OPTIMAL FRAMEWORK FOR LOW BIT-RATE BLOCK CODERS

Yaakov Tsaig, Michael Elad, Gene H. Golub

SCCM Program
Computer Science Department
Stanford University
Stanford, CA 94305, USA

Peyman Milanfar

Baskin School of Engineering
237 Baskin Engineering Bldg.
University of California
1156 High Street, Santa Cruz, CA 95064

ABSTRACT

Block coders are among the most common compression tools available for still images and video sequences. Their low computational complexity along with their good performance make them a popular choice for compression of natural images. Yet, at low bit-rates, block coders introduce visually annoying artifacts into the image. One approach that alleviates this problem is to downsample the image, apply the coding algorithm, and interpolate back to the original resolution. In this paper, we consider the use of optimal decimation and interpolation filters in this scheme. We first consider only optimization of the interpolation filter, by formulating the problem as least-squares minimization. We then consider the joint optimization over both the decimation and the interpolation filters, using the *Variable Projection* method. The experimental results presented clearly exhibit a significant improvement over other approaches.

1. INTRODUCTION

Block-transform coders are among the most common compression tools available for still images and video sequences. These coders divide the image to non-overlapping square blocks and apply a transform on each block. Among the available transforms, the DCT is the most widely adopted as it exhibits very good energy compaction and decorrelation properties. The low complexity of block-based methodology along with its good performance make it the prominent choice for image compression. Both the JPEG standard for still image compression [1] and the MPEG standards for compression of video sequences [2, 3] rely on block-based compression.

Yet, at low bit rates images compressed with block coders exhibit visually disturbing phenomena, known as *blocking artifacts*. These are characterized by visually noticeable changes in pixel values along block boundaries. Various post-processing techniques have been suggested for the reduction of blocking artifacts (see [4] for an extensive survey), but they often introduce excessive blurring, ringing, and in many cases produce poor deblocking results at certain areas of the image.

In [5], Bruckstein *et al.* considered downsampling an image before applying the JPEG coding algorithm, and interpolating at the decoder stage to obtain the image in full resolution. This method has several attractive properties. First and foremost, at low bit-rates there is a marked gain in performance, both in terms of PSNR and in terms of visual quality. Second, it substantially reduces the computational complexity involved in coding/decoding, since the input to the JPEG encoder is considerably smaller in size. In addition, the range of low bit-rates is expanded, allowing us to compress an image at lower bit-rates than possible when using JPEG directly. Finally, since the method does not change the coding algorithm, it can be used in applications where the JPEG codec is already implemented without making substantial modifications.

Motivated by these features, Bruckstein *et al.* [5] went on to analytically derive a model for the JPEG encoder in order to obtain an optimality criterion on the downsampling factor for a given input image. Throughout their experiments, they used fixed filters for decimation and interpolation, and did not consider the effect of different filters on the quality of the results. In this paper, we take a more general point of view, and consider the use of optimal filters for the decimation and interpolation stages in order to achieve better performance. We will show that optimizing over each of the filters separately, and on both of the filters jointly, results in a significant gain in performance, both visually and quantitatively. Unlike [5], we do not derive a model for the encoder, but consider it to be a "black box". Therefore, our derivations are not restricted to the JPEG mechanism, and can be applied to other coders as well.

In section 2, we shall present the algorithms for finding the optimal filters. We first consider the optimization of the interpolation filter, followed by the joint optimization of both filters. Experimental results for both cases are also presented in this section. Section 3 discusses the results and has some concluding remarks.

2. OPTIMAL FRAMEWORK

Throughout this section, we consider the system shown in Fig. 1. An input image is convolved with a linear filter $\mathbf{f}$, and then downsampled by a factor k. The low-resolution image is then encoded using a block coder. At the decoder, the image is first decoded using the block decoder, then upsampled back to its original resolution and filtered by a filter $\mathbf{g}$ to produce the reconstructed result. The authors in [5] took $\mathbf{f}$ to be a standard anti-aliasing filter, and $\mathbf{g}$ to be a linear interpolation kernel. The sampling factor k was chosen according to their analytical predictions. Throughout this paper, we will assume, for simplicity, $k = 2$. For a thorough discussion on choosing the optimal k, the reader is referred to [5].

