MPEG-4 Real Time FD-CD Transcoding

Yong Wang*, Jae-Gon Kim#, Shih-Fu Chang*

*Department of Electrical Engineering, Columbia University, New York, USA
#Electronics and Telecommunications Research Institute, Daejeon, Korea

Abstract:

In this report we summarize our work on real time transcoding implementation combining frame dropping (FD) and DCT coefficient dropping (CD). FD can realize the temporal down sampling resulting in a low frame rate video stream, while CD is implemented by truncating the high frequency AC DCT coefficient bits. The video quality is optimized by adopting the Lagrange optimization. We also specify the issues of rate allocation within one GOP. One advantage of this FD-CD transcoding is its light weighted computational complexity. We present some experiment results, which demonstrate that our FD-CD transcoding can achieve satisfied output quality and promising computational efficiency.

1. Introduction

In the framework of universal multimedia access, one challenge for video transmission (communication) is to deliver video content through heterogeneous network channels matching the diversity of client devices. In order to do this it is often necessary to transcode the original bit stream. Media transcoding is a process adapting original media into a new version and meanwhile matching the resource (e.g., bandwidth) constraint or user's preference. Many adaptation methods exist for adjusting the bit rate of compressed video streams. For example, requantization of DCT coefficients, frame dropping, DCT coefficients dropping, and resolution reduction are commonly used. There is another family of transcoding, which deals with the transcoding between different video formats. We will not discuss it here.

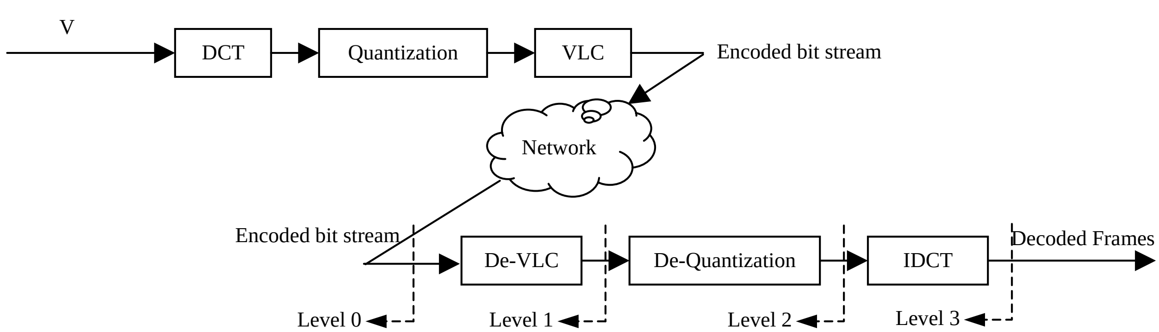


Figure 1: Conceptual datagram of a MPEG video coding system

Figure 1 is a conceptual datagram for a MPEG video coding system. Generally the image frames in a video stream will undergo DCT transformation, quantization and VLC entropy coding. In the decoding side, an inverse operation will be employed to get the decoded frames. In order to transcode the encoded bit stream, there are several ways depending on where the rate shaping operation is utilized.

The most straightforward way is to fully decode and then re-encode the frames at the level 3 in figure 1. It is quite predictable that by this way the transcoded can achieve optimized video quality by adopting the rate control in the encoding procedure. However, obviously the disadvantage of this method is its considerable computational complexity and cascade processing delay, which makes it inapplicable in a real time scenario. In order to reduce the computation cost, some variances at this level by reusing the motion vector information are discussed in [2]. But the workload like IDCT and optimization calculation is still non-ignorable.

Another widely considered transcoding method is re-quantization based on the de-quantized DCT coefficients at the level 2 in figure 1. The basic idea is to increase the quantization step to achieve a coarser version of original bit stream. The optimization is accomplished by selecting suitable steps. Since the optimization space is smaller than at the level 3, the transcoded video quality will reach a sub-optimum and perform worse than the re-encode case. But it is a good balance between the computational complexity and the video quality and thus extensively used in the literatures [3, 4, 5, 6].

If a more light-weighted transcoder is needed, the expenditure on de-quantization can also be stinted, by high frequency DCT coefficients dropping (CD) right after the variable length decoding at level 1 in figure 1. The remarkable advantage of this method is its computational economy. And also according to the result reported in [7], when the rate reduction is not severe, the transcoding in this level might perform better than requantization approaches.

The transcoders at level 1~3 are of somewhat fine granularity scalability in sense that they adjust the target bit rate finely. Sometimes when dramatic rate changing is needed, these methods might not be suitable any more due to their inevitable coding cost for overhead, motion vectors and baseline quality. In this case a rougher transcoding method, frame dropping (FD), is considerable. I.e., the bits from a specific frame are all truncated. Compared with the spatial resolution down-sampling algorithm [8], FD is a temporal down-sampling method to achieve big scale bit rate changing. For uniform system view, FD can be considered as a transcoding method introduced on the original bit stream without any decoding processing at level 0 in figure 1.

