Proceedings, 2nd IEEE International Conference on Image Processing (ICIP-95), Arlington, Virginia, October 1995.

CONSTRAINED AND GENERAL DYNAMIC RATE SHAPING OF COMPRESSED DIGITAL VIDEO

Alexandros Eleftheriadis and Dimitris Anastassiou

Department of Electrical Engineering
and Center for Telecommunications Research
Columbia University, New York, NY 10027, USA
{eleft,anastas}@ctr.columbia.edu

ABSTRACT

We introduce the concept of *Dynamic Rate Shaping*, a technique to adapt the rate of compressed video bitstreams (MPEG-1, MPEG-2, H.261, as well as JPEG) to dynamically varying bit rate constraints. The approach provides an interface (or filter) between the encoder and the network, with which the encoder's output can be perfectly matched to the network's quality of service characteristics. Since the presented algorithms do not require interaction with the encoder, they are fully applicable to precoded, stored video (e.g., video-on-demand systems). By decoupling the encoder and the network, universal interoperability can be achieved. In essense, DRS bridges the gap between CBR and VBR video, allowing a contiuum of possibilities between the two. A set of low-complexity algorithms for so-called unconstrained dynamic rate shaping are presented, and both optimal and extremely fast designs are discussed. Experimental results are provided using actual MPEG-2 bitstreams.

1. INTRODUCTION

We introduce the concept of *Dynamic Rate Shaping* (DRS), a technique to adapt the rate of compressed video bitstreams (MPEG-1, MPEG-2, H.261, as well as JPEG) to dynamically varying bit rate constraints. The approach has several applications. It provides an interface (or bitstream filter) between an encoder and a network, with which the encoder's output can be perfectly matched to the network's quality of service characteristics. It can match the rate capabilities of encoders with a variety of decoders. It can facilititate multipoint communication with mobile hosts, without compromising the signal quality of wired participants. It can provide smoother trick-mode (fast forward, fast reverse) operation in digital video recorders, and so on.

Since the presented algorithms do not require interaction with the encoder, they are fully applicable to precoded, stored video (as in, for example, video-on-demand systems). By providing decoupling of the encoder and channel or decoder in terms of rate, universal interoperability can be achieved. In essense, DRS bridges the gap between constant and variable bit rate video, allowing a contiuum of possibilities between the two.

Although techniques have been developed to employ rate control for live sources based on network feedback [5, 7], no general solution is currently available for prerecorded material. Also, the approaches used in [9] and [11] to manipulate the rate of compressed H.261 streams are ad-hoc, with no characterization of the performance of the proposed schemes (in fact, in both cases considerable error drift is introduced due to the—ignored—non-linearity of motion compensation).

A family of DRS algorithms is presented for two cases of DRS, namely *constrained* and *general* or *unconstrained*. Both optimal and extremely fast designs are discussed; the latter are very close to optimal (within 0.5 dB) and are simple enough to allow even software-based implementation. Experimental results are provided using actual MPEG-2 bitstreams. Familiarity with MPEG techniques and terminology is assumed; detailed descriptions can be found in [1, 8].

2. DYNAMIC RATE SHAPING

We define rate shaping as an operation which, given an input compressed video bitstream and a set of rate constraints, produces another compressed video bitstream that complies with these constraints. For our purposes, both bitstreams are assumed to meet the same syntax specification, and we also assume that a—possibly motion-compensated—block-based transform coding scheme is used. This includes both MPEG-1 and MPEG-2, as well as H.261 and so-called "motion" JPEG. If the rate constraints are allowed to vary with time, the operation will be called *dynamic* rate shaping.

The rate shaping operation is depicted in Fig. 1. Note that no communication path exists between the rate shaper and the source of the input bitstream, which ensures that no access to the encoder is necessary. Of particular interest is the source of the rate constraints $B_T(t)$. In the simplest of cases, $B_T(t)$ may be just a constant and known a priori (e.g. the bandwidth of a circuit-switched connection). It is also possible that $B_T(t)$ has a well (a priori) known statistical characterization (e.g. a policing function). Finally, another alternative is that $B_T(t)$ is generated by the network over which the output bitstream is transmitted; this could be potentially provided by the network manage-

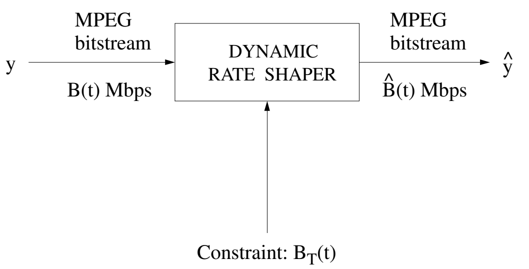

Figure 1: Operation of a dynamic rate shaper.

