LOCALLY OPTIMAL JOINT ENCODING OF IMAGE TRANSFORM COEFFICIENTS

Piya Bunyaratavej and David J. Miller

Department of Electrical Engineering
The Pennsylvania State University
Rm. 227-C EEW, University Park, PA 16802
e-mail:miller@perseus.ee.psu.edu
Tel: (814) 865-6510
Fax (814) 865-7065

ABSTRACT

We address the choice of encoder for conditional entropy-constrained trellis-coded quantization (CECTCQ), applied to image transform coefficients. The optimal CECTCQ encoder requires an (utterly intractable) exhaustive search and the standard method of greedy, sequential encoding of the coefficient "sources" is suboptimal. Alternatively, we suggest a *locally* optimal encoding algorithm, guaranteed to improve performance over greedy encoding, and yet with manageable increases in encoding complexity. This method uses dynamic programming as a local optimization encoding "step", repeatedly applied until convergence. Simulations demonstrate up to 1.5 dB gain over greedy CECTCQ encoding of block-transformed images.

1. INTRODUCTION

In virtually all practical, high-performance image coding systems, a transform or subband filtering operation is applied, with the aim of achieving data compaction, as well as decorrelation, of the resulting image features. When scalar quantization is separately applied (based on a bit allocation strategy) to each transformed feature, the resulting system is a constrained form of high-dimensional vector quantization (VQ). Although this is a heuristic form of VQ, it achieves good performance, and unlike full-search VQ, it is quite practically feasible for the (rate, vector dimension/block size) pairs (e.g. (1 bpp, 64 dimensions)) that are typically considered. While both transform and subband coding mitigate the suboptimality of scalar quantization, significant rate-distortion performance gains can still be achieved through some form of *joint* feature encoding, rather than individual feature encoding. In particular, the lossy JPEG standard (and, in a more ambitious way, hierarchical wavelet coders [1]) apply efficient techniques for approximate, joint entropy coding of quantized image features. Likewise, entropy-constrained trellis-coded quantization (ECTCQ) [2],[9], when applied in a transform coding context , jointly encodes the samples of the coefficient "source" at each spatial frequency. The former (JPEG) scheme only exploits statistical dependencies in the quantized source within each transformed block, while the latter (ECTCQ) approach only captures statistical homogeneity and redundancy of like coefficients *across* blocks, without exploiting intra-block redundancy. However, intra-block joint entropy coding and inter-block trellis coding are complementary, which motivates coding schemes based on both paradigms. One such approach is *conditional* ECTCQ (CECTCQ) of block-transformed images, with conditioning used to capture both intra-block *and* inter-block redundancy. Conditional entropy-constrained (CEC) encoding has been effectively applied in several coding contexts [4],[5]. However, for CEC coding of images, one difficulty is the choice of the encoder – *optimal* CEC encoding for typical conditioning contexts is utterly intractable, while standard greedy encoding methods may be quite suboptimal[1]. The fact that there are few techniques which bridge the gap between these extremes motivates the present work, which introduces a *locally* optimal CECTCQ encoder, applied to image transform coefficients. This technique provides performance gains of up to 1.5 dB over greedy encoding at a cost of increased, albeit manageable implementation complexity. In the next section we describe the joint feature encoding problem and then develop our locally optimal encoding algorithm. In section 3, experimental results are presented.

[1] This intractability is actually *not* attributable to the choice of TCQ as the quantization method. Even if scalar quantizers are used, optimal CEC encoding of transformed images is intractable for typical conditioning contexts, as will be further indicated in the sequel.

2. CONDITIONAL ENTROPY-CONSTRAINED ENCODING FOR A COLLECTION OF SOURCES

2.1. Preliminaries

Consider a collection of source sequences $\{\underline{s_1}, \underline{s_2}, \ldots, \underline{s_M}\}$, where $\underline{s_m} \equiv (s_{m1}, s_{m2}, \ldots, s_{mN})$, $s_{mn} \in \mathcal{R}$. For transform coding of images, this collection is obtained by dividing the image into blocks (e.g. 8×8), applying a linear transform (e.g. 2D DCT) to each block, and then grouping together all coefficients at a common frequency, effectively creating a "source" at each spatial frequency. Thus, the number of sources, M, is given by the block size, with N the number of blocks in the image. Each source is represented as a 1D sequence by using a fixed scan order for visiting transformed blocks, e.g. row-by-row. Moreover, the sources are themselves ordered, usually according to increasing spatial frequency, e.g. via a zig-zag scan defined for coefficients within a block. The resulting collection of sources can be organized as a 2D array, with source $\underline{s_m}$ given by row m (see Figure 1).

