A NOVEL HYBRID TECHNIQUE FOR DISCRETE RATE-DISTORTION OPTIMIZATION WITH APPLICATIONS TO FAST CODEBOOK SEARCH FOR SVQ[1]

*Youngjun Yoo** *Antonio Ortega ** *Kannan Ramchandran***

*Signal and Image Processing Institute
University of Southern California, Los Angeles, CA 90089
**Beckman Institute
University of Illinois, Urbana, IL 61801

ABSTRACT

A key part of any efficient source coder involves the optimal allocation of bit rate among a discrete set of competing quantization choices, as employed in the selected coding paradigm. This can be classified under the general label of budget-constrained discrete optimization, with coding applications including optimal bit allocation in scalar or vector quantizer based frameworks, entropy-constrained quantization frameworks, and codebook search for scalar-vector quantization (SVQ). For this class of problems, dynamic programming (DP) methods (such as the Viterbi algorithm) provide the optimal solution. However DP is typically very costly computationally. An alternate technique to solving this class of problems uses Lagrange mulipliers. This approach is much more efficient than DP but cannot guarantee optimality in general as it limits itself to convex-hull operating points, which may be sparse in many applications. In this work, we propose a novel hybrid technique that combines the speed of the Lagrangian approach with the versatility of the DP technique that is aimed at extracting the "best of both worlds." We present an application of our hybrid technique to the codebook search problem for SVQ, demonstrating significantly improved speed over the previously proposed DP-based search methods while mitigating the suboptimality of the Lagrangian based approach.

1. INTRODUCTION

With an increasing demand for efficiently compressed representation of images and video signals, the task of optimal bit allocation has assumed renewed importance: how should the available resources (coding rate R) be allocated to minimize the coding distortion D (measured using some metric)? Rate-distortion (R-D) optimization pervades all of source coding, both from an information-theoretic standpoint as well as for the design of practical coding systems (where the term *operational* rate-distortion is used). Applications include the traditional design of off-line vector quantizers (VQ) based on optimization of statistical training data [1], as well as more recent non-training based frameworks where the encoder makes "on the fly" decisions on how many bits to assign to different coding units in the system [2], or how to find the best subtree of a global quantization and/or transform tree [3, 4, 5]. An example of the benefits of R-D optimized allocation can be found in [6], where an R-D optimization of the zerotree coding framework of [7] results in significant performance gain.

Typically, an operational R-D optimization problem involves the specification of the choice of coding parameters in the picked framework which minimize the coding distortion for a given available bit rate budget. A key point is that the set of parameters is a discrete set therefore dictating the use of discrete optimization techniques. The encoder's task is then to find a good (if not optimal) operating point from the available discrete set of operating points, i.e. to search through the operational R-D space for the point which has the least distortion while not exceeding the target rate. An additional requirement is obviously that the search be computationally efficient.

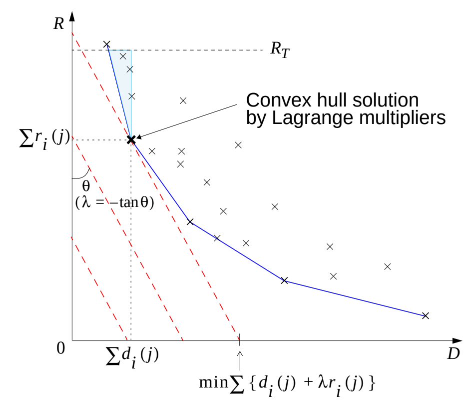

Figure 1. **Lagrangian optimization using the constant slope method. The convex hull solution is determined as the first point hit by a plane wave of slope $-1/\lambda$. The corresponding minimum cost is represented as a D-intercept of the plane wave. The shaded area corresponds to points that can be potentially optimal but cannot be reached using convex hull techniques. For example, if the rate budget was R_T we would have to settle for the nearest convex hull point below R_T.**