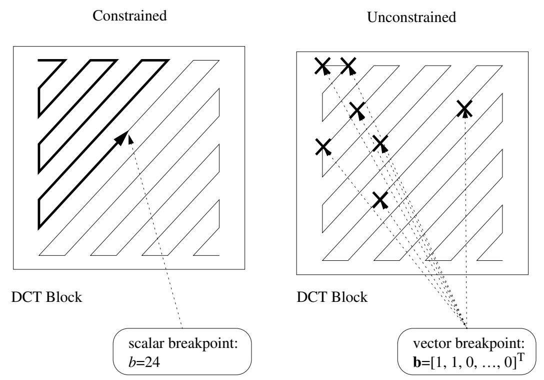

Figure 2: Breakpoint definition for constrained and unconstrained DRS.

ment layer, or may be the result of end-to-end bandwidth availability estimates (as in [5, 7]). The objective of a rate shaping algorithm is to minimize the conversion distortion, i.e.:

$$\min_{\hat{B}(t)\leq B_T(t)} \{\|y-\hat{y}\|\} \tag{1}$$

where $\|\cdot\|$ denotes the squared error criterion.

Note that no assumption is made on the rate properties of the input bitstream, which can indeed by arbitrary. The attainable rate variation ($\hat{B}/B$) is in practice limited, and depends primarily on the number of B pictures and the original rate of the bitstream. Also, no indication is given on the length of the optimization window, which can be arbitrary. Complexity and delay considerations make it desirable that it is kept small, and our interests will focus in the case where the window is up to a complete picture (frame or field).

Assuming that a motion-compensated block-based transform coding technique is used to generate the input bitstream and decode the output one, there are two fundamental ways to reduce the rate: 1) modifying the quantized transform coefficients by employing coarser quantization, and 2) eliminating transform coefficients. In general, both schemes can be used to perform rate shaping; requantization, however, leads to recoding-like algorithms which are not amenable to very fast implementation and, as we have shown [4, 2], do not perform as well as selective-transmission ones. Note that full recoding represents the brute-force approach in effecting rate changes, and falls in this category. In the rest of this paper we only consider selective-transmission based algorithms. In particular we examine two different cases: truncation, and arbitrary selection. In the former case, a set of DCT coefficients at the end of each block is eliminated. This approach will be referred to as *constrained* DRS. In the latter case, our algorithm is allowed to arbitrarily select DCT coefficients for elimination from the bitstream, and hence will be called *general* or *unconstrained* DRS.

The constrained DRS problem is similar to that of optimal data partitioning of MPEG-2 video [5]; an analysis in the context of DRS is provided in [4]. The general DRS problem, on the other hand, is similar to optimal thresholding [6]. In constrained DRS, the number of DCT run-length codes within each block which will be kept is called the *breakpoint*, paralleling the data partitioning terminology [3]. All DCT coefficients above the breakpoint are eliminated from the bitstream. In general DRS, the breakpoint becomes a 64-element binary vector, indicating which coefficients within each 8×8 block will be kept. Fig. 2 illustrates the difference between the two approaches.

Assuming use of MPEG, and to avoid certain syntax complications[1], we require that at least one DCT coefficient remains in each block. Consequently, scalar breakpoint values range from 1 to 64, while vector ones will be non-zero. In the following, we concentrate on general DRS and its performance, particularly in comparison to constrained DRS.