The encoding paradigm we suggest can be applied to systems using either scalar or trellis quantization. In the experiments, we will evaluate our method for the (more powerful) trellis coding framework. However, for clarity and concision, we develop the method in this section assuming scalar quantizers. The description for trellis quantizers follows naturally from this development, and will thus be omitted.

Denote the quantization index used for sample s_{mn} by i_{mn}, with the associated quantization level given by the discrete mapping $q_m(i_{mn})^2$. Furthermore, we will need to represent the *sequence* of encoded indices for each source, i.e. $\underline{i_m} \equiv (i_{m1}, i_{m2}, \ldots, i_{mN})$. The difficulties with optimal CEC encoding arise due to the statistical dependencies assumed to exist between the quantization indices. While even more complex models can be considered, it will suffice in this section (to illustrate both difficulties and our new paradigm) to assume a *second-order Markov model* (see Figure 1), with the probability of i_{mn} conditionally independent of its causal support given both $i_{m,n-1}$ and $i_{m-1,n}$, i.e. the model will be based on $\{\mathrm{Prob}[\mathrm{i_{mn}}|\mathrm{i_{m,n-1}}, \mathrm{i_{m-1,n}}]\}^3$. These probabilities, which capture both intra-source and inter-source statistical dependency, drive formation of the Huffman code tables. Accordingly, the variable length codeword for i_{mn} will be a function of $i_{m,n-1}$ and $i_{m-1,n}$, as will its length in bits, $l(i_{mn}; i_{m,n-1}, i_{m-1,n})$.

[2] The index i_{mn} will have range $\{0, 1, 2, \ldots, L_m - 1\}$, where L_m is the number of levels allocated for source $\underline{s_m}$.

[3] This reduces to first order at the left and upper array borders and to zeroth order in the upper left corner, i.e. we have $\mathrm{Prob}[\mathrm{i_{1,n}}|\mathrm{i_{1,n-1}}], \mathrm{n} \geq 2$, $\mathrm{Prob}[\mathrm{i_{m,1}}|\mathrm{i_{m-1,1}}], \mathrm{m} \geq 2$, and $\mathrm{Prob}[\mathrm{i_{11}}]$.

2.2. Formulation

The objective of the CEC encoder is to select an array of quantization indices to minimize the Lagrangian cost function $J \equiv D + \lambda B$, with D the quantizer distortion and B the source coding description length in bits. We find it useful to represent J as a sum of *source-wise* costs, i.e.

$$J = (D_1(\underline{i_1}) + \lambda B_1(\underline{i_1})) + \sum_{m=2}^{M} (D_m(\underline{i_m}) + \lambda B_m(\underline{i_m}, \underline{i_{m-1}})). \tag{1}$$

Here, $D_m(\underline{i_m}) = \sum_{n=1}^{N} d(s_{mn}, q(i_{mn}))$ with $d(\cdot,\cdot)$ a specified scalar distortion measure, $B_1(\underline{i_1}) = l(i_{11}) + \sum_{n=2}^{N} l(i_{1n}; i_{1,n-1})$, and $B_m(\underline{i_m}, \underline{i_{m-1}}) = l(i_{m1}; i_{m-1,1}) + \sum_{n=2}^{N} l(i_{mn}; i_{m-1,n}, i_{m,n-1}),\ m \geq 2$. In (1), notation in $B_m(\cdot)$ involving $\underline{i_m}$ and $\underline{i_{m-1}}$ emphasizes that the bit length associated with the mth source ($m \geq 2$) depends on the encoding choices for both $\underline{s_m}$ and $\underline{s_{m-1}}$. It is this dependence which makes optimal CEC encoding only achievable by an exhaustive search over all possible ($\prod_{m=1}^{M} (L_m)^N$) image encodings. Since this exhaustive encoder is quite infeasible, a practical, albeit *greedy* alternative must typically be used, e.g. [3].

