OPTIMAL BUFFER-CONSTRAINED SOURCE QUANTIZATION AND FAST APPROXIMATIONS

Antonio Ortega[*] *Kannan Ramchandran*[†] *Martin Vetterli*[‡]

Department of Electrical Engineering
and Center for Telecommunications Research,
Columbia University,
New York, N.Y. 10027

ABSTRACT

We formalize the description of the buffer-constrained quantization problem. For a given set of admissible quantizers to code a discrete non-stationary signal sequence in a buffer-constrained environment, and any global distortion minimization criterion which is additive over the individual elements of the sequence, we formulate the optimal solution as well as slightly suboptimal but much faster approximations. As a first step, we define the problem as one of constrained, discrete optimization and establish its equivalence to some of the problems studied in the field of integer programming. Dynamic programming using the Viterbi algorithm is shown to provide a way of computing the optimal solution. Finally, we provide a heuristic algorithm based on Lagrangian optimization using an operational rate-distortion framework that, with much-reduced computing complexity, approaches the optimally achievable SNR.

1. INTRODUCTION

The application of variable bit rate (VBR) techniques to non-stationary sources, such as video sequences, in a constant bit rate (CBR) transmission environment, requires the definition of buffer control policies. Recently, the need to address the problem of buffer-constrained quantization in the context of image and video coding has risen sharply to the point where buffer-control algorithms are being proposed for the MPEG video standard [1]. Many applications like CD-ROM storage of images and video sequences, windows applications for workstations, buffer-limited JPEG [2] coding, and MPEG buffer control strategies are *non real-time* finite-buffer constrained coding applications where computationally expensive methods are not taboo if the complexity-performance tradeoff is worthwhile, specially if the one-time coding complexity can reduce transmission cost, e.g. limiting the amount of buffer memory needed by the user.

This provides the motivation to investigate optimal quantization strategies for the coding of signal sequences in a finite buffer environment, and to quantify the performance tradeoffs involving key design parameters like buffer size or buffer occupancy "granularity". The possession of an optimal solution can also be an invaluable benchmark for assessing the performance of real-time constrained and practical coders as well as for quantifying the suboptimality of fast heuristics. In asynchronous network applications (Asynchronous Transfer Mode or ATM networks), the idea of "self-policing" by the user, to guarantee conformance with the negotiated transmission parameters while ensuring an optimal grade of quality delivered, can also be very appealing, as the user has more "control" over the quality of service he can expect from the network. Besides, negating the value of buffer-control algorithms by making the encoder resort to a large enough buffer size to absorb all source bitrate variations may not only be unacceptable because of end-to-end delay restrictions, but also economically unwise even when delay is not an issue, as there may exist "smarter" shorter-buffer solutions that yield the same performance.

We formalize the generalized problem of buffer-constrained independent quantization of a sequence and describe how, given a set of quantizers, a finite buffer, and any additive cost measure over the sequence elements, an optimal solution can be found [3]. We show how this problem, one of discrete optimization with constraints, can be construed as a deterministic dynamic programming problem with the Viterbi algorithm used to compute the optimal solution. After drawing parallels between this buffer-constrained quantization problem and the less complex budget-constrained unbuffered quantization problem [4], we present a recursive Lagrange-multiplier based algorithm that provides a fast nearly-optimal solution with much reduced complexity. For simplicity, we use the mean-squared-error (MSE) distortion criterion in our simulations, though any additive criterion is admissible in general. The source sequence elements (8x8 pixels image blocks in our simulation) are quantized with the JPEG coder [2] using a finite number of quantization scales as the admissible quantization set.

2. PROBLEM DEFINITION

2.1. A First Formulation

Let us consider a sequence made of blocks, representing discrete analog samples or sets of samples (depending on the application), that are to be coded independently, possibly after some unitary transformation. For a given finite set of quantizers, the problem consists in choosing, for each block,

[*] Work supported in part by the Fulbright Commission and the Ministry of Education of Spain

[†] Work supported in part by the New York State Science and Technology Foundation's CAT at Columbia University

[‡] Work supported in part by the National Science Foundation under grants ECD-88-11111 and MIP-90-14189.

0-7803-0593-0/92 $3.00 1992 IEEE

that quantizer within the admissible set, that will minimize the *global* cost of coding the sequence, where the cost is additive over the independent individual blocks.

Our system consists of three basic elements: the encoder, the decoder (each including a buffer, See Figure 1) and the transmission channel. In the general case, although transmission need not be synchronous (e.g. video transmission over ATM networks), it can be seen that, since the encoder and the decoder are usually attached to synchronous devices, a constant delay restriction exists between the input to the coder and the output of the decoder. See Figure 2. As a consequence, given the constant delay through the system and the finite channel rate, we conclude that the buffer size will be finite.