FD and CD are both light weighted transcoding method and a combination of them can achieve rate shaping in a wide range while keeping satisfied spatial-temporal video quality. In this paper, we will extensively discuss the proposed FD-CD combined transcoding method for MPEG-4 video. MPEG-4 is the latest video coding standard from MPEG [1], which mainly aims at network video applications. In such a scenario, a fast transcoder with satisfied quality is necessary to match the requirement of the universal multimedia access.

The rest of this report is organized as follows: in section 2, we describe the details of FD-CD combined transcoding; the involved rate control issue is discussed in section 3; the experiment result is given in section 4 and section 5 concludes the report.

2. FD-CD transcoding

2.1 FD

FD is a temporal down-sampling method by dropping some frames to achieve big scale bit rate changing. Recently, content-based approaches have been proposed to preserve the quality of frame-dropped stream as much as possible by dropping less important frames [9]. On the other hand, when an anchor frame (I or P frame) is dropped, the following frames that refer to the dropped frame should be re-encoded. In order to reduce the complexity of re-encoding for this case, the re-use of the existing motion vectors has been investigated in the previous works [10].

We adopt more straightforward operations of FD that drops B and/or P frames that do not have the decoding dependency, in the unit of group of pictures (GOP), by taking into account the GOP structure. FD provides only a coarse approximation to the target rate with several limited reducible rates since the smallest unit of data that can be removed is an entire frame. For instance, we define a set of FD operations in the case that GOP has the anchor frames distance, M = 3: no dropping, one B frame dropping in each sub-GOP, all B frames dropping, and all B and P frames dropping. The defined operations evenly distribute the dropped frames in the temporal range results in more comfortable temporal quality.

One issue involving FD is the rate control. In short, the truncated bits belonging to the dropped frame should be carefully considered in order to reach a fine bit rate. In section 3, we will have an extensive discussion on this issue.

2.2 CD

Our CD work is based on the previous work of dynamic rate shaping (DRS) [7] as spatial adaptation since it is more amenable to fast processing than requantization that leads to recoding-like algorithms. More specifically we assume that a set of high frequency DCT coefficient bits run-length coded at the end of each block is eliminated, which is the constrained DRS case in [7]. There is another unconstrained DRS as indicated in figure 2(b), where dropping an arbitrary set of coefficients is considerable and done by optimization search. Because the zigzag-scanning pattern of DCT block provides a quite successful ordering of the DCT coefficients according to their importance, most of time the sub-optimum achieved by constrained DRS vary very slightly from the unconstraint scheme in terms of decoded video quality [7].

Another important conclusion we adopted from [7] is the memoryless modeling in the rate-distortion formulation of the optimization, where we ignores the accumulated errors caused by motion compensation and treat each picture as an intra one due to its simplicity. It has been shown that the memoryless algorithm does not much affect the quality and allows achieving essentially optimal (within 0.3 dB) performance [7].

Unlike FD, CD provides the ability to meet the available bandwidth quite accurately by adjusting the amount of dropped coefficients. We only drop AC DCT coefficients to avoid somewhat complicated syntax changes and to keep minimum necessary quality. We can define lots of CD operations by specifying the percentage of rate reduction to be achieved by coefficient dropping rather than directly specifying the dropped coefficients themselves. For example, the operation of CD (10%) represents the CD that reduces 10% reduction of the bit rate.

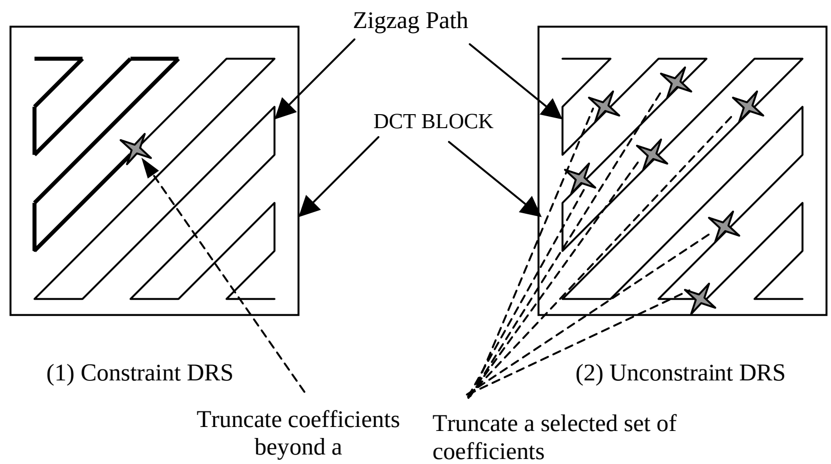


Figure 2: Constrained and unconstrained DRS

In the following part of this sub section, we will present two CD methods: Uniform Rate-based CD (URCD) and Lagrange Optimization CD (LOCD), respectively. URCD truncates the coefficients only based on the bit budget, without considering the imported distortion. However, the LOCD tries to find an optimal truncation point for each block within an optimization window of one frame. This is accomplished by using Lagrangian search to minimize the distortion caused by the CD. Please note no matter which CD method is utilized, the buffering of the whole frame is necessary for either the bit statistic purpose or the frame-based optimization search, or both. This is not a problem for MPEG-4 video stream due to its natural requirement of frame buffering for bi-directional motion compensation.