Greedy CEC Encoding

1. Use dynamic programming (Viterbi algorithm) to minimize $D_1(\underline{i_1}) + \lambda B_1(\underline{i_1})$ over $\underline{i_1}$. Let $\underline{i_1^{(0)}}$ denote the solution.

2. For m=2 to M
Use dynamic programming to minimize $D_m(\underline{i_m}) + \lambda B_m(\underline{i_m}, \underline{i_{m-1}^{(0)}})$ over $\underline{i_m}$. Let $\underline{i_m^{(0)}}$ denote the solution.
End

While this procedure is *step-wise* optimal in the sense of choosing the best sequence $\underline{i_m}$ given fixed $\underline{i_{m-1}^{(0)}}$, it does not guarantee even a *locally* optimal collection $\{\underline{i_m},\ m = 1, \ldots, M\}$. In particular, after determining $\underline{i_m^{(0)}}$ given $\underline{i_{m-1}^{(0)}}$, it is quite possible that some index $i_{m-1,n}$ could be re-chosen to reduce the Lagrangian cost. In fact, such "revisiting" of previously made encoding choices is what forms the basis of our new paradigm.

Crucial to our method is a new "source-wise" cost to be minimized in choosing $\underline{i_m}, m < M$. Note in particular that the sequence $\underline{i_m}$ affects $D_m(\underline{i_m})$, $B_m(\underline{i_m}, \underline{i_{m-1}})$, *and* $B_{m+1}(\underline{i_{m+1}}, \underline{i_m})$. Thus, suppose we have already used the greedy encoding technique to determine $\{\underline{i_m^{(0)}}, m = 1, \ldots, M\}$. Re-selecting $\underline{i_m}$ to minimize $D_m(\underline{i_m}) + \lambda B_m(\underline{i_m}, \underline{i_{m-1}^{(0)}})$ will simply lead again to the solution $\underline{i_m^{(0)}}$. However, suppose instead that we now minimize $D_m(\underline{i_m}) + \lambda B_m(\underline{i_m}, \underline{i_{m-1}^{(0)}}) +$

$\lambda B_{m+1}(\underline{i^{(0)}_{m+1}}, \underline{i_m})$. This minimization can *also* be achieved via dynamic programming, simply by adding the terms $\lambda l(i^{(0)}_{m+1,n}; i_{m,n}, i^{(0)}_{m+1,n-1})$ to the branch metrics used in the dynamic programming algorithm for minimizing $D_m(\underline{i_m})+ \lambda B_m(\underline{i_m}, \underline{i^{(0)}_{m-1}})$. Since this new "source-wise" cost $(D_m(\underline{i_m})+ \lambda B_m(\underline{i_m}, \underline{i^{(0)}_{m-1}}) + \lambda B_{m+1}(\underline{i^{(0)}_{m+1}}, \underline{i_m}))$ is composed of $(D_m(\underline{i_m}) + \lambda B_m(\underline{i_m}, \underline{i^{(0)}_{m-1}}))$ plus an *additional* term from J, and since dynamic programming is guaranteed to find the global minimum solution, the resulting sequence $\underline{i^{(1)}_m}$ *must* be at least as good in the sense of J as the initial one, $\underline{i^{(0)}_m}$. Moreover, in practice, such minimizations are likely to provide at least some reduction in J. Thus, we suggest such minimizations as the basis for the following iterative encoding algorithm.

Iterative, Locally Optimal CEC Encoding

1. Implement the greedy encoding procedure to determine $\{\underline{i^{(0)}_m}, m = 1, \ldots, M\}$. Set $t = 0$.

2. Do {

$t \leftarrow t + 1$

Choose $\underline{i_1}^{(t)}$ to minimize $D_1(\underline{i^{(t)}_1}) + \lambda B_1(\underline{i^{(t)}_1}) + \lambda B_2(\underline{i^{(t-1)}_2}, \underline{i^{(t)}_1})$ via dynamic programming.