3. GENERAL DYNAMIC RATE SHAPING

Let $\mathbf{b}_i = [b_i^k \in \{0,1\}, k=1,\ldots,K]^T$ denote the breakpoint vector for block i, where b_i^k is 1 iff the k-th DCT coefficient run-length of block i-th is retained. In order to avoid a perceptually ill-defined distortion, we only include the luminance component in our distortion calculations. As a result, breakpoint decisions will be made on a per macroblock bases, i.e. all blocks of a given macroblock will use the same breakpoint vector. Rate calculations include, of course, the two chrominance components. It can then be shown that the optimal DRS problem for intra-coded pictures can be formulated as:

$$\min_{\sum_{i=1}^{N} R_i(\mathbf{b}_i)\leq B_T} \left\{\sum_{i=1}^{N} D_i(\mathbf{b}_i)\right\} \tag{2}$$

with

$$D_i(\mathbf{b}_i) = \sum_{j\in\mathcal{Y}}\sum_{k} \overline{b_i^k} E_j^i(k)^2 \tag{3}$$

where N is the number of macroblocks over which optimization takes place, $\mathcal{Y}$ denotes the luminance blocks of a macroblock, $R_i(\cdot)$ denotes the number of bits required to code macroblock i with the breakpoint configuration $\mathbf{b}_i$, $D_i(\mathbf{b}_i)$ denotes the distortion associated with $\mathbf{b}_i$, $E_j^i(k)$ is

[1] These include recoding the coded block patterns, and reexecuting DC prediction loops.

the value of the DCT coefficient of the k-th run in the j-th block of the i-th macroblock, and $\overline{b_i^k}$ is the logical inverse of b_i^k.

Due to the temporally recursive structure of motion compensated predictive coding, any modification in the bitstream will propagate to future P and B pictures. Consequently, an optimal solution for (1) would have to take into account a complete group of pictures (I to I). Since the delay in doing so would be unacceptable, we consider the "causally" optimal approach in which the algorithm takes into account only the error accumulated from past pictures, and not the one that will be propagated to future ones. Note that the latter error will be considered when the algorithm performs rate shaping decisions for these pictures. The decoding process without rate shaping can be described by the equation:

$$y_i = \mathcal{M}_i(y_{i-1}) + e_i \qquad e_0 = y_0 \tag{4}$$

where y_i is the i-th decoded picture, $\mathcal{M}(\cdot)$ is the motion compensation operator, and e_i denotes the coded prediction error. Only one reference picture is shown here for simplicity; generalization to include multiple pictures (for B pictures) is trivial. When rate shaping is applied, we have:

$$\hat{y}_i = \mathcal{M}_i(\hat{y}_{i-1}) + \hat{e}_i \qquad \hat{e}_0 = \hat{y}_0 \tag{5}$$

Hence Eq. (1) becomes:

$$\min_{\hat{B}(t) \le B_T(t)} \{\|a_i + e_i - \hat{e}_i\|\} \tag{6}$$

where $a_i = \mathcal{M}_i(y_{i-1}) - \mathcal{M}_i(\hat{y}_{i-1})$ is the accumulated rate shaping error up to the current picture.

Using the causality argument and taking into account only the accumulated error, it can be shown that Eq. (2) can be generalized to include P and B pictures by defining the distortion as follows:

$$\begin{aligned} D_i(\mathbf{b}_i) \;=\; & \sum_{j\in\mathcal{Y}} \Big\{ \sum\nolimits_k A_j^i(k)^2 + \sum\nolimits_k 2\overline{b_i^k} A_j^i(\mathcal{I}_j^i(k)) E_j^i(k) + \\ & \sum\nolimits_k \overline{b_i^k} E_j^i(k)^2 \Big\} \end{aligned} \tag{7}$$

where $A_j^i(k)$ is the k-th DCT coefficient (in zig-zag scan order) of the i-th block of the j-th macroblock of accumulated error, and $\mathcal{I}_j^i(\cdot)$ maps run-length positions to zig-zag scan positions.

The constrained minimization problem of (2) and (7) can be converted to an unconstrained one using Lagrange multipliers [3, 6, 10]: instead of minimizing D given R, we minimize $L = D + \lambda R$. These problems are not equivalent; for some value of λ, however, which our algorithm will have to find, their solutions become identical. A fast, iterative algorithm for the determination of the optimal λ is provided by bisection [3, 6]. Briefly, the bisection algorithm starts with two extreme values for λ, and iteratively decreases their distance until convergence occurs (typically within 10–12 iterations). A key characteristic of this algorithm is that it operates in the convex hull of the "operational" $R(D)$ curves of each block.