2.2.1 Uniform Rate-based CD (URCD)

URCD operates as follows: based on the target bit rate, a uniform ratio of bits, denoted as η_{bits}, will be truncated from each frame. Practically η_{bits} will be further converted into a more applicable *coefficient-dropping ratio* η_{coeff}. Then, the amount of truncated bits is uniformly distributed into each DCT block. The uniform distribution is simple but reasonable: it will yield even distortion to every frame as is desired. Specifically, suppose a video bit stream undergoes reshaping from rate R into R'. $R' < R$. And a given frame has B_f for the whole frame and B_c for the AC DCT coefficient bits only (only AC DCT coefficients are considered for two reasons: firstly it is recommended to keep DC coefficient from dropping to maintain baseline quality; secondly DC coefficients are sometimes coded separately from the VLC coding). We have following equations:

$$\eta_{bits} = 1 - \frac{R'}{R}$$
$$\eta_{coeff} = \frac{B_f \cdot \eta_{bits}}{B_c} \quad (1)$$

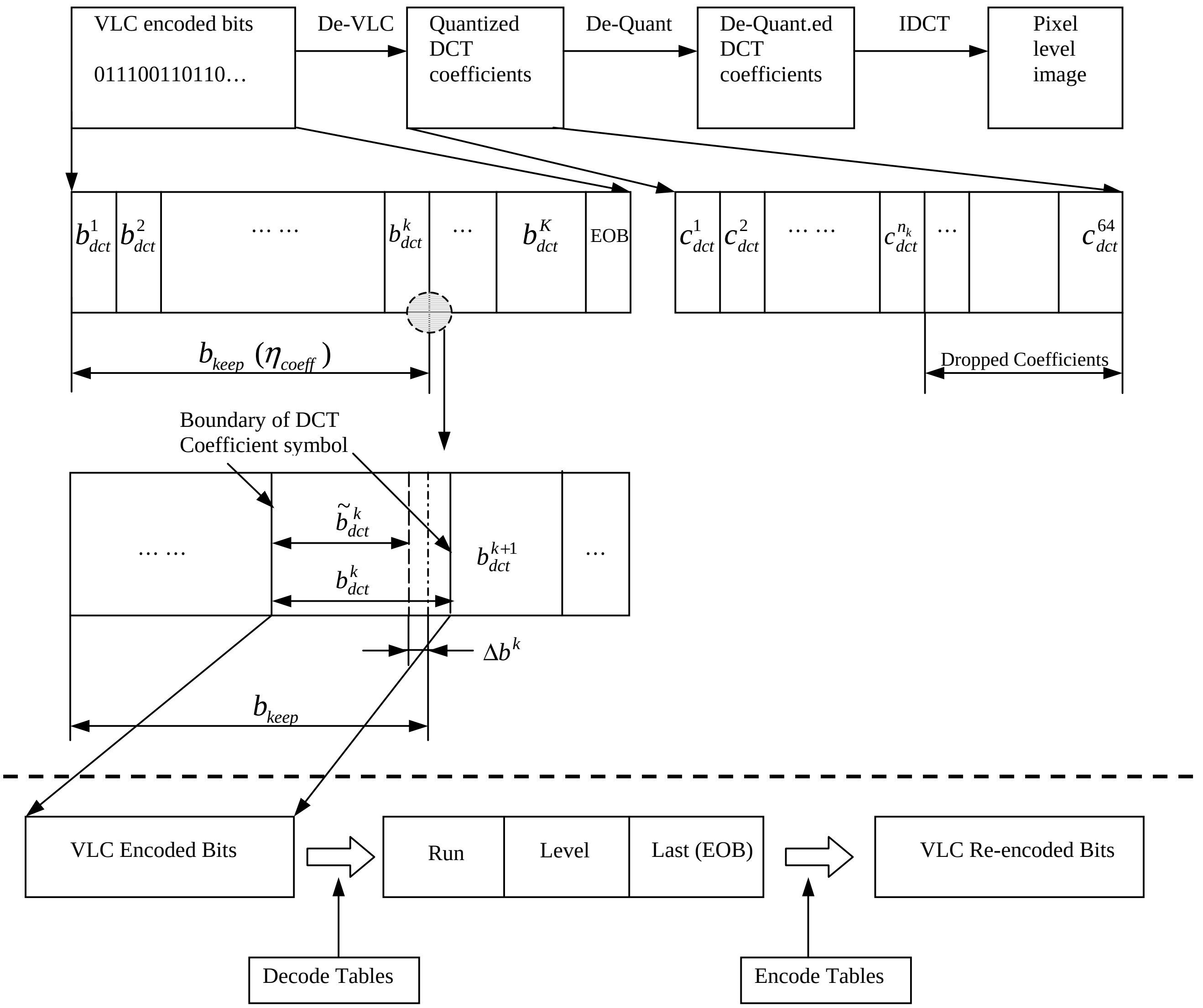


Figure 3: CD operation datagram

Figure 3 is a CD operation datagram for each block. For a DCT block with K non-zero coefficient symbols (a symbol here means the non-zero coefficient and its preceding zero runlength), use b_{dct}^i to denote the amount of bits for i^{th} symbol in the VLC coded symbol. $i = 1, 2, \ldots, K$. Algorithm 2.1 summarizes this process.