In this paper we will describe efficient ways of finding this operating point for the important class of block-based coders which consist of independently coded blocks that can change their coding parameters (e.g. quantizers) on a block by block basis. In this regard, our problem framework is similar to that considered by Shoham and Gersho [2] with the vital distinction that we do not restrict ourselves to operating points that are on the convex hull of the coder's R-D characteristic (see Fig. 1 for an example of a non-convex hull solution which is *not* attainable by the Lagrange multiplier based method of [2].) Typical examples of the application scope of our proposed technique include coding of intraframe (I-frame) blocks in MPEG (which typically occupy 75% of the overall budget), entropy-constrained scalar/vector quantization (ECSQ/ECVQ) [8, 9], and in general any block-coding type framework where convex-hull operating points will not suffice in general. An illuminating

[1] This work was supported in part by the National Science Foundation under awards MIP-9502227 (CAREER-95) and MIP-9409587 (RIA-94).

example of such a framework involves the problem of optimal codebook search for the recently introduced Scalar-Vector Quantizer (SVQ) [10] which is a structured fixed-rate quantizer that approximates the performance of ECSQ. We will present experimental results in Section 4 verifying the superior performance of our proposed technique over the (standard) dynamic-programming (DP)-based method proposed in [10] for the SVQ codebook search problem.

We now provide a brief motivation for our proposed method, a hybrid between the Lagrangian-based and DP-based optimization techniques.

The Lagrangian-based optimization addressed in [2] has the advantage of being fast but, as mentioned, has the drawback of achieving solutions that lie strictly on the convex hull of the available rate distortion characteristic. While this is not a major problem when the convex-hull is densely populated, sparse convex-hull sets are problematic and may result in unacceptably suboptimal performance. This can occur in practical scenarios in tree-based coding when trying to operate at low rates, as mentioned in [4]. Note that the suboptimality of the Lagrangian approach stems from the fact that the set of operating points is discrete. In a continous optimization problem where all optimal operating points necessarily live on the convex hull of the R-D curve, the Lagrangian approach will always yield the optimal operating point. While in the discrete case, statistical "time-sharing" of convex hull operating points can optimize performance [3], this is usually not feasible in most image coding applications, where a "one-time" coding of the image blocks is called for.

At the other end of the spectrum lives the class of DP-based techniques which are guaranteed to find the optimal operating point, whether or not they are convex-hull residents (see Section 3.1 for details). Indeed, for the SVQ codebook search problem, Laroia and Farvardin in [10] propose such a DP-based method to find the optimal operating point. The main drawback of the DP technique is that optimality comes at the price of a substantial increase in complexity, and indeed for most practical "on-line" image coding scenarios, dynamic programming is out of the question unless the number of candidate operating points is small or can be reduced.

In this paper, we introduce a novel hybrid method that is targeted at combining the advantages of the Lagrangian and DP-based approaches. The basic idea is to use the fast Lagrangian-based solution as an initial guess for a DP-based technique that has a fraction of the complexity of the regular full-DP approach. We will demonstrate how our proposed technique finds application in formulating a much improved way of solving the SVQ codebook search compared to the DP-based approach suggested in [10]. Although we use SVQ as a vehicle to demonstrate our technique, we emphasize that its scope of application is quite broad, and includes any R-D optimization framework where convex-hull-only operating points may not suffice. Indeed, its scope extends beyond image coding or even engineering, as it addresses the universal problem of optimal resource allocation in a discrete optimization setting [11].

The paper is organized as follows. In Section 2 we formulate the bit allocation problem In Section 3 we describe the two above-mentioned techniques, DP and Lagrangian optimization. In Section 4 we introduce a set of hybrid techniques which will allow us to trade off performance and complexity. Finally, in Section 5 we present the results of applying this new technique to codebook search for SVQ.