Fig. 3 shows an $R(D)$ curve (more precisely, "cloud") from the "Mobile" sequence, where 12 DCT coefficients are present. The plot includes $2^{12} = 4,096$ points, one per combination of DCT coefficients. Observe that the plot consists of two identical and spatially displaced clouds, due to the presense of a dominant (DC) DCT coefficient. Also note that the convex hull is comprised of a small number of points (the upper cloud is excluded in its entirety). As a result, the convergence of the bisection algorithm is not significantly affected compared with the constrained DRS approach [3, 4].

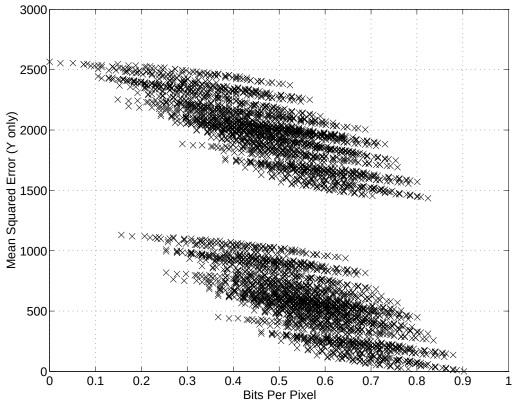

Figure 3: $R(D)$ "clouds" from a macroblock of "Mobile", coded at 4 Mbps. The 12 DCT coefficients generate 4,096 different breakpoint vector possibilities.

Within each iteration of the bisection algorithm, we have to determine the optimum breakpoint vector configuration $\mathbf{b}_i$ for each block (note that this can be performed independently for each block). This is not as simple as in data partitioning [3] or constrained rate shaping [4] due to the interdependence of run-length codes within each block. A fast solution can, however, be obtained using a slightly modified version of the optimal thresholding dynamic programming algorithm presented in [6]. A brief description of the algorithm is as follows; a more detailed description can be found in [2].

Let us consider a single macroblock and a given λ. For notational economy, we drop the macroblock index in the sequel. The algorithm starts from the DCT coefficient of the first run-length and, moving towards the end, examines the benefit of including each run-length. At initialization (step 0), we then have an all-zero breakpoint configuration and an optimal (at step 0) Lagrangian distortion L_0^* which equals the maximum:

$$L_0^* = \sum_{j\in\mathcal{Y}} \sum_{k=1}^{N-1} \left\{A^j(k) + E^j(k)\right\}^2 \tag{8}$$

In succeeding steps we consider the incremental cost reduction ΔL_{ij} of going from run-length n directly to m, skipping those in between:

$$\Delta L_{nm} = \sum_{j\in\mathcal{Y}} \left\{ -A^j(\mathcal{I}^j(m)) E^j(m) + E^j(m)^2 + \lambda R_j(n,m) \right\} \tag{9}$$

where $R_j(n,m)$ is the number of bits needed to encode the run-length code of he DCT coefficient of the m-th run-length code when the run begins at the position of run-length code n. These values can be precomputed at the

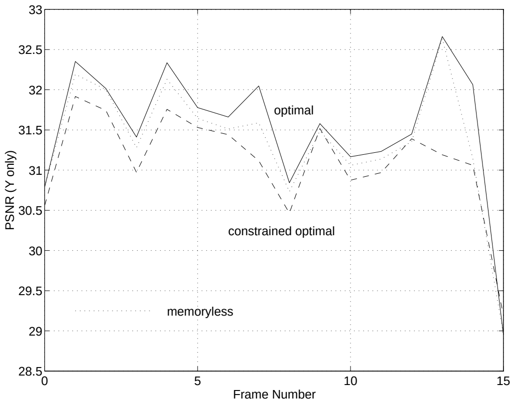

Figure 4: "Mobile" sequence, coded using MPEG-2 MP@ML at 4 Mbps and rate-shaped at 3.2 Mbps, using GDRS, CDRS, and memoryless GDRS algorithms.

beginning of the bisection algorithm, and thus be reused for all different values of λ.

At each step k, the algorithm considers the minimum Lagrangian cost associated with having k as the last run-length. A set of candidate optimal predecessors S_k at step k is maintained. In a full-search approach, this set would always contain all the preceeding DCT run-lengths; what allows for a fast algorithm is the fact that this set can be very effectively pruned [6, 2]. This is a result of the monotonicity property of the Huffman run-length codes used in MPEG and JPEG: for any given DCT coefficient level, longer zero run-lengths correspond to codes that have non-decreasing lengths. Thus if we consider a predecessor l to k with equal or higher optimal Lagrangian cost from k, the cost of going to a future run-length from either l or k will always be less than or equal if we do so from k (since the run-length encoding for the longer path will take at least as many bits).