2.1. Optimal Interpolation Filter

Let X denote the input image, of size $m \times n$, Y denote the image after decimation (size $\frac{m}{2} \times \frac{n}{2}$), and $\tilde{X}$ the reconstructed image

This work was supported in part by the National Science Foundation Grants *CCR-9984246* and *CCR-9971010*

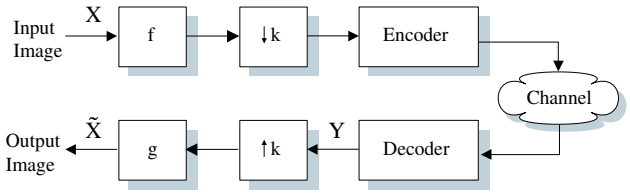

Fig. 1: Sampling/filtering scheme for block coders.

after interpolation, as in Fig. 1. Our ultimate goal is to minimize the L^2 error norm $\|\tilde{X} - X\|_2$. At first, we shall only consider the optimization of the interpolation filter $\mathbf{g}$, while keeping $\mathbf{f}$ fixed. To do so, we note that the interpolation stage can be equivalently expressed as matrix multiplication. In our formulation below, we consider the upsampling and filtering steps as a unified process and consequently use a set of filters, rather than first inserting zeros and then using one filter for interpolation.

Specifically, we define $\tilde{X}^{(p,q)}, p, q \in \{0, 1\}$, such that

$$\tilde{X}^{(p,q)}(i, j) = \tilde{X}(2i + p, 2j + q)$$

i.e., $\tilde{X}^{(p,q)}$ is $\tilde{X}$ shifted by (p, q) and then downsampled by 2. Clearly, the set $\{\tilde{X}^{(p,q)}, p, q \in \{0, 1\}\}$ is just a reordered version of $\tilde{X}$. In addition, we let $\tilde{\mathbf{x}}^{(p,q)}$ denote the row-stacked form of $\tilde{X}^{(p,q)}$, i.e. $\tilde{\mathbf{x}}$ has length $mn/4$, with elements

$$\tilde{\mathbf{x}}^{(p,q)}(i \times \frac{n}{2} + j) = \tilde{X}^{(p,q)}(i, j)$$

Now, for our 2-D interpolation filter $\mathbf{g}$ with dimensions $l \times l$, we similarly define $\left\{\mathbf{g}^{(p,q)}, 0 \leq p, q \leq 1\right\}$, where each $\mathbf{g}^{(p,q)}$ is a vector of length l^2 that represents the filter in the filter set which produces $\tilde{X}^{(p,q)}$ by filtering Y. Finally, we construct a matrix $\mathbf{\Phi}$ with dimensions $mn \times l^2$ out of the image Y, of the form

$$\mathbf{\Phi} = \begin{bmatrix} \phi_{0,0}^T \\ \phi_{0,1}^T \\ \vdots \\ \phi_{m-1,n-1}^T \end{bmatrix} \tag{1}$$

where $\phi_{i,j}$ is the row-stacked form of an $l \times l$ window, centered around the pixel location (i, j) of Y.