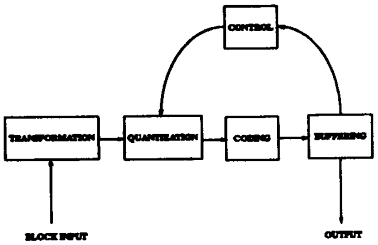

Figure 1: Block diagram of the encoder

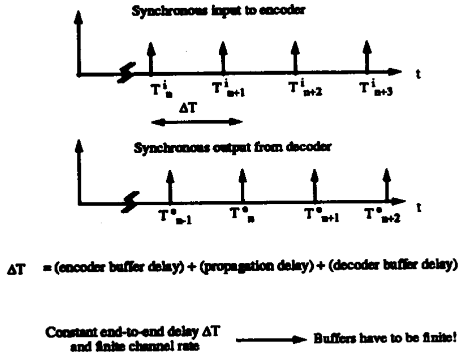
Synchronous input to encoder
t
T^i_n
T^i_{n+1}
T^i_{n+2}
T^i_{n+3}
ΔT
Synchronous output from decoder
t
T^o_{n-1}
T^o_n
T^o_{n+1}
T^o_{n+2}
ΔT = (encoder buffer delay) + (propagation delay) + (decoder buffer delay)
Constant end-to-end delay ΔT and finite channel rate
Buffers have to be finite!

Figure 2: Synchronous devices at encoder and decoder in a system with limited channel rate imply constant end-to-end delay and therefore finite buffering.

We can now formulate the problem of optimal buffer-constrained quantization:

Formulation 1 ***Given a set of quantizers, a finite buffer, an average throughput rate, and a sequence of blocks to be coded independently, select the optimal sequence of quantizers corresponding to each block such that the total cost measure is minimized and buffer overflow is avoided.***

2.2. An Integer Programming Formulation

A fundamental facet of this problem is that the set of available quantizers is finite in size, which discretizes the set of admissible solutions and makes it natural to look at integer programming (IP) formulations [5, 6] to help solve it.

Consider the allocation for N blocks and suppose there are M quantizers available to code each block. To denote the use of a given quantizer, the set of variables to be optimized is defined as: x_{ij}, which is 1 if quantizer j is used for block i and is 0 otherwise, for $i = 1, \ldots, N$ and $j = 1, \ldots, M$.

Let d_{ij} and b_{ij} be, respectively, the distortion and the number of bits produced by the coding of block i with quantizer j.

Formulation 2 *(0-1 Integer Programming)*
Given B_{MAX} (the buffer size) and r (the channel constant bit rate), find values for $x_{ij} \in \{0,1\}$ to minimize:

$$D_{TOT} = \sum_{i=1}^{N}\sum_{j=1}^{M} d_{ij} x_{ij}$$

subject to:

$$\sum_{j=1}^{M} x_{ij} = 1, \quad \forall i = 1, \ldots, N \tag{1}$$

and

$$\sum_{i=1}^{k}\sum_{j=1}^{M} b_{ij} x_{ij} - (k-1)\cdot r \leq B_{MAX}, \quad \forall k = 1, \ldots, N \tag{2}$$

Constraint (1) requires that only one quantizer is used for each block, while constraint (2) is the overflow restriction. Note that underflow will be avoided under the condition that distortion has to be minimized.

3. DYNAMIC PROGRAMMING SOLUTION USING THE VITERBI ALGORITHM

3.1. The Viterbi Algorithm

This problem can be solved using dynamic programming (DP) and, in particular, the Viterbi algorithm [7, 8], can be employed. The basic idea consists of starting at the initial buffer state and growing a path for every admissible quantizer (that does not cause buffer overflow), resulting in a trellis diagram whose states are the buffer-occupancy levels. See Figure 3. Each trellis path, corresponding to a quantizer choice, has a cost associated with it corresponding to the distortion incurred by the quantizer while the quantizer's coding bitrate dictates the destination state of the path. For an additive cost function, the well-known Viterbi algorithm provides the optimum choice of quantizers to code the sequence. This technique establishes a rule to prune out the suboptimal paths in the trellis: if a node can be reached by more than one path, only the minimum cost path will be kept.