ALGORITHM 2.1: Uniform Rate-based Block CD

1. *calculate coefficient dropping ratio* η_{coeff} *by calculating equation* (1).

2. *calculate amount of coefficient bits kept* b_{keep} :

$$b_{keep} = (1-\eta_{coeff}) \cdot \sum_{i=1}^{K} b_{dct}^{i} \qquad (2)$$

3. $k = \arg\max_{k} \{\sum_{i=1}^{k} b_{dct}^{i} \le b_{keep}\}$

4. *re-encode the* k^{th} *coefficient symbol with EOB flagged. The bit amount will be changed from* b_{coeff}^{k} *to* $\tilde{b}_{coeff}^{k}$

5. *calculate the bit amount error* :

$$\Delta b_k = b_{keep} - \sum_{i=1}^{k-1} b_{coeff}^{i} - \tilde{b}_{coeff}^{k} \qquad (3)$$

The bit amount error Δb_k in step 5 comes from two facts: the coefficient dropping can only be implemented at the boundary of VLC coded symbol, which might not exactly match the required bit amount b_{keep}; the re-encoding in step 4 usually gives a different codeword based on the EOB-included 3D VLC [1]. Δb_k will be absorbed by next processed block. This is important to achieve a fine rate control and will be further discussed in section 3.

URCD truncate the original video bit stream with nearly no additional computational cost. The tradeoff is its decoded quality usually cannot be guaranteed because distortion impact is not considered. In LOCD, the Lagrange search is employed to optimize the truncation point for each block.

2.2.2 Lagrange Optimization CD (LOCD)

For a DCT block with K non-zero coefficient symbols, use b_{dct}^{i} to denote the amount of bits for i^{th} symbol. $i = 1, 2, \ldots, K$. If the coefficients after k^{th} symbol are dropped, the corresponding distortion b_{dct}^{i} can be denoted as:

$$D_k = \sum_{i=k+1}^{K} {x_i}^2 \qquad (4)$$

where x_i is the non-zero quantized DCT coefficient corresponding to the i^{th} symbol. Here a memoryless assumption is hold as is discussed in section 2.2. If the video comes with width *W* and height *H*, there are totally $N = W \cdot H \cdot 1.5/64$ blocks (For typical MPEG-4 planar YUV format, the sampling ratio is 4:2:0 for Y:U:V. I.e., every four blocks share one U and one V block). For each of these blocks, we can find a truncation point $k_j, j = 1,2,\ldots,N$. So the distortion for each frame can be denoted as:

$$D_k = \sum_{j=1}^{N} \sum_{i=k_j+1}^{K} x_i^{\,2} \qquad (5)$$

Thus, the optimization task is to find a set of truncation points $\vec{k} = (k_1, k_2, ..., k_N)$ in each frame, such that the distortion defined in equation (5) can be minimized under the bit rate constraint. For each frame our problem can be described as in table 1.

Table 1: Description of the CD optimization model

Known conditions:	Amount of bits in original frame B_f and the coefficient bits B_c Amount of bits $B_{keep} = B_c \cdot \eta_{coeff}$ that should be kept for DCT coefficients
Decision variables	Truncation points set $\vec{k} = (k_1, k_2, ..., k_N)$
Objective:	Minimize $D_k = \sum_{j=1}^{N} \sum_{i=k_j+1}^{K_j} x_i^{\,2}$
Constraints	$R_k = \sum_{j=1}^{N} \sum_{i=1}^{k_j} b_{coeff}^{ji} \le B_{keep}$

where b_{coeff}^{ji} is the amount of bits for the i^{th} symbol in the j^{th} block, and η_{coeff} is calculated by using equation (1).

Importing Lagrange multiplier λ, this constraint problem can be modeled as:

$$\text{minimize}\{D_k + \lambda R_k\} \text{ or minimize } \{\sum_{j=1}^{N}(\sum_{i=k_j+1}^{K_j} x_i^{\,2} + \lambda \sum_{i=1}^{k_j} b_{coeff}^{ji})\} \qquad (6)$$

Our goal is to find a suitable λ, such that when unconstraint problem (6) reach its optimum, we have

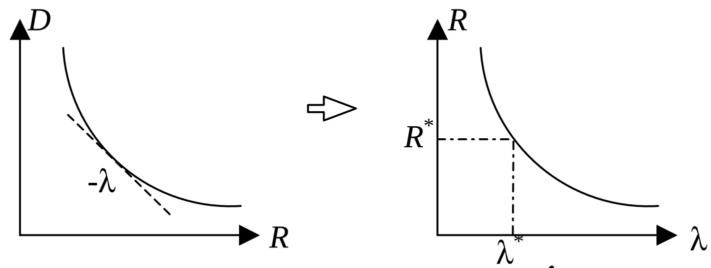
$$\left|B_{Keep} - R_k(\lambda)\right| \le \varepsilon, \varepsilon > 0 \qquad (7)$$

This is done based on an iteration approach: in each iteration, we use a Lagrange multiplier λ to find a corresponding optimum solution, i.e., the truncation points set $\vec{k}(\lambda) = (k_1, k_2, ..., k_N)_\lambda$. This is implemented using the exhausted search within a block since the maximum number of searches is $K \le 64$, which is not a heavy load. Once equation (7) is satisfied, the iteration stops. Otherwise, λ is adjusted afterwards and undergoes next iteration. the searching of suitable λ can be model as another unconstraint problem to minimize the objective function $\left|B_{Keep} - R_k(\lambda)\right|$. Since it is a one-dimension problem, we can use simple approach, such as bisection search, to resolve it.

