hard-disk is much higher than the maximum output bitrate of the encoded stream, the decoder buffer size is not a constraint. Moreover, real-time recording means that multiple encoding passes cannot be used.

In this report, we propose a one-pass VBR encoding algorithm for a scenario, like the above, in which the total bitrate is fixed but there are no constraints on theshort-term average bitrate. The goal is to produce an almost constant quality compressed video which has as high a quality as possible under the above constraint.

Algorithm

Overview

First we present an overview of the proposed algorithm. The proposed one-pass VBR algorithm encodes an MPEG-4 video stream consisting of I and P frames, under a fixed total bitrate constraint. The bit allocations for frames are selected so as to gradually force the bitrate used to the allowed average bitrate. The period over which the algorithm endeavours to do this is given by the length of a sliding window. The algorithm encodes the current video frame in two steps.

In the first step, the total number of bits to be used for encoding the frame is computed. This is called the frame target. The frame target is computed on the basis of

- The length of the sliding window
- The deviation of the average bit-rate used to encode the previous frames of the video from the allowed long-term average bitrate
- The average motion and quantization scale observed in the previously encoded video frames
- The motion and the quantization scale in the current frame, ie the frame which is to be encoded
- The type of the current frame, ie if its an I frame or a P frame.

In the second step, quantization scales are selected on a per-macroblock basis, such that the frame target calculated in the first step is met. In actuality, this step consists of two substeps. First the global quantization parameter is selected, based on

- The motion and the amount of bits used to encode the last frame of the same type as the current frame
- The motion in the current frame and the frame target calculated above.
- The average quantization scale used to encode the last frame of the same type as the current frame.

Secondly the quantization scale for a particular macroblock is selected, based on

- The global quantization parameter computed above
- The activity of the current macro-block
- The amount of over-production or under-production in the number of bits used to encode the previous macroblocks of the current frame.

The calculated quantization scale is clipped to a minimum and maximum value. The rationale for this is, that rather than increase the quantization scale for a frame beyond a point, it is preferable to encode a number of frames using a moderately high quantization scale.

In addition to the quantization scale, the encoder also selects the resolution at which the output video bitstream is to be encoded. The selection of the resolution is based on

- The excess number of bits used to encode the previous frames of the video compared to the allowed long-term average bitrate
- The rolling motion average
- The quantization scale used to encode the previous frames

Different criteria, based on the above statistics, are used to decide when to begin encoding in low-resolution and when to revert back to high-resolution.

Details - Frame Target Allocation

The rate model assumed in the following, is

$$R = aQ^{-1}M \tag{1}$$

where a is a constant, Q is the average quantization scale for a frame, M is a coding dificulty measure and R is the number of bits used to encode the frame.For P frames M is computed as the sum of absolute values of the motion-compensated residual. For I frames, M is calculated as the sum of absolute values of the I frame coefficients.

Let B_{tot} be the total number of bits allowed to encode the video stream. To ensure that the entire video stream is encoded and no frames need to be dropped, a certain amount of extra space is reserved, which can be used if B_{tot} is exceeded. Let B_{ex} denote this extra space. Let F_{tot} be the total number of frames to be encoded. Then the long-term average allowed bitrate is given by $R_{av} = \frac{B_{tot}}{F_{tot}}$. The maximum allowed bitrate is given by $R_{max} = \frac{B_{tot}+B_{ex}}{F_{tot}}$.

Assume that, at the current time instant f frames have been encoded and frame number $f+1$ is to encoded next. Let $R_{act}(f)$ denote the average bitrate actually used to encode the f frames. Then the amount of bits that have been overused, compared to the allowed average bitrate, are

$$\Delta = (R_{act}(f) - R_{av}).f \tag{2}$$

Let the length of the sliding window be given by l_w. Then, as mentioned in the summary, the algorithm makes frame allocations with a view to forcing the actual bitrate to the allowed average bitrate over the length of the sliding window. In other words, the aim is to ensure

$$R_{act}(f + l_w) = R_{av} \tag{3}$$

This can be ensured by allotting to the current frame, a target given by

$$T_{curr} = R_{av} - \frac{\Delta}{l_w} \tag{4}$$

The allocation given by 4 will force the actual bitrate to converge to the allowed average bitrate over the next l_w frame, *if* the next l_w frames all have the same complexity as the current frame. This will, in general, not be true. The problem is that in one-pass VBR, no statistical information about the next l_w frames is known, However, some way of estimating this information is required, to correctly compute a frame allocation for the current frame. We solve this problem by assuming that both f and l_w are large enough, such that the statistical information derived from encoding the f frames is representative, to some extent, of the expected statistics of the next l_w frames - at least in the sense that the mean statistics are the same.