3.2. The Optimal Solution

By investigating the optimal solution, one can study the characteristics of the optimal system configuration, and use it, as motivated earlier, as a benchmark for evaluating existing buffer-control practices and heuristics. For example,

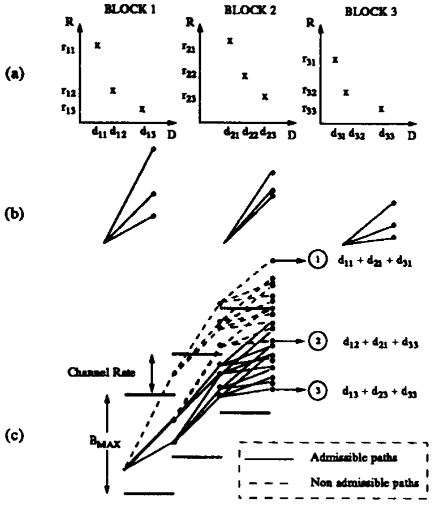
BLOCK 1
BLOCK 2
BLOCK 3
(a)
R
r_{11}
r_{12}
r_{13}
d_{11} d_{12} d_{13} D
R
r_{21}
r_{22}
r_{23}
d_{21} d_{22} d_{23} D
R
r_{31}
r_{32}
r_{33}
d_{31} d_{32} d_{33} D
(b)
(c)
1 $d_{11} + d_{21} + d_{31}$
2 $d_{12} + d_{21} + d_{33}$
3 $d_{13} + d_{23} + d_{33}$
Channel Rate
B_{MAX}
Admissible paths
Non admissible paths

Figure 3: The problem seen from the Viterbi algortihm point of view: (a) The rate-distortion characteristics of the the blocks for the available quantizers. (b) Equivalent representation. Each of the branches corresponds to the choice of a specific quantizer and has attached a cost. The length of the branch is proportional to the rate. (c) All possible paths for the three blocks considered. Path 2 cannot be used because of overflow. 1 and 3 are, respectively, the maximum and minimum distortion paths.

Figure 4 shows how, for different rates, increases in buffer size beyond a certain level produce no significant quality improvement.

As DP methods like the Viterbi algorithm are enumerative techniques with complexity that is exponential in the number of states in the trellis, it is useful to study the merits of reducing computational complexity at the cost of incurring acceptable suboptimality. Our results show that both reducing the number of nodes at each level (for example, by "quantizing" the number of buffer states – see Figure 5) and limiting the memory of the problem (i.e. confining the decision making, in releasing a branch in the path, to a finite number of consecutive sequence blocks) can reduce the complexity while not significantly decreasing the performance.

4. HEURISTIC METHODS TO APPROACH THE OPTIMAL SOLUTION

4.1. Rationale for the Heuristic Approach

To develop fast heuristics for our problem, we turn to rate-distortion theory by noting that our allocation problem *without the buffer constraint* reduces to the classical budget-constrained allocation problem cited in the literature [4], for which a fast Lagrange multiplier based solution exists.

Further, since our problem has an essentially limited memory (due to the finite size of the buffer), we can, in practice, reduce the horizon of our optimization problem to just a finite number of sequence blocks, i.e. we can decouple the future beyond a certain point from the decision on the current block, and this without significantly reducing the coding quality (see the top curve in Figure 7).

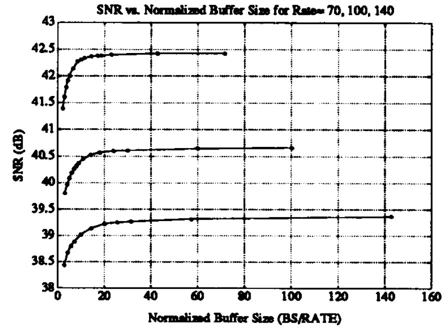

Figure 4: Optimally attainable SNR vs. normalized buffer size. Each point corresponds to the optimal quantization for the sequence at a given buffer size. Note that "rate" refers to the average number of bits used to code each source block.

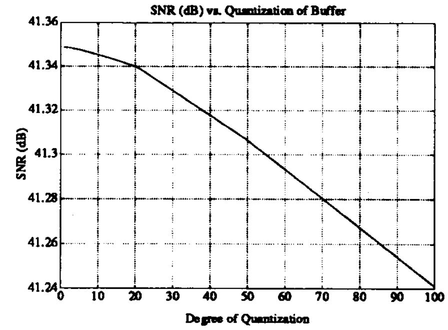

Figure 5: Effect of buffer quantization. The degree of quantization refers to the buffer state reduction factor, i.e. the number of consecutive physical buffer occupancy states that are "collapsed" into one virtual state, or node, in the trellis diagram.

4.2. Recursive Lagrangian Optimization

We combine these two ideas to formulate the following algorithm:

Algorithm 1

- *(step 1) At every stage k, use Lagrangian optimization [4], with budget constraint $n \cdot r + B(k) - B_{MAX}/2$, to obtain the best non-buffer-constrained allocation for the following n blocks, where $B(k)$ is the buffer occupancy level at the k-th stage as determined by the recursive algorithm.*
- *(step 2) Use the quantizer choice found by the previous step for block k and release it to the buffer, and repeat the first step for stage $k+1$.*

This is equivalent to performing a sliding window optimization so that the quantizer choice for the k-th block depends only on the rate-distortion characteristics of the following n blocks and on the buffer occupancy level at the k-th stage. Thus by exploiting the finite memory exhibited in practice by the problem (see results in Figures 4 and 7) and the fact that we can perform Lagrangian optimization very inexpensively due to the convexity of the resulting "unconstrained" case, we can approach the optimal solution. Our simulations verify that this heuristic yields solutions very close to the optimal one, as obtained using the more "brute force" Viterbi Algorithm. Experimental results have shown that, typically, using Algorithm 1 results in less than 10% of the blocks being coded with a non-optimal choice of quantizer (as computed with the Viterbi algorithm, under the same conditions).