4. EXPERIMENTAL RESULTS

Experimental results with the "Mobile" sequence coded (using MPEG-2) at 4 Mbps and rate-shaped at 3.2 Mbps are shown in Fig. 4. PSNR values are with respect to the actual decoded signal. The algorithm performed optimization on a picture-basis, and $R(D)$ values were collected on a macroblock basis. We observe that general DRS (GDRS) outperforms constrained DRS (CDRS) by only 0.5 dB. This appears to be due to the fact that the zig-zag scan is a particularly effective ordering of DCT coefficients. In other words, the GDRS most of the time selects breakpoint vectors which essentially immitate the truncation operation of CDRS. Also shown in Fig. 4 are results of memoryless GDRS, in which the accumulated error is totally ignored. As in CDRS [4, 5] it is verified that the memoryless algorithm performs almost identically to the optimal one.

This is a very important result, as it implies that we can dispense with error accumulation tracking, use a much simpler design, and achieve results very close to optimal. Note that while the GDRS algorithm has complexity between than of a decoder and an encoder, the memoryless GDRS algorithm is less complex than a decoder. In addition, the memoryless CDRS algorithm is even simpler (since the dynamic programming step is avoided), yet it performs almost identically to optimal CDRS. We conclude that the memoryless CDRS algorithm provides an excellent performance-complexity tradeoff. Early experiments have shown that even real-time software-based implementations of memoryless CDRS should be possible using clustering (joint selection of breakpoint value for a small cluster of macroblocks—e.g. 4 MBs). This makes the approach very attractive, since it essentially eliminates implementation and deployment costs.

5. REFERENCES

[1] Information Technology – Generic Coding of Moving Pictures and Associated Audio, ITU-T Draft Recommendation H.262, ISO/IEC 13818 Draft International Standard (MPEG-2). 1994.

[2] A. Eleftheriadis. *Dynamic Rate Shaping of Compressed Digital Video*. PhD thesis, Columbia University, New York, New York, 1995.

[3] A. Eleftheriadis and D. Anastassiou. Optimal Data Partitioning of MPEG-2 Coded Video. In *Proceedings, 1st IEEE International Conference on Image Processing*, pages I.273–I.277, Austin, Texas, November 1994.

[4] A. Eleftheriadis and D. Anastassiou. Meeting Arbitrary QoS Constraints Using Dynamic Rate Shaping of Coded Digital Video. In *Proceedings, 5th International Workshop on Network and Operating System Support for Digital Audio and Video*, pages 95–106, Durham, New Hampshire, April 1995.

[5] A. Eleftheriadis, S. Pejhan, and D. Anastassiou. Architecture and Algorithms of the Xphone Multimedia Communication System. *ACM/Springer Verlag Multimedia Systems Journal*, 2(2):89–100, August 1994.

[6] K. Ramchandran and M. Vetterli. Rate-distortion optimal fast thresholding with complete JPEG/MPEG decoder compatibility. *IEEE Transactions on Image Processing, Special Issue on Video Sequence Compression*, 3(5):700–704, September 1994.

[7] H. Kanakia, P. P. Mishra, and A. Reibman. An Adaptive Congestion Control Scheme for Real-Time Packet Video Transport. In *Proceedings, ACM SIGCOMM '94 Conference*, pages 20–31, September 1993.

[8] Didier LeGall. MPEG: A Video Compression Standard for Multimedia Applications. *Communications of the ACM*, 34(4):46–58, April 1991.

[9] D. G. Morrison, M. E. Nilsson, and M. Ghanbari. Reduction of the Bit-Rate of Compressed Video While in its Coded Form. In *Proceedings, Packet Video Workshop '94*, pages D17.1–D17.4, 1994.

[10] Y. Shoham and A. Gersho. Efficient bit allocation for an arbitrary set of quantizers. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 36(9):1445–1453, 1988.

[11] M. Yong, Q.-F. Zhu, and V. Eyuboglu. VBR Transport of CBR-Encoded Video over ATM Networks. In *Proceedings, Packet Video Workshop '94*, pages D18.1–D18.4, 1994.