Variable Projection for Near-Optimal Filtering in Low Bit-Rate Block Coders

Yaakov Tsaig, *Member, IEEE*, Michael Elad, Peyman Milanfar, *Senior Member, IEEE*, and Gene H. Golub

***Abstract*—Recent work on block-based compression for low bit-rate coding has shown that employing a block coder within a sampling scheme where the image is downsampled prior to coding (and upsampled after the decoding stage) results in superior performance compared to standard block coding. In this paper, we explore the use of optimal decimation and interpolation filters in this coding scheme. We show that the problem of finding optimal filters for a general, unknown, "black-box" coder can be written as a separable least squares problem in two sets of variables. We then elegantly solve this optimization problem using the *Variable Projection* method. The experimental results presented clearly exhibit a significant improvement over existing approaches.**

***Index Terms*—Block coders, deblocking, separable least squares, variable projection.**

I. Introduction

Block-based coders, such as the JPEG standard for still image compression [1] and the MPEG-1 [2] and MPEG-2 [3] standards for coding of video sequences, are among the most common compression tools used for compression of visual data. Their low computational complexity along with their good performance make them a very attractive choice for many applications. A well-known drawback of block coding schemes is the introduction of visually disturbing blocking artifacts in the reconstructed image. As block-based coders became increasingly more popular, the need to diminish the effects of blocking artifacts grew stronger. A great deal of effort has been invested in attempts to reduce these artifacts, while preserving the information content of the image.

Various postprocessing techniques have been suggested for the reduction of blocking artifacts, based on different approaches. Among these we count techniques based on adaptive filtering [4], [5], projection on convex sets (POCS) [6]–[8], Markov random field modeling [9], [10], and many others (see [11] for a review). Most of these methods work well when the image is compressed at high or medium bit rates, but the majority of them fail to produce satisfactory results when applied on images compressed at low bit rates.

In search of a more suitable scheme for low bit-rate block coders, several authors considered the idea of downsampling before coding and subsequent upsampling at the decoder stage. Indeed, in [12], for instance, 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. Earlier, Zeng and Venetsanopoulos [13] considered an interpolative image coding scheme based on these notions, and Jung, Mitra, and Mukherjee in [14] introduced the idea of "subband DCT," which they later used for image resizing in the compressed domain [15]. Most recently, Dugad and Ahuja [16] proposed an elegant scheme for spatial scalability using nonscalable coders that again makes use of the down/up-sampling methodology.

These approaches have 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, the computational complexity involved in coding/decoding is substantially reduced, since the input to the encoder is considerably smaller in size. In addition, the range of low bit rates is expanded, allowing compression at lower bit rates. Finally, since most of these methods do not modify the basic coding algorithm, but only apply preprocessing/postprocessing, they can be used in applications where the codec is already implemented without introducing substantial modifications.

Motivated by these features, the authors of [12] specifically derived an analytical model for the JPEG encoder in order to obtain an optimality criterion on the sampling 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. To the best of our knowledge, all of the papers that have taken the same approach used fixed, standard filters in the down/up-sampling process.

The main contribution of this paper is to address the issue of finding optimal filters in practice for the decimation and interpolation stages in order to achieve better performance. We shall see that the use of carefully chosen filters, based on least squares (LS) optimality, results in a significant gain in performance, both visually and quantitatively. We do not derive a specific model for the encoder, but consider it to be a "black box." We view this as an advantage, rather than a shortcoming because under this framework, the results of this paper are not restricted solely to a particular block coding scheme, and can be applied to other coders as well.

The rest of this letter is organized as follows. Section II presents the algorithm for finding the optimal interpolation filter, by formulating the problem as linear LS minimization. Section III addresses the problem of jointly optimizing both filters, which we formulate as a nonlinear, but separable LS problem. We apply a mechanism known as *variable projection* to reduce this problem into a nonlinear LS problem in one set of variables, and then demonstrate that even a suboptimal

Manuscript received January 5, 2004; revised March 26, 2004. This work was supported in part by the National Science Foundation under Grant CCR-9984246 and Grant CCR-9971010. The work of P. Milanfar was also supported in part by U.S. Air Force under Grant AFOSR F49620-03SC20030835. This paper was recommended by Associate Editor H. Chen.

Y. Tsaig and G. H. Golub are with the Department of Computer Science, Stanford University, Stanford, CA 94305-9025 USA (e-mail: tsaig@stanford.edu; golub@sccm.stanford.edu).

M. Elad is with The Computer Science Department, The Technion–Israel Institute of Technology, Haifa 32000, Israel (e-mail: elad @ cs.technion.ac.il).

P. Milanfar is with the Electrical Engineering Department, Baskin School of Engineering, University of California, Santa Cruz, CA 95064 USA (e-mail: milanfar@ee.ucsc.edu).

Digital Object Identifier 10.1109/TCSVT.2004.839980

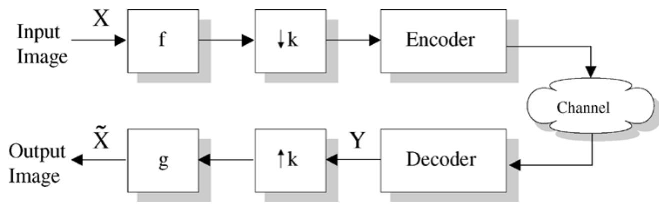

Fig. 1. Sampling/filtering scheme for block coders.

approach results in superior performance. In Section IV we provide a discussion of the results, summarize our findings and make some concluding remarks.

II. Optimal Interpolation by Linear Least Squares

Throughout the rest of this paper, 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 [12] took $\mathbf{f}$ to be a standard antialiasing filter, and $\mathbf{g}$ to be a linear interpolation kernel (hat function). 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 [12].

Let X denote the input image of size $m \times n$, Y denote the image after decimation (size $m/2 \times n/2$), and $\tilde{X}$ the reconstructed image 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., each $\tilde{\mathbf{x}}^{(p,q)}$ has length $mn/4$, with elements

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

Now, for our two-dimensional (2-D) interpolation filter $\mathbf{g}$ with dimensions $l \times l$, we similarly define $\{\mathbf{g}^{(p,q)}, 0 \le p, q \le 1\}$, 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,

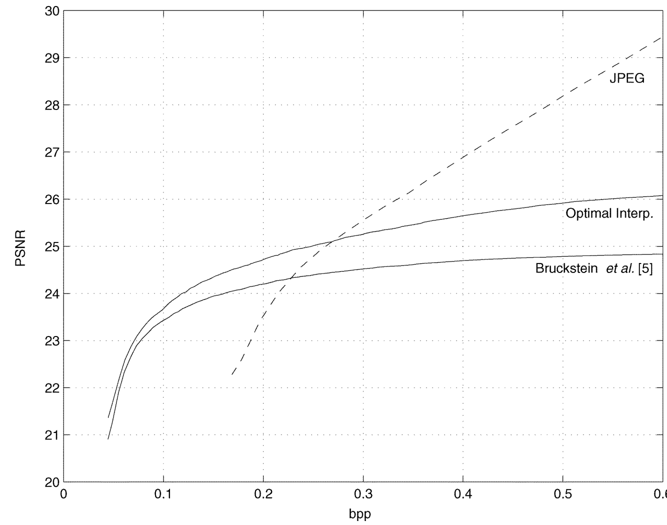

Fig. 2. Optimal interpolation for *Barbara*.

we construct a matrix $\mathbf{\Phi}$ with dimensions $mn/4 \times l^2$ out of the image Y, of the form

$$\mathbf{\Phi} = \begin{bmatrix} \boldsymbol{\phi}_{0,0}^T \\ \boldsymbol{\phi}_{0,1}^T \\ \vdots \\ \boldsymbol{\phi}_{(m/2)-1,(n/2)-1}^T \end{bmatrix} \tag{1}$$

where $\boldsymbol{\phi}_{i,j}$ 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 $\mathbf{\Phi}$ 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 (2), we can immediately see that an optimal solution (in the LS sense) is obtained by setting $\tilde{X} = X$ and minimizing over all $\mathbf{g}^{(p,q)}$. Specifically, we solve

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

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

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

where $\mathbf{\Phi}^{+}$ denotes the pseudo-inverse of $\mathbf{\Phi}$ [17]. Obviously, in practice, there is no need to construct the matrix $\mathbf{\Phi}$ nor its pseudo-inverse $\mathbf{\Phi}^{+}$. Rather, (3) is solved by applying recursive least squares (RLS).

To test the performance of this method, we used a standard JPEG coder as our block coder and applied this optimization algorithm to several standard test images. We also compared our results with those of Bruckstein *et al.* [12]. Figs. 2–4 show the rate-distortion curves obtained for the images *Barbara*, *Goldhill*, and *Boats*, respectively. For the decimation filter, we used an 11-tap antialiasing filter, designed using a Hamming window and normalized cutoff frequency $\omega_n = 0.5$. For the interpolation filter, we set $l = 5$, and solved (4) to find the optimal $\mathbf{g}^{(p,q)}$. The results clearly display a significant gain in performance over

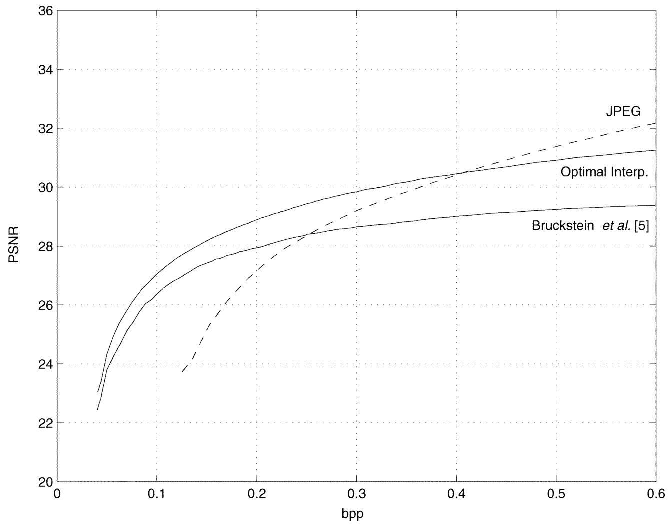

Fig. 3. Optimal interpolation for *Goldhill*.

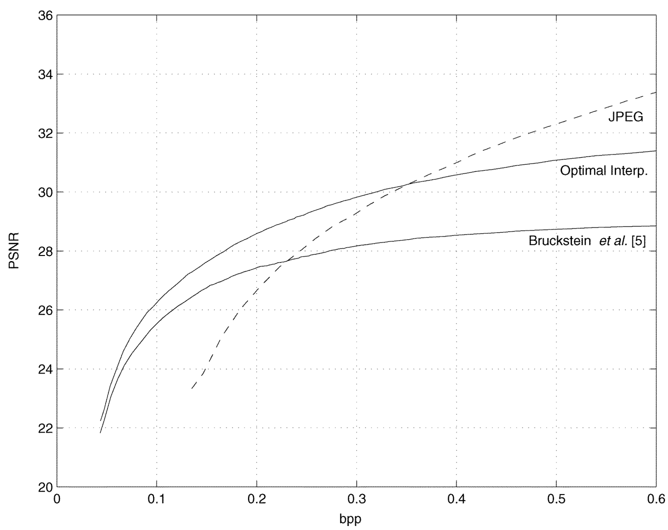

Fig. 4. Optimal interpolation for *Boats*.

Fig. 5. Compression results for *Barbara*, 0.2 bpp. (a) JPEG, $\mathrm{PSNR} = 23.42$ dB. (b) Bruckstein *et al.*, $\mathrm{PSNR} = 24.19$ dB. (c) Optimal interpolation, $\mathrm{PSNR} = 24.74$ dB.

the original results in [12]. We see that the optimal curve intersects the JPEG curve at a higher bit rate 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 [12] is clearly evident in Figs. 5–7. We can see that the optimal filter provides a significantly sharper image, while virtually eliminating the