Denote $f(k) = D_k(k) + \lambda R_k(k)$, we have

$$\nabla f(k) = \nabla D_k(k) + \lambda \nabla R_k(k) = 0 \;\Rightarrow\; \lambda = -\frac{\nabla D_k(k)}{\nabla R_k(k)} \qquad (8)$$

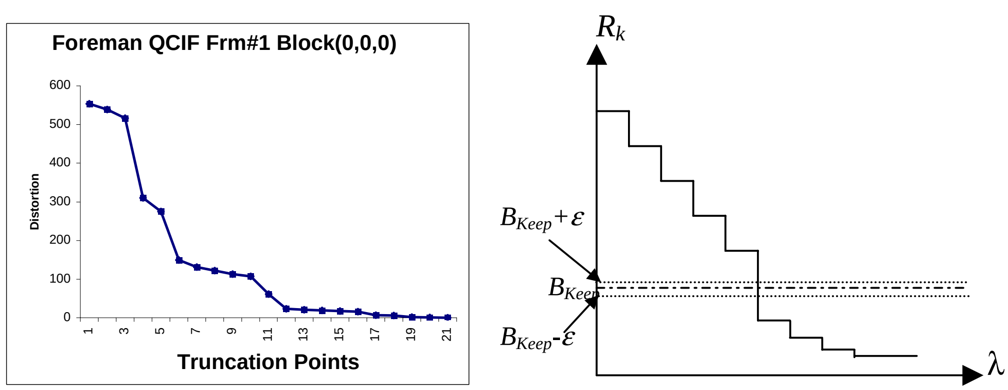
That is, at the optimum point λ is the slop of the R-D curve. Figure 4(1) shows an ideal R-D curve. Each point in the R-D curve stands for an optimum solution for the problem listed in table 1 given the slope λ. Since we are trying to find λ matching the target rate, the corresponding λ-R curve is shown in figure 4(2). Since it is monotonic, R and λ has a simple one-one mapping, we can definitely use bisection method to find the right solution.



(1) *R-D* (2) λ-*R*

Figure 4: Feasibility of Lagrange model

Figure 5(1) shows an actual R-D curve example from our experiment. It is taken from the video *Foreman* and for the first frame and first block. It is easy to see the local concave region from it. It is because of the discrete nature of CD. That is, the truncation of coefficient component can only be done discretely because each symbol has variable amount of bits. The consequence is some target rate might never be reachable as indicated in figure 5(2). Fortunately, according to our experiment result, this local concave is not a big problem, and we can use the rate adjustment among adjacent frames to handle the non-convergence issue in figure 5(2). Thus we can still employ the bisection algorithm yielding satisfied result.



(1) Actual R-D curve (2) Discrete Nature of CD

Figure 5: Actual R-D curve and discrete nature of CD

According to the description above, the Lagrange research is summarized in Algorithm 2.2.

ALGORITHM 2.2 Lagrange Optimization CD

$step\,0: initialization:$

$set\ \varepsilon > 0$

$set\ \omega > 0$

$set\ \lambda_{\min} = 0\ and\ \lambda_{\max} = M > 0$

$For\ \lambda_{\min}\ and\ \lambda_{\max},\ find\ optimum\ truncation\ set\ \vec{k}(\lambda_{\min})\ and\ \vec{k}(\lambda_{\max})\ optimizing\ problem\ in\ equation\,(6)$

$\vec{k}(\lambda) = (k_1, k_2, ..., k_N)_\lambda. Each\ k_i\ is\ got\ by\ exhausted\ search\ within\ the\ block.$

$Denote\ the\ corresponding\ rate\ as\ R_k(\lambda_{\min})\ and\ R_k(\lambda_{\max}).$

$Make\ sure\ R_k(\lambda_{\min}) < B_{Keep} < R_k(\lambda_{\max}), then\ goto\ step1$

$otherwise\ stop\ because\ there\ is\ no\ feasible\ solution;$

$step1: Iteration:$

$\lambda = (\lambda_{\min} + \lambda_{\max})/2,\ find\ optimum\ k(\lambda)\ and\ R_k(\lambda).$

$if\ \left|R_k(\lambda) - B_{Keep}\right| < \varepsilon\ then\ stop. the\ optimum\ solution\ is\ k(\lambda)$

$else\ if\ R_k(\lambda) - B_K > \varepsilon$

$\lambda_{\max} = \lambda$

$else\ if\ B_K - R_k(\lambda) > \varepsilon$

$\lambda_{\min} = \lambda$

$step\ 2: Decision:$

$if\ \left|\lambda_{\max} - \lambda_{\min}\right| < \omega\ stop. Current\ k(\lambda)\ is\ output\ and\ the\ rate\ adjustement\ will\ be\ applied\ in\ next\ frame$

$else\ goto\ step\ 1$

In [7], a MB based Lagrange optimization is adopted. I.e., all of the blocks within a macroblock (or even slice) will share the same truncation point. However, in our work the block based search is use. The difference between these two methods can be illustrated in figure 6, where four Y blocks within a macroblock are shown. It is clear to notice that by using block-based truncation, we can achieve an evener distortion and the bit nature of original block is considered. That is, the more bits a block has, the more bits will be dropped. So the improved quality is predictable.