2. PROBLEM FORMULATION

Suppose we have N input blocks, with M different quantizers being available to quantize each block. Assume that the rate and the distortion, $r_i(j)$ and $d_i(j)$ respectively, for each block i and quantizer j are known. Our goal is to find, among all the possible allocations $\mathbf{x}$ such that a quantizer $x(i)$ is assigned to block i, the optimal allocation $\mathbf{x}^*$ which minimizes the total distortion

$$\mathbf{x}^* = \arg\min_{\mathbf{x}} \sum_{i=1}^{N} d_i(x(i)) \qquad (1)$$

while not exceeding the predetermined rate budget R_T, i.e.,

$$\sum_{i=1}^{N} r_i(x^*(i)) \leq R_T. \qquad (2)$$

This scenario can be encountered for example in block-based DCT image coding where each input block in the above formulation corresponds to a DCT coded block and each of the quantizers corresponds to a choice of quantization scale as in MPEG.

3. STANDARD OPTIMIZATION TECHNIQUES

3.1. Lagrangian Optimization

The Lagrange multiplier method has been a popular choice for the above problem [2]. The basic idea is to introduce the Lagrangian cost

$$J = \sum_{i=1}^{N} (d_i(j) + \lambda r_i(j)). \qquad (3)$$

Then the search is performed by choosing, for a given Lagrange multiplier λ, the quantizer $j(\lambda)$ which minimizes the cost $d_i(j) + \lambda r_i(j)$ at a given stage. The goal is then to find the Lagrangian cost which results in a total rate $\sum_{i=1}^{N} r_i(j)$ that is just below the desired budget R_T. If the budget is exactly met then this solution is also optimal.

This technique has a limitation of reaching only the codevectors on the convex hull of the R-D characteristics (see Fig. 1) and thus only approximates the performance of the optimal DP-based algorithms. This shortcoming could be crucial in problems such as SVQ where the block sizes are small and where the convex hull is sparse.

3.2. Dynamic Programming

To optimally solve the above allocation problem we can alternatively resort to dynamic programming (DP). The algorithm sequentially builds a trellis to represent possible solutions. Refer to Fig. 2 where the x axis corresponds to the stage (or input block) and the y axis represents the accumulated rate. Each trellis stage corresponds to one of the N inputs and each trellis state is associated with the accumulated distortion and rate. For example, consider the node (i,s) in the trellis which represents a solution for which s bits have been used up to block i at a total distortion cost Δ. Then if we select quantizer j for block $i+1$ we will generate a branch connecting (i,s) to $(i+1, s+r_{i+1}(j))$ and the accumulated distortion will be $\Delta + d_{i+1}(j)$. DP can then be used in the usual way to prune out the trellis. Based on Bellman's optimality principle of all paths arriving to a node only the one have less distortion has any chance of being globally optimal. Thus we can eliminate out of all the paths with the same total rate at a given stage, those with larger distortion.

This method has the advantage of achieving solutions that are not reachable using the Lagrangian method. This is particularly useful when, as is the case in SVQ, there might be "gaps" in the convex hull. However the major drawback is that the complexity of the search grows linearly with the number of states. Since for each stage (or block) each state represents a possible total rate, it is clear that the number of states and thus the complexity will increase linearly with the rate budget R_T, thus making this technique impractical in many situations. While this method can be efficient for small vector size SVQ it would clearly not be useful for bit allocation in a large image!

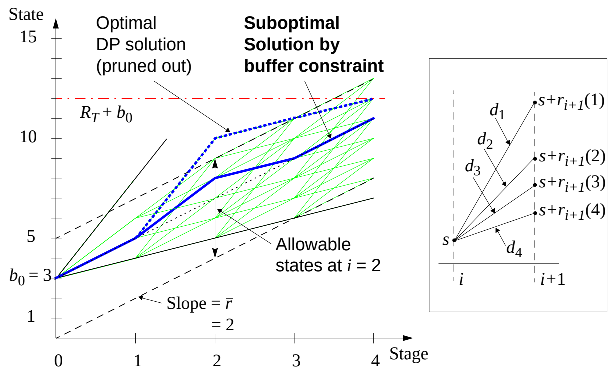

Figure 2. **DP and Buffer-constrained DP optimization. In the general DP case the number of states grows linearly. When the "buffering" constraint is introduced the number of states at each stage is kept constant. The states are allowed only between the buffer state bounds denoted by the dashed lines. In this example for the chosen buffer size the optimal solution cannot be achieved. However note that one could always find a sufficiently large buffer that would not prune out the optimal DP solution.**