4.3. Heuristic Improvement

A computationally efficient heuristic that sacrifices little quality is the following: use the algorithm described above, except undertake optimization over n blocks only when the algorithm results in paths whose buffer occupancy levels at any stage violate certain empirical thresholds. Call the heuristic percentage the fraction of the size of the buffer that is used as the threshold. Thus a 10% heuristic would mean that the algorithm would be recomputed when it results in any path whose buffer occupancy level is below 10% or above 90% of the total buffer size (note that a 50% heuristic would be equivalent to Algorithm 1). This algorithm can be formulated as:

Algorithm 2

(step 1) At every stage k, if the buffer occupancy is within the defined thresholds use the allocation previously computed for this block. Otherwise, use (step 1) of Algorithm 1 to compute the allocation for the following n blocks,

(step 2) Use the quantizer choice found by the previous step for block k and release it to the buffer, and repeat the first step for stage $k + 1$.

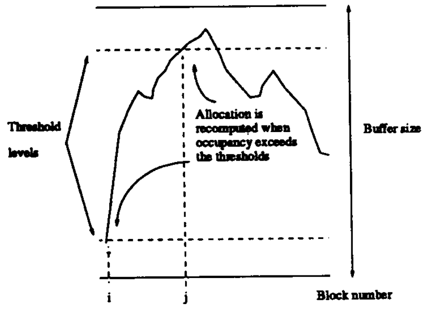
Threshold levels
Allocation is recomputed when occupancy exceeds the thresholds
Buffer size
i
j
Block number

Figure 6: Algorithm 2: the allocation, using Algorithm 1 for n consecutive blocks is recomputed only when the buffer occupancy exceeds the thresholds.

In Figure 7, the SNR of both the Viterbi solution with limited memory (top curve) and the heuristic (10%) approximation are compared. For a sufficiently large number of blocks, simulations indicate that the heuristic approximation comes within 0.05 dB of the optimal value, while consuming about 1/20 of the CPU time.

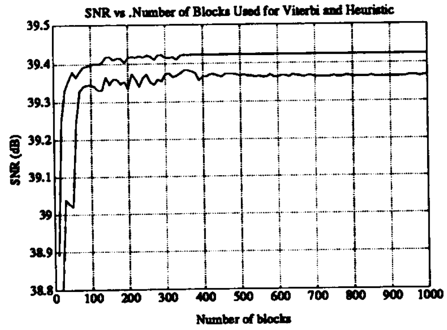

Figure 7: SNR Comparison between the Viterbi algorithm (top curve) and recursive 10% buffer-threshold Lagrangian heuristic for finite memory case. Note that the Viterbi algorithm is applied in a sliding window fashion, i.e. a on on a quantizer assignment is made based on the following n blocks and not on the whole sequence.

5. CONCLUSIONS

In this paper, we have examined the problem of optimal buffer-constrained independent quantization for an additive cost criterion. The problem is formulated in an integer programming framework and a way of reaching the optimal solution using the Viterbi algorithm is described. The results obtained from the optimal solution are studied and a fast heuristic algorithm, based on recursive Lagrangian optimization using rate-distortion concepts, is proposed which provides a close approximation to the optimal solution with much lower computational complexity.

REFERENCES

[1] "MPEG video simulation model three, ISO, coded representation of picture and audio information," 1990.

[2] "JPEG technical specification: Revision (DRAFT), joint photographic experts group, ISO/IEC JTC1/SC2/WG8, CCITT SGVIII," Aug. 1990.

[3] A. Ortega, K. Ramchandran, and M. Vetterli, "Optimal buffer-constrained source quantization and fast approximations," *IEEE Transactions on Image Processing*, Mar. 1992. Submitted.

[4] Y. Shoham and A. Gersho, "Efficient bit allocation for an arbitrary set of quantizers," *IEEE Transactions on ASSP*, vol. 36, pp. 1445–1453, Sept. 1988.

[5] M. Minoux, *Mathematical Programming: Theory and Algorithms*. Wiley, 1986.

[6] G. L. Nemhauser and L. A. Wolsey, *Integer and combinatorial optimization*. Wiley, 1988.

[7] A. J. Viterbi and J. K. Omura, *Principles of Digital Communication and Coding*. McGraw-Hill, 1979.

[8] J. G. Proakis, *Digital Communications*. McGraw-Hill, 1989.