For m=2 to M-1
Choose $\underline{i^{(t)}_m}$ to minimize $D_m(\underline{i^{(t)}_m}) + \lambda B_m(\underline{i^{(t)}_m}, \underline{i^{(t)}_{m-1}}) + \lambda B_{m+1}(\underline{i^{(t-1)}_{m+1}}, \underline{i^{(t)}_m})$ via dynamic programming.
End

Choose $\underline{i_M}^{(t)}$ to minimize $D_M(\underline{i^{(t)}_M})+\lambda B_M(\underline{i^{(t)}_M}, \underline{i^{(t)}_{M-1}})$ via dynamic programming.

} (Until (There are no encoding changes) OR (a convergence criterion is met))

2.3. Algorithm Summary

1. Each iteration (t) has complexity roughly equivalent to that of greedy encoding. Thus, the encoding complexity is roughly (number of iterations + 1) times greater than greedy encoding. In practice, most of the cost reduction is gleaned after 4-5 iterations.

2. Each dynamic programming "step" is non-increasing in J.

3. The algorithm is an extension of an iterative encoding technique first proposed in [6] for product code VQ, e.g. for mean-gain-shape VQ. Our method extends this technique, a) by jointly optimizing over a much larger encoding space (a full image) and b) by taking optimization steps that are substantially less greedy – in [6], one encoder index was re-optimized at a time (e.g. a single mean, gain, or shape index) with all the remaining ones fixed, whereas in our approach a long *sequence* of indices is re-optimized together, given the remaining ones fixed.

4. If TCQ, rather than scalar quantization is used, the "pseudocode" description remains the same. Note in particular that dynamic programming can still be used to optimize each sequence $\underline{i_m}$, albeit *again* by suitably modifying the branch metrics (used by dynamic programming), this time to account for the constrained set of TCQ trellis quantization choices.

5. Our iterative *encoding* method can also be directly applied to the problem of joint source-channel *decoding* for quantization information transmitted across a noisy channel (again, using a Markovian index model). In this context, the algorithm iteratively maximizes the joint likelihood of transmitted indices given received ones.

3. EXPERIMENTAL RESULTS

We have evaluated our method for transform coding, in comparison with greedy CECTCQ encoding and with ECTCQ (where no conditioning was used). We used 8×8 image blocks, the 2-D DCT transform, and a zig-zag scan for 1-D ordering of the sources. Four-state TCQ was used on each source, with the trellis as given in the original TCQ paper [7]. The codebook size for each source was determined by a standard bit allocation strategy. A training set of 8 images was used for designing the coders (the TCQ codebooks and the Huffman code tables). One Huffman code was designed for each (TCQ state, conditioning context) pair. For ECTCQ, the standard design algorithm was used, based on set partitioning for code initialization and an entropy-constrained version of Stewart and Gray's trellis-based Lloyd algorithm [8], to refine the quantization levels and entropy codes [2],[9]. For standard greedy CECTCQ, a sequential design algorithm (paralleling the encoder's operation) was employed, with the coders for each individual source designed sequentially (in zig-zag scan order), with each in turn then used to create conditioning context for the next source coder design. Finally, for our new CECTCQ coder, we also devised a design algorithm matched to the encoder's operation. First, the (just described) standard CECTCQ design was used to obtain initial coders. Then, paralleling the iterative encoder, we implemented an iterative *design* algorithm, with each source coder in turn rechosen to minimize its associated "source-wise" Lagrangian cost, with the minimization over both the quantizers and the entropy codes. Cycling through the coder designs continues until either there are no further changes or until a stopping condition is reached. Mirroring the encoder's operation, this design algorithm is non-increasing in the (training set) cost J.

Our results, shown in Figure 3, are for the 512×512 Lena image. We have evaluated performance both for i) 1D source dependencies (based on zig-zag scan order) and ii) 2D source dependencies, as indicated for a transformed block in Figure 2. For i), our method uses $\{\mathrm{Prob}[\mathrm{i}_{\mathrm{mn}}|\mathrm{i}_{\mathrm{m,n-1}}, \mathrm{i}_{\mathrm{m-1,n}}]\}$ and iteratively encodes sources in zig-zag scan order. For ii), our method uses $\{\mathrm{Prob}[\mathrm{i}_{(\mathrm{x,y}),\mathrm{n}}|\mathrm{i}_{(\mathrm{x,y}),\mathrm{n-1}}, \mathrm{i}_{(\mathrm{x-1,y}),\mathrm{n}}, \mathrm{i}_{(\mathrm{x,y-1}),\mathrm{n}}]\}$[4]. In this case, we iteratively encode sources row-by-row, from left-to-right within each row (see Figure 1).