Then, the frame target allocation to the current frame is

$$T_{av} = min(\frac{R_{av}}{3}\ ,\ T_{curr} * k * tan^{-1}(k_M * k_Q * tan(\frac{\pi}{6}))) \tag{5}$$

In the above equation, the constant k depends on the proportion of available excess bits P_{ex} that has been used up so far, ie

$$k = median(\frac{2}{\pi}\ ,\ \frac{2}{\pi} * (3 - 2 * P_{ex})\ ,\ \frac{6}{\pi}) \tag{6}$$

where

$$P_{ex} = \frac{R_{act} - R_{av}}{R_{max} - R_{av}} \tag{7}$$

k_M compares the current motion to the average motion observed so far

$$k_m = median(0.5\ ,\ \frac{M_{av}}{\widehat{M_{av}} + \zeta}\ ,\ 2) \tag{8}$$

In equation 8

$$M_{av} = \frac{N_I M_I + N_P M_P}{N_I + N_P} \tag{9}$$

where M_I and M_P are the values of M for the last I and P frames respectively. N_I and N_P are the number of I and P frames present in a GOP. Also,

$$\widehat{M_{av}} = \frac{N_I \widehat{M_I} + N_P \widehat{M_P}}{N_I + N_P}$$

where $\widehat{M_I}$ and $\widehat{M_P}$ are the average M values for I and P frames estimated over the last f frames. ζ is a small value.

Going back to equation 5,

$$k_Q = \frac{Q_{av}}{\widehat{Q_{av}}}$$

where Q_{av} and $\widehat{Q_{av}}$ are defined similarly to M_{av} and $\widehat{M_{av}}$, except that they are in terms of the quanztization scales rather than the M parameters.

Finally, the frame type information is incorporated into the frame allocation

$$T_{fr} = \frac{X_I}{N_I X_I + N_P X_P + N_B X_B} . T_{av} . (N_I + N_P + N_B) \tag{10}$$

where X is the frame complexity, similar to the complexity measure used in MPEG2. T_{fr} is the frame target used to encode the current frame. The amount of allowed overproduction D_{fr} is set equal to $max(T_{av}, T_{fr})$ while the allowed underproduction C_{fr} is set equal to $\frac{T_{fr}}{2}$.

Details - Quantization Scale

In accordance with the rate model given by equation (1), the global quantization scale for the current frame is calculated as

$$Q_g = \frac{M_{curr}}{M^j_{last}} . \frac{R^j_{last}}{T_{fr}} . Q^j_{last} \tag{11}$$

Here M_{curr} is the M parameter for the current frame. M^j_{last} is the M parameter for the last encoded frame of the same type as the current frame (thus j can be I or P). R^j_{last} and Q^j_{last} are the rate and the average quantization scale repectively of the last encoded frame of the same type as the current frame.

The quantization scale for a given macroblock is calculated in exactly the same manner as that for the MPEG-2 rate control with one important difference - the quantization scale is clipped to a minimum and maximum value, currently 5 and 15.

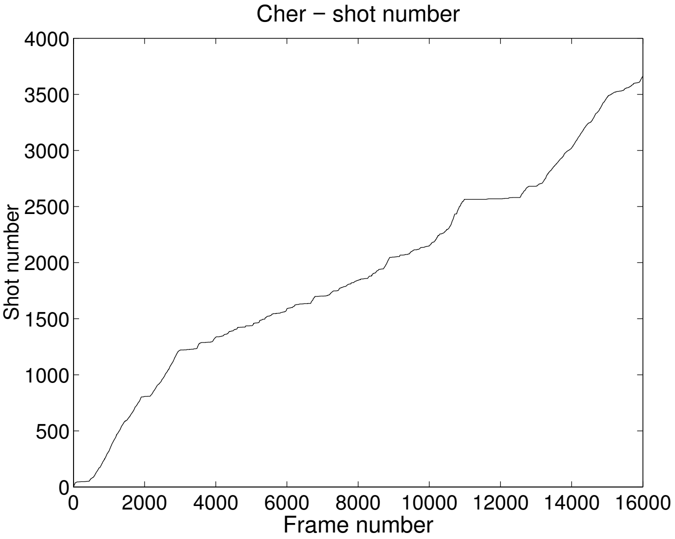

Figure 1: Scene changes for Cher sequence

Details-Reduced Resolution

Finally, we describe the conditions for switching into low resolution and switching back to high resolution. The resolution selection is done prior to the frame target allocation.

The above decisions are made on the basis of the following parameters

- Pex defined by equation (7) .
- M_{roll} which is a rolling motion average. (The motion estimator used is the sum of absolute values of the motion-compensated residual). The forgetting factor is currently set to $\frac{14}{15}$.
- F_{shot} which is the frame number at which the last scene change occurred.
- Q_{last} which is the average quantization scale used to encode the last frame. In case the last frame was encoded in reduced resolution, it's effective quantization scale (which is the value used in all calculations) is assumed to be three times the quantization scale actually used.