4. LAGRANGIAN-INITIALIZED BUFFERED OPTIMIZATION

4.1. Buffer constrained DP optimization

Since the linear growth in the number of states in the DP formulation is the major obstacle to its applicability for large input sizes we first propose a method to maintain a fixed number of states in order to reduce the computational burden. Consider $\bar{r} = R_T/N$, the average rate per block, as the average value that is allocated to each block by the optimal solution. Then, the "average" solution path through the trellis can be defined as that one where most branches use rate close to $\bar{r}$. Note that this may not correspond to an actual path since $\bar{r}$ may not be integer but we use it to illustrate the idea. Now, the basic idea is to assume that most paths will not deviate too much from the average path and thus we introduce a new state, b_i at stage i as

$$b_i = s_i - \bar{r}, \qquad (4)$$

where s_i is the state at stage i as initially introduced in the previous section. This can be seen to be equivalent to considering b_i as a virtual buffer state. This virtual buffer keeps track of how much each path deviates in rate from the average path. Since most paths tend to be close to the average, we can impose an additional constraint so that the admissible solutions $\mathbf{x}$ not only need to have total rates bounded by R_T but also cannot exceed certain finite buffer state bounds at every stage i. Hence we now search for the minimum distortion solution satisfying,

$$b_{min} \leq b_i \leq b_{max} \quad \text{for all } i, \qquad (5)$$

as well as the constraint (7), where b_{min} and b_{max} are appropriately chosen buffer state bounds[2]. With this change the same DP techniques can be used except that now the number of states is constant (as defined by our choice of buffer size).

In this formulation, which is analogous to the buffer constrained optimization of [12], the main benefit is that the number of states is kept constant and thus complexity is kept reasonable even for large N. The additional constraint may potentially eliminate the true optimal solution (see Fig. 2), however, in general we can expect that this will not occur often (and thus suboptimality will be limited). Note, however that a large enough buffer can *always* be found such that the optimal solution can still be achieved.

4.2. Lagrangian-initialized Buffer Constrained Optimization

The new proposed technique is based on the intuitively obvious fact that even though a convex hull solution may not be optimal it may be *close* to the optimal solution. We thus first run the Lagrangian algorithm and then use this solution as our initial guess. This is accomplished by employing the buffered optimization introduced in Section 4.1 *but using the Lagrangian solution as the virtual channel rate instead of* $\bar{r}$.

More formally, let $r_i(\lambda)$ be the optimal rate assignment using the Lagrangian technique for the value of λ which results in a total rate closest to (but smaller than) the budget. Then we define a new virtual buffer state b'_i for block i as

$$b'_i = s - r_i(\lambda) \qquad (6)$$

we then apply the buffer constrained optimization algorithm of section 3.2 to find the optimal solution. In this manner we are bounding the solution to be *close* to that of the Lagrangian technique. For sufficiently large buffer size we can guarantee that the optimal solution will be achieved, as confirmed by our experiments. Since the Lagrangian technique already gives a good approximation the important point to note is that *the buffer size needed to guarantee optimality will typically be small.* Note also that while other Lagrangian techniques to reach non-convex hull points have been proposed [2] they do not guarantee optimality.

5. APPLICATION TO SVQ

5.1. Scalar-Vector Quantization

Scalar-vector quantization (SVQ), a fixed-rate VQ scheme, has been proposed to approximate the performance of ECSQ while being robust in noisy environments [10]. SVQ is theoretically justified by the *asymptotic equipartition property* (AEP) for i.i.d. random variables [13]. From the AEP, a code with fixed-length close to n-times the source entropy can represent most of the sample sequences of length n with minimal error. The basic idea of SVQ is to use an underlying scalar quantizer (USQ) and associate a "length" based on the sample entropy to each of its quantization levels. Of all possible combinations of USQ levels, only those with sample entropies (i.e., the total length) no greater than a *threshold* are made part of the codebook. The threshold is determined so that the total number of codevectors in the codebook matches the design rate for SVQ.