The performance curves were obtained by designing the various coders for a sequence of λ values, to sweep out a rate/distortion curve. Note that the new CECTCQ encoder achieves gains (at selective rates) greater than 1.5 dB over greedy CECTCQ and greater than 3.0 dB over standard ECTCQ.

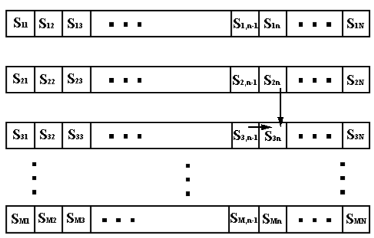

Fig. 1. Source $\underline{s_m}$ is represented by the m^{th} row; arrows show assumed statistical dependencies both within and between sources.

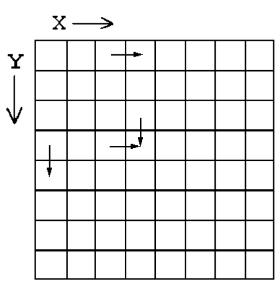

Fig. 2. A transformed block, with arrows showing 2-D dependencies between sources. For sources in the first row and first column, only 1-D dependency is used. Note that the *intra-source* dependency is not shown in this figure.

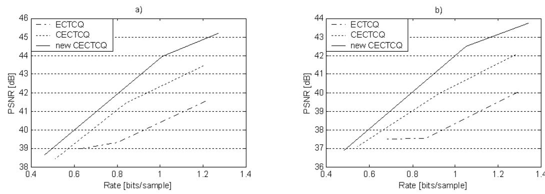

Fig. 3. PSNR vs Rate result on Lena: a) 2-D case, with Lena outside the training set. b) 1-D case, with Lena outside the training set.

4. REFERENCES

[1] J. M. Shapiro, "Embedded Image Coding Using Zerotrees of Wavelet Coefficients," *IEEE Trans. on Signal Processing*, Vol. 41, No. 12, pp. 3445-3462, December 1993.

[2] T. R. Fischer and M. Wang, "Entropy-Constrained Trellis-Coded Quantization," *IEEE Trans. on Info. Theory*, Vol. IT-38, No. 2, pp. 415-426, March 1992.

[3] S. J. Lee, and C. W. Lee, "Conditional Entropy-Constrained Trellis-Searched Vector Quantization for Image Compression," *Signal Processing: Image Communication*, Vol. 8, pp. 99-104, 1996.

[4] F. Kossentini, W. C. Chung, and M. J. T. Smith, "Conditional Entropy-Constrained Residual VQ with Application to Image Coding," *IEEE Trans. on Image Processing*, vol. 5, No. 2, pp. 311-320, February 1996.

[5] P. A. Chou, and T. Lookabaugh, "Conditional Entropy- Constrained Vector Quantization of Linear Predictive Coefficients," *ICASSP*, Vol. 1, pp. 197-200, April 1990.

[6] M. Lightstone, D. Miller, and S. K. Mitra, "Entropy-Constrained Product Code Vector Quantization with Application to Image Coding," *IEEE International Conference on Image Processing*, pp. 623-627, November 1994.

[7] M. W. Marcellin, and T. R. Fischer, "Trellis Coded Quantization of Memoryless and Gauss-Markov Sources," *IEEE Trans. on Communications*, Vol. 38, No. 1, pp. 82-93, January 1990.

[8] L. C. Stewart, R. M. Gray, and Y. Linde, "The Design of Trellis waveform coders," *IEEE Trans. on Communication*, Vol. 30, pp. 702-710, April 1982.

[9] M. W. Marcellin, "On Entropy-Constrained Trellis Coded Quantization," *IEEE Trans. on Communications*, Vol. 42, No. 1, pp. 14-16, January 1994.

[4]Here, $i_{(x,y),n}$ is the index of the n^{th} sample for the source at spatial frequency (x, y).