Then, the condition for switching into low reolution mode is

$$C_1 = (P_{ex} > 0.2)\ \&\&\ ((P_{ex} > 0.9)\,||\,(\frac{M_{roll}^2}{64} + \frac{(P_{ex} - 0.2)}{0.8} > 1.0))$$

The condition for switching out of low resolution mode is

$$C_2 = (P_{ex} < 0.9)\ \&\&\ (Q_{last} < 21)\ \&\&\ (M_{roll} < 6.0)\ \&\&\ (\frac{M_{roll}^2}{64} + \frac{(P_{ex} - 0.2)}{0.8} < 0.75)\ \&\&\ (F_{curr} - F_{shot} > 2)$$

Results

The figures show the results for 1-pass VBR encoding of the *Cher* sequence. Fig 1 shows the scene changes for the sequence. As can be seen, nearly 1 in 4 frames is a scene-change. A high frequency of scene-changes translates into high coding difficulty - ie a large bitrate is required to code the sequence at a reasonable quality. The bitrate used for VBR encoding was 750 kbps at 24 fps.Fig 2 shows R_{act}, R_{max} and R_{av} for the sequence. As can be seen R_{act} (the actual average bitrate used) gradually converges to a value between R_{max} and R_{av} as required. Fig 3 and Fig 4 show how the motion and excess bits used effect the swithching of the RRV mode. The motion measure is a rolling motion average (with forgetting factor of $\frac{14}{15}$) and the excess bits are given as a fraction of the total number of excess bits present.

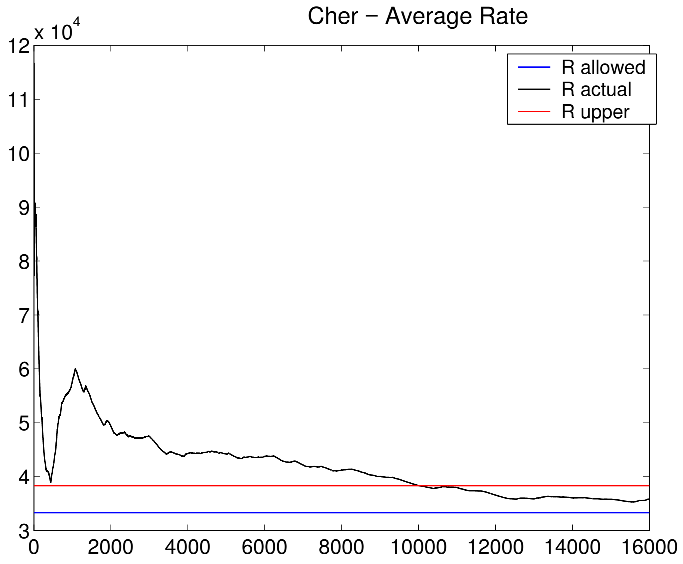

Figure 2: R_{act}, R_{max} and R_{av} for the Cher sequence

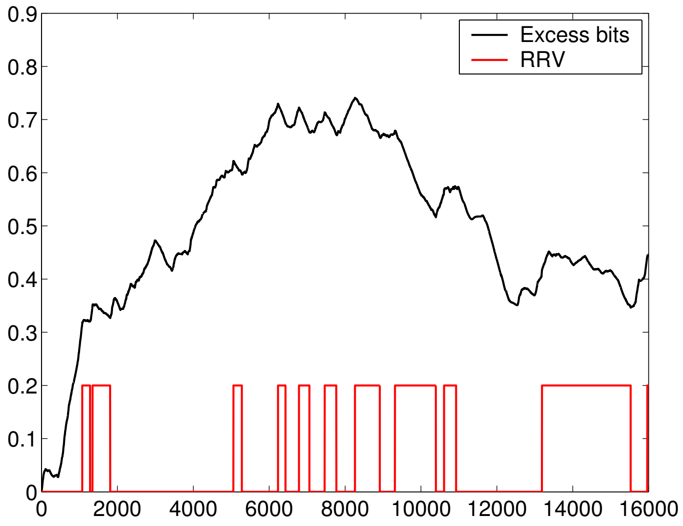

Figure 3: Excess bits, RRV mode for Cher sequence

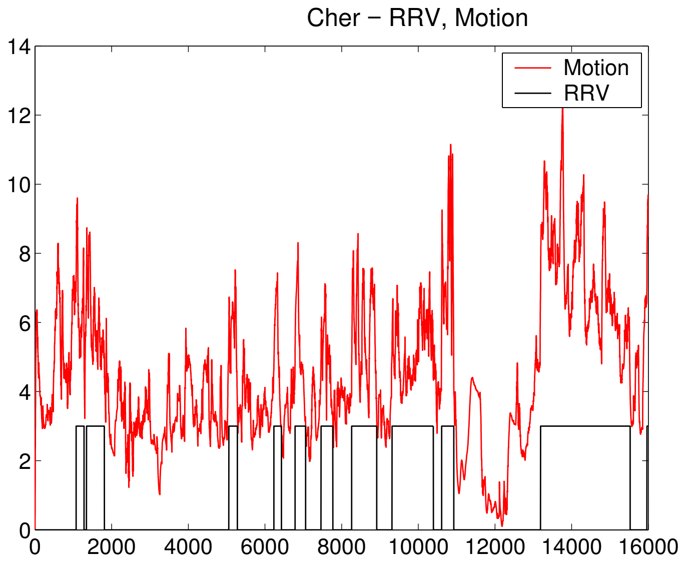

Figure 4: Motion, RRV mode for Cher sequence