SVQ quantizes an input vector by choosing, among all the combinations of USQ levels, the one which minimizes the distortion *without exceeding the threshold.* More formally, an m-dimensional rate-n scalar-vector quantizer (SVQ) can be defined by its parameter triple $(\mathcal{Q}, \mathcal{L}, L)$, i.e., (i) $\mathcal{Q}$, the set of the n $(n \geq 2^r)$ levels q_i of an underlying scalar quantizer (USQ), (ii) $\mathcal{L}$, the set of the corresponding lengths ℓ_i based on the source entropy, and (iii) L, a threshold on the

[2] We emphasize that there is no actual buffer at the encoder and we just use the analogy between buffer-constrained allocation and the SVQ encoder to clarify our algorithm. Also note that b_{min} can be negative.

total length $\sum_{k=1}^{m} \ell(z_k)$ for a codevector $\mathbf{z} \equiv (z_1, \cdots, z_m)$ where $\ell(z_k) \in \mathcal{L}$ is the length of z_k. Scalar quantizing m samples is conceptually equivalent to vector quantizing an m-vector with a VQ codebook containing all n^m possible combinations in $\mathcal{Q}^m$. The SVQ codebook contains only codevectors such that

$$\sum_{k=1}^{m} \ell(z_k) \leq L \tag{7}$$

where L limits the codebook size ($\leq 2^{mr}$) making SVQ fixed-bit-rate.

We can readily see that SVQ encoding is analogous to the allocation problem introduced in Section 2, where the rate $r_i(j)$ is replaced by the length associated to each quantization level $\ell(z_k)$ and the rate budget R_T is replaced by the length threshold, L. Thus all the described techniques can be applied in the SVQ context. In this particular case the allocation problem results in a sparse R-D characteristic and it is thus a perfect testbed for our algorithm.

5.2. Experimental Results

We now compare four codebook search schemes which represent the four resource allocation algorithms discussed in this paper. These are: (i) optimal (full search) DP (SVQ), the method originally proposed in [10], (ii) Lagrangian approximation (L-SVQ), (iii) buffer-constrained DP (B-SVQ), and (iv) Lagrangian-initialized buffer-constrained DP (LB-SVQ). Note that our results do not affect the codebook design, only the codebook search.

Results are provided for a Gaussian sequence of 80,000 i.i.d. samples with Gaussian distribution $N(0, 1^2)$. For all three schemes we use identical SVQ parameters $(\mathcal{Q}, \mathcal{L}, L)$ designed on a distinct training set of 40,000 samples with $N(0, 1^2)$ using the methods in [10]. The SVQ rate is $r = 2.0$ (bits/sample) and the number of the USQ quantization levels is $n = 7$.

As discussed above the buffer size in buffer-constrained DP environments can be selected so as to guarantee that the optimal solution is achieved. In this case we obtain the minimum buffering requirements to guarantee optimality for the given codebook. We denote B_U and B_L, respectively, the upper and lower bounds on the buffer state that guarantee that the optimal DP solution is still achieved. In our results B-SVQ$_{P\%}$ denotes the buffer constrained codebook search where P% indicates the buffer size relative to the buffer requirements determined by B_U and B_L. Thus B-SVQ$_{100\%}$ results in optimal performance (equal to SVQ) while decreasing values of P produce suboptimality but allow faster search.

The data obtained for L-SVQ are plotted also in Fig. 3. Compared to the other schemes, the codebook search by L-SVQ is extremely fast. However, we can observe the suboptimality due to the non-convex hull points. We finally experiment with LB-SVQ where we use B_U and B_L for the buffering constraint. The performance of LB-SVQ is, as we expected, optimal for all vector dimensions while the reduction in search time is very significant (over an order of magnitude with respect to DP).