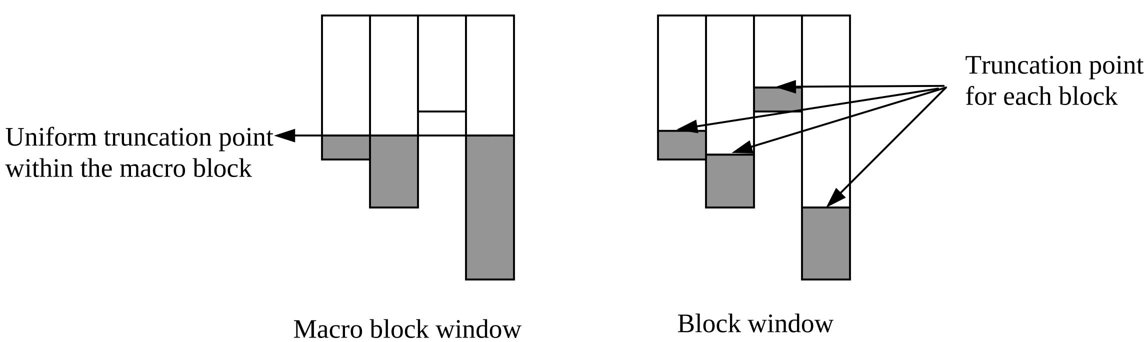


Figure 6: Macro block versus block based truncation

Also our experiment result demonstrates that the optimization on chrominance (U and V components) blocks with yield little gain in video quality. In order to reduce the computational cost, in our implementation, the optimization will be run only on the Y block. The truncation points in UV blocks will take the value of the mean from the four related Y blocks.

2.3 FD-CD combination

For higher bit rate reduction, CD alone is not sufficient to accommodate the available bandwidth. Moreover, only a set of coarse discrete values of bit rates can be achievable by FD along. Therefore, the FD-CD combining both enables us to extend the dynamic range of the reducible rate while providing finable video quality control. Moreover, the FD-CD probably yield better perceptual quality than either technique working alone, especially for large rate reductions, by trade-off between spatial and temporal quality.

In FD-CD, some frames are dropped and for remaining frame, CD will be utilized to get a fined bit rate, as is indicated in figure 7. The dramatic bits dropping by FD should be carefully considered during rate control, which will be discussed in next section.

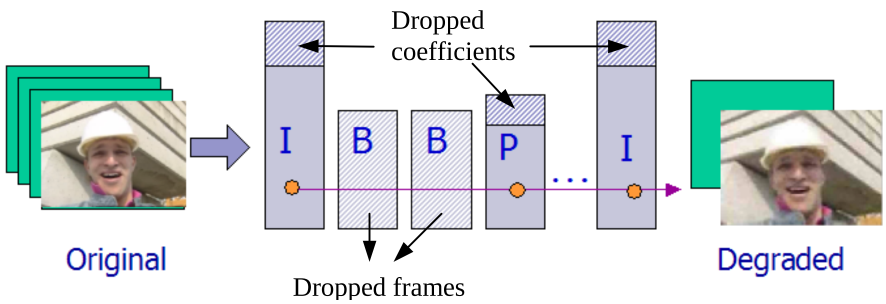


Figure 7: FD-CD combined transcoding

3. Rate Control

Rate control is important in current bandwidth-precious network. Though the issue of rate control in FD-CD transcoding is relatively simple compared with the rate control in encoding or re-quantization case, it still need careful design to reach a fined bit rate. In the FD-CD transcoding, the rate control requirement comes mainly from two aspects: bit budget adjustment among blocks and frames due to imperfect coefficient bit truncation, and the dramatic bit dropping due to FD. We will analyze them respectively and propose a unified rate control mechanism.

3.1 Handle imperfect coefficient bit truncation

Bit amount adjustment among blocks and frames is due to imperfect coefficient bit truncation, defined as bit drift in this report, for which four reasons are responsible:

(1) Bit truncation can only be applied on the coefficient symbol boundary. For a block with K coefficient symbols, if the truncation point is $k \le K$, the bit drift amount is:

$$\Delta b_{tr} = b_{keep} - \sum_{i=1}^{k} b_{coeff}^{i} \qquad (9)$$

where b_{keep} is the amount of coefficient bits after the truncation and defined in algorithm 2.1.

(2) Re-encoding the last symbol using 3-D VLC usually generates a new symbol with different bit amount. The bit drift amount is:

$$\Delta b_{re} = b^k_{coeff} - \tilde{b}^k_{coeff} \qquad (10)$$

where b^k_{coeff}, $\tilde{b}^k_{coeff}$ is the amount of coefficient bits for k^{th} symbol before and after the truncation respectively.