Using these definitions, we can now express each $\tilde{\mathbf{x}}^{(p,q)}$ (and equivalently $\tilde{X}$) as a product of the matrix Φ and the filter $\mathbf{g}^{(p,q)}$ in vector form,

$$\tilde{\mathbf{x}}^{(p,q)} = \mathbf{\Phi}\mathbf{g}^{(p,q)} \tag{2}$$

Looking at Eq. (2), we can immediately see that an optimal solution (in the least-squares sense) is obtained by setting $\tilde{X} = X$ and minimizing over all $\mathbf{g}^{(p,q)}$. Specifically, we solve

$$\min_{\mathbf{g}^{(p,q)}} \|\mathbf{x}^{(p,q)} - \mathbf{\Phi}\mathbf{g}^{(p,q)}\|_2^2 \tag{3}$$

This is a linear least squares (LS) problem, with the solution $\mathbf{g}^{(p,q)}$ given by

$$\mathbf{g}^{(p,q)} = \mathbf{\Phi}^{+}\mathbf{x}^{(p,q)} \tag{4}$$

For practical purposes, one can avoid constructing the matrix $\mathbf{\Phi}$ and its pseudo-inverse $\mathbf{\Phi}^{+}$ by applying recursive least squares (RLS), or any of its block forms.

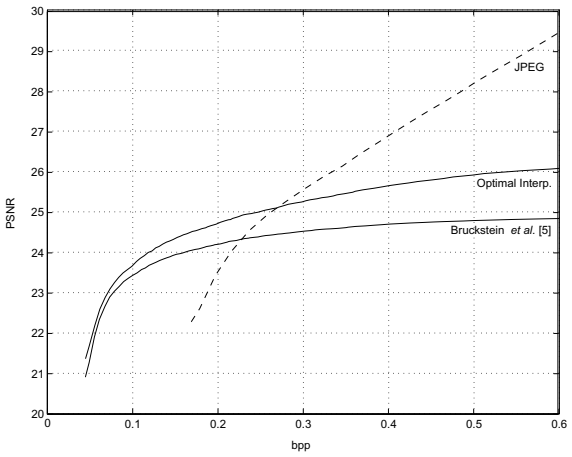

Fig. 2: Optimal Interpolation for *Barbara*.

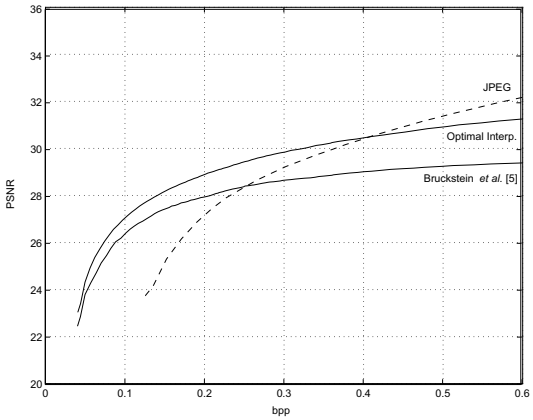

Fig. 3: Optimal Interpolation for *Goldhill*.

We applied this optimization algorithm to two of the JPEG standard test images. Figs. 2,3 show the rate distortion curves obtained for the images *Barbara* and *Goldhill*, respectively. For the decimation filter, we used the same anti-aliasing filter used in [5]. For the interpolation filter, we set $l = 5$, and used Eq. (4) to find the optimal $\mathbf{g}^{(p,q)}$. The results clearly display a significant gain in performance over the original results in [5]. We see that the optimal curve intersects the JPEG curve much later than the curve obtained by Bruckstein *et al.* This essentially means that our algorithm is applicable to a wider range of bit-rates, since it performs better than direct JPEG compression up to a higher bit-rate. The visual improvement over [5] is clearly evident in Fig. 4 for *Goldhill*. We can see that the optimal filter provides a significantly sharper image, while virtually eliminating the blockiness. We note that the overhead of sending the filter coefficients to the decoder is not included in the rate calculation, yet it is in the order of 200 bits, which is negligible even for very low bit rates.

(a)

(b)

(c)

Fig. 4: Compression results for *Goldhill*, 0.2bpp: (a) JPEG, PSNR = 27.43dB (b) Bruckstein *et al.*, PSNR = 27.95dB (c) Optimal Interpolation, PSNR = 28.91dB.

2.2. Optimal Decimation Filter

Inspired by the results obtained when optimizing over the interpolation filter, we now turn our attention to the decimation filter. If we consider minimizing the difference between X and $\tilde{X}$ over both $\mathbf{f}$ and $\mathbf{g}$, then the image Y at the output of the decoder is no longer fixed, hence our matrix $\mathbf{\Phi}$ of Eq. (1) which originated from Y is now dependent on $\mathbf{f}$, i.e. $\mathbf{\Phi} = \mathbf{\Phi}(\mathbf{f})$. Nonetheless, we can still write our problem as a matrix system in the form of Eq. (2),

$$\tilde{\mathbf{x}}^{(p,q)} = \mathbf{\Phi}(\mathbf{f})\mathbf{g}^{(p,q)} \tag{5}$$

and our minimization problem (3) now becomes

$$\min_{\mathbf{f},\mathbf{g}^{(p,q)}} \|\mathbf{x}^{(p,q)} - \mathbf{\Phi}(\mathbf{f})\mathbf{g}^{(p,q)}\|_2^2 \tag{6}$$

This is a non-linear LS problem with respect to the variables $\mathbf{f}, \mathbf{g}^{(p,q)}$. To find a solution for this problem, we apply the *Variable Projection* (VP) method [6]. This method uses the fact that our LS problem has two separable sets of variables, namely $\mathbf{f}$ and $\mathbf{g}^{(p,q)}$, and the dependence on $\mathbf{g}^{(p,q)}$ is linear. More specifically, assume for the moment we know the optimal $\mathbf{f}$. If we plug this $\mathbf{f}$ into Eq. (5), then $\mathbf{\Phi}(\mathbf{f})$ is now fixed, hence we again face a linear LS problem, with the solution readily given by

$$\mathbf{g}^{(p,q)} = \mathbf{\Phi}(\mathbf{f})^{+}\mathbf{x}^{(p,q)} \tag{7}$$

Now, we can use this expression for $\mathbf{g}^{(p,q)}$ in our minimization problem (6), leading to

$$\min_{\mathbf{f}} \|x^{(p,q)} - \mathbf{\Phi}(\mathbf{f})\mathbf{\Phi}(\mathbf{f})^{+}\mathbf{x}^{(p,q)}\|_2^2 = \min_{\mathbf{f}} \|\mathbf{P}^{\perp}_{\mathbf{\Phi}(\mathbf{f})}\mathbf{x}^{(p,q)}\|_2^2 \tag{8}$$

where $\mathbf{P}^{\perp}_{\mathbf{\Phi}(\mathbf{f})} \equiv I - \mathbf{\Phi}(\mathbf{f})\mathbf{\Phi}(\mathbf{f})^{+}$ is the projector on the orthogonal complement of the column space of $\mathbf{\Phi}(\mathbf{f})$.

As we can see, by using VP, we have essentially eliminated the minimization with respect to $\mathbf{g}^{(p,q)}$, and we are left with a non-linear LS problem with respect to the decimation filter $\mathbf{f}$. This is still a difficult task, due to the non-linearity of the problem at hand. For the purpose of validating our idea and demonstrating its performance, we shall restrict our discussion to a fraction of the parameter space for this optimization problem. We notice that the parameter space for this problem is l-dimensional, where l is the length of the filter $\mathbf{f}$ (assuming separability of $\mathbf{f}$). Rather than considering the entire space, we consider a family of lowpass filters with a varying cutoff frequency. For the design of the lowpass filter $\mathbf{f}_{LP}(\omega)$, we use the windowing method with a Hamming window to design a separable filter with cutoff frequency ω. Consequently, a sub-optimal solution of (5) is given by the lowpass filter $\mathbf{f}_{LP}(\hat{\omega})$, where $\hat{\omega}$ is given by

$$\hat{\omega} = \arg\min_{\omega} \|\mathbf{P}^{\perp}_{\mathbf{\Phi}(\mathbf{f}_{\mathbf{LP}}(\omega))}\mathbf{x}^{(p,q)}\|_2^2 \tag{9}$$

This univariate problem can be easily solved with various methods.

Fig. 5 shows the rate-distortion curve obtained when applying this algorithm on the test image *Barbara*. Clearly, optimizing over both filters results in significantly better quality compared to optimization of the interpolation filter alone. This is also inherent in Fig. 6, which displays the visual results for a bit rate of 0.2bpp. The image obtained from the joint optimization is sharper and exhibits more details, and is free of any blocking artifacts. Interestingly enough, the optimal decimation filter found in this case is a lowpass filter with cutoff frequency $\omega = 0.97$, which is essentially an identity filter. Intuitively, this means that we do not want any filtering prior to downsampling, in order to preserve the texture that dominates *Barbara*.

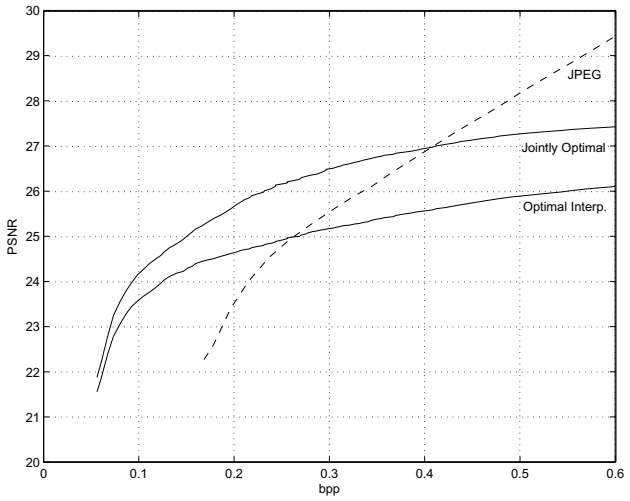

Fig. 5: Optimal Interpolation & Decimation for *Barbara*.

3. DISCUSSION AND CONCLUSIONS

In this paper, we presented an optimal framework for improving the low bit-rate performance of block coders. By carefully selecting the filters used in the decimation and interpolation steps to be optimal, we have achieved a significant gain in performance, compared to using common filters. We demonstrated that the optimal framework outperforms JPEG for a wider range of bit-rates, making it applicable in more diverse situations.

Current work in our research is focused in extending the idea to spatially adaptive filters. By allowing different filters for different areas in the image, we can gain further improvement over shift-invariant filters. For instance, we can use one set of filters for block boundaries, and another for block contents.

4. REFERENCES

[1] ISO/IEC JTC1 Committee Draft 10918-1, "Digital compression and coding of continuous-tone still images, part 1, requirements and guidelines," February 1991.

[2] ISO/IEC 11172, "Information technology - coding of moving pictures and associated audio for digital storage media at up to about 1.5 mbit/s.," 1993.

[3] ISO/IEC 13818, "Information technology - generic coding of moving pictures and associated audio information.," 1994.

[4] M.-Y. Shen and C. C. Jay Kuo, "Review of postprocessing techniques for compression artifacts removal," *Journal of Visual Communication and Image Representation*, vol. 9, no. 1, pp. 2–14, March 1998.

[5] M. Elad A. M. Bruckstein and R. Kimmel, "Down-sampling for better transform compression," *IEEE Trans. on Image Process.*, to appear.

[6] G. H. Golub and V. Pereyra, "The differentiation of pseudoinverses and nonlinear least squares problems whose variables separate," *SIAM J. Numer. Anal.*, vol. 10, pp. 413–432, 1973.

(a)

(b)

(c)

Fig. 6: Compression results for *Barbara*, 0.2bpp: (a) JPEG, PSNR = 23.42dB (b) Optimal Interpolation, PSNR = 24.74dB (c) Jointly optimal, PSNR = 25.5dB.