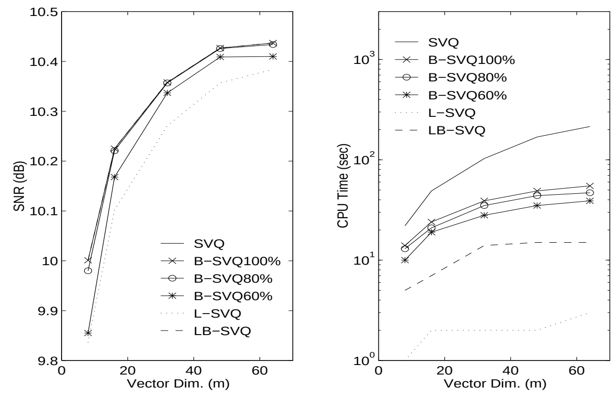

Figure 3. **Comparison of codebook search algorithms: SVQ with optimal full-search DP, B-SVQ for various buffer sizes, L-SVQ, and LB-SVQ with B_U and B_L. B-SVQ$_{P\%}$ represents the buffered SVQ with P% of B_U and B_L as its maximum and minimum allowable states. Note that both B-SVQ$_{100\%}$ and LB-SVQ achieve the optimal (DP) operating point, thus these three curves are superimposed. LB-SVQ guarantees optimality with a reduction of about an order of magnitude in search time. L-SVQ is the fastest scheme but is limited to non-convex hull points and thus cannot achieve optimality.**

REFERENCES

[1] A. Gersho and R. M. Gray, *Vector Quantization and Signal Compression.* Boston, MA: Kluwer Academic Press, 1992.

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

[3] P. A. Chou, T. Lookabaugh, and R. M. Gray, "Optimal pruning with applications to tree-structured source coding and modeling," *IEEE Trans. on Inform. Theory*, vol. 35, pp. 299–315, Mar. 1989.

[4] S.-Z. Kiang, R. L. Baker, G. J. Sullivan, and C.-Y. Chiu, "Recursive optimal pruning with applications to tree structured vector quantizers," *IEEE Trans. on Image Proc.*, vol. 1, pp. 162–169, Apr. 1992.

[5] K. Ramchandran and M. Vetterli, "Best wavelet packet bases in a rate–distortion sense," *IEEE Trans. on Image Proc.*, vol. 2, pp. 160–175, Apr. 1993.

[6] Z. Xiong, K. Ramchandran, and M. Orchard, "Joint optimization of scalar and tree-structured quantization of wavelet image decomposition," in *Proc. of Asilomar Conf.*, (Pacific Grove, CA), pp. 891–895, Nov. 1993.

[7] J. M. Shapiro, "Embedded image coding using zerotrees of wavelet coefficients," *IEEE Trans. on Signal Processing*, vol. 41, pp. 3445–3462, Dec. 1993.

[8] N. Farvardin and J. W. Modestino, "Optimum quantizer performance for a class of non-gaussian memoryless sources," *IEEE Trans. IT*, vol. IT-30, pp. 485–497, May 1984.

[9] P. A. Chou, T. Lookabaugh, and R. M. Gray, "Entropy-constrained vector quantization," *IEEE Trans. ASSP*, vol. 37, pp. 31–42, Jan. 1989.

[10] R. Laroia and N. Farvardin, "A structured fixed rate vector quantizer derived from a variable-length scalar quantizers: Part I - Memoryless sources," *IEEE Trans. Inform. Theory*, vol. 39, pp. 851–867, May 1993.

[11] H. Everett, "Generalized Lagrange multiplier method for solving problems of optimum allocation of resources," *Operations Res.*, vol. 11, pp. 399–417, 1963.

[12] A. Ortega, K. Ramchandran, and M. Vetterli, "Optimal trellis-based buffered compression and fast approximation," *IEEE Trans. on Image Proc.*, vol. 3, pp. 26–40, Jan. 1994.

[13] T. M. Cover and J. A. Thomas, *Elements of Information Theory.* Toronto, Canada: J. Wiley & Sons, 1991.