(3) The Lagrange optimization search sometimes fails to converge. The search of optimal Lagrange multiplier λ using equation (7) might not be converged, consequently importing a bit drift:

$$\Delta b_{la} = B_{Keep} - \tilde{R}_k(\lambda) \qquad (11)$$

where B_{Keep} is the target bit rate for the frame and $\tilde{R}_k(\lambda)$ is the actual reached rate considering both the convergence failure and the truncation bit drift described in (1) and (2).

(4) The padding bits at the end of each frame for alignment. This is somewhat trivial because only up to 8 bits are involved per frame [1]. In the meantime we will not discuss this in this report.

Among all of these reasons, Δb_{tr} and Δb_{la} are associated with the URCD algorithm. Please note $\Delta b_k = \Delta b_{tr} + \Delta b_{re}$, where Δb_k is the observed bit error after the truncation as defined in algorithm 2.1. The bit allocation adjustment for URCD is simple: Δb_k will be merged into the next block. When the end of the frame is reached, Δb_k will be passed into the first processed block in next frame. E.g., the bit budget b_{keep} of next block will be adjusted as:

$$\tilde{b}_{keep} = b_{keep} - \Delta b_k \qquad (12)$$

For LOCD, Δb_{la} is the observed bit drift for each frame. Since the bit budget allocation is optimized based on a frame-size window, the block-by-block adjustment above employed in URCD is not suitable. Instead we propose to use the VLC re-encoded symbol to calculate the distortion optimization. I.e., the equation (6) will be rewritten as:

$$\text{minimize}\{D_k + \lambda R_k\} \text{ or minimize } \{\sum_{j=1}^{N}(\sum_{i=k_j+1}^{K_j} x_i^{\,2} + \lambda(\sum_{i=1}^{k_j-1} b^{ji}_{coeff} + \tilde{b}^{jk_j}_{coeff}))\} \qquad (13)$$

where $\tilde{b}^{jk_j}_{coeff}$ is the VLC re-encoded symbol with EOB flagged. In this report, this method is refered as VLC Re-encoding Drift Compensation (VRDC).

Further more, with absence of FD, the observed drift in LOCD Δb_{la} will be considered in next frame using a similar way in equation (12), with the difference that a frame bit budget B_{Keep} is considered.

3.2 Handle FD

The challenge of handling FD is how to allocate bit budget for remaining frames. In CD without FD, the bit budget is distributed into each frame using a uniform truncation ratio. In FD, however, all of the bits of a frame are dropped and the uniform distribution is destroyed. In order to keep a uniform quality among the kept frames, we adopt the GOP windowed rate control scheme well known in video encoding rate control [11]. FD here is somewhat like the frame skip case during encoding in order to avoid the buffer overflow. We proposed a rate control method called Adaptive Frame Bit Allocation (AFBA) to handle the FD-CD combined transcoding. The key technique is to adjust the bit allocation for each processed frame dynamically along the transcoding in a GOP. In the remaining of this section we will detail this method using the typical MPEG-4 GOP structure, i.e., GOP size N=15 and sub-GOP size M=3. AFBA can be applied to other GOP structure in a similar way. The following definition is used during the illustration:

$FPS: frame\ rate$

$N: GOP\ size$

n_I : # of remaining I frames in a GOP without considering FD

n_B : # of remaining B frames in a GOP without considering FD

n_P : # of remaining P frames in a GOP without considering FD

n'_I : # of remaining I frames in a GOP considering FD

n'_B : # of remaining B frames in a GOP considering FD

n'_P : # of remaining P frames in a GOP considering FD

w_I : Estimated weight for I frame in bit allocation.

w_P : Estimated weight for P frmae in bit allocation

w_B : Estimated weight for B frame in bit allocation.

r_U : Estimated frame original bit amount unit

r'_U : Estimated frame target bit amount unit

R : Original bit rate

R': Target bit rate

B_f : Frame total bit amount

B_c : Amount of bit amount for coefficient symbols in a frame

Firstly, the bit budget for a GOP is allocated using the following calculation:

$$\begin{aligned} &Original\ bits\ for\ GOP: R_{GOP} = \frac{R \cdot N}{FPS} \\ &Allocate\ target\ bits\ rate\ for\ GOP: R'_{GOP} = \frac{R' \cdot N}{FPS} \end{aligned} \qquad (14)$$

Secondly, before transcoding, the estimated bit amount unit is worked out. In order to do this, we adopt w_I, w_P and w_B. So,

$$\begin{aligned} r_U &= \frac{R_{GOP}}{w_I \cdot n_I + w_P \cdot n_P + w_B \cdot n_B} \\ r'_U &= \frac{R'_{GOP}}{w_I \cdot n'_I + w_P \cdot n'_P + w_B \cdot n'_B} \end{aligned} \qquad (15)$$

w_I , w_P and w_B reflect the bit impact from I, P and B frame. Empirically, the initialization values for them are set to be 4.0, 2.0 and 1.0 respectively. The precise initialization is not crucial because for different video stream they might vary, and during the transcoding, these weights will be adjusted according to the collected latest statistic, as will be introduced soon. This adaptation makes sense based on the assumption that similar video contents will perform similar coding behavior. n_I, n_P, n_B are decided by the GOP structure, while n'_I, n'_P, n'_B are decided by both the GOP structure and the FD mode. These six numbers will be adjusted after transcoding every frame. For a better understanding, figure 8 instantiate the relationship among these parameters, where a FD mode is dropping the first B frame in each sub-GOP. The frames are aligned in the actual coding order.

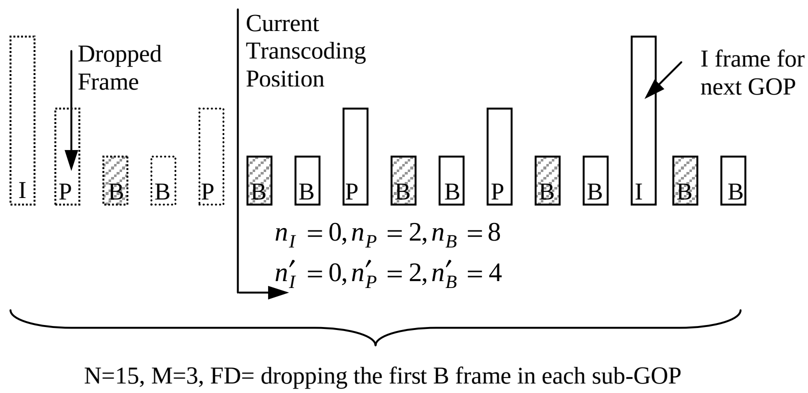


Figure 8: Parameters involved in FD-CD rate control

w_I and will be adjusted during the transcoding.

In summary, the algorithm is summarized in Algorithm 3.

ALGORITHM 3 Rate Control In FD-CD Transcoding

For each GOP :

step 0 : initialization :

Original bits for GOP : $R_{GOP} = \frac{R \cdot N}{FPS}$

Allocate target bits rate for GOP : $R'_{GOP} = \frac{R' \cdot N}{FPS}$

Calculate n_I, n_P, n_B *after FD*

Estimate truncation ratio $\tilde{\eta}_{bits} = \frac{R'_{GOP} \ / \ (w_I \cdot n_I + w_B \cdot n_B + w_P \cdot n_P)}{R_{GOP} \ / \ (w_I \cdot n_I + w_B \cdot n_B + w_P \cdot n_P)}$

step 1 : FD – CD

while $n > 0$

if current frame is not dropped

buffer current frame and get its bit statistic B_f *and* B_c

if using URCD

run CD using algorithm 2.1, generating target frame with bit amount $\tilde{R}_k$

elseif using LOCD

run CD using algorithm 2.2, generating target frame with bit amount $\tilde{R}_k$

4. Experiment Results

5. Conclusion

Reference:

[1] ISO/IEC 14496-2, "Coding of audio-visual objects: Visual".

[2] Huifang Sun; Kwok, W.; Zdepski, J.W., "Architectures for MPEG compressed bitstream scaling". Page(s): 191-199. Circuits and Systems for Video Technology, IEEE Transactions on. Volume: 6, Issue: 2, April 1996

[3] Sorial, H.; Lynch, W.E.; Vincent, A.; "Selective requantization for transcoding of MPEG compressed video". ICME 2000. 2000 IEEE International Conference on. Volume: 1, 2000. Page(s): 217 -220 vol.1

[4] Kwang-Deok Seo; Sang-Hee Lee; Jae-Kyoon Kim; Jong-Seog Koh. "Rate control algorithm for fast bit-rate conversion transcoding". Consumer Electronics, IEEE Transactions on, Volume: 46 Issue: 4, Nov 2000. Page(s): 1128 –1136

[5] Werner, O.” Requantization for transcoding of MPEG-2 intraframes”. Image Processing, IEEE Transactions on, Volume: 8 Issue: 2, Feb 1999. Page(s): 179 –191

[6] Nakajima, Y.; Hori, H.; Kanoh, T. “Rate conversion of MPEG coded video by re-quantization process”. Image Processing, 1995. Proceedings. International Conference on , Volume: 3 , 23-26 Oct 1995. Page(s): 408 -411 vol.3

[7] A. Eleftheriadis, *Dynamic Rate Shaping of Compressed Digital Video*, Doctoral Dissertation, Graduate School of Arts and Sciences, Columbia University, June 1995.

[8] Kwang-deok Seo; Jae-kyoon Kim. “Motion vector refinement for video downsampling in the DCT domain”. IEEE Signal Processing Letters, Volume: 9 Issue: 11, Nov 2002. Page(s): 356 –359

[9] Y. Liang and Y.-P. Tan, “A new content-based hybrid video transcoding method,” Proc. ICIP’01, October 2001, pp. 429-432

[10] T. Shanableh and M. Ghanbari, “Heterogeneous video transcoding to lower spatio-temporal resolutions and different encoding formats,” *IEEE Trans. Multimedia*, vol. 2, pp. 101-110, June 2000.

[11] Hung-Ju Lee, Tihao Chiang, Ya-Qin Zhang. “Scalable Rate Control for Very Low Bit Rate (VLBR) Video”. Proceeding of ICIP’97 3-Volume Set-Volume 2. October 26 - 29, 1997 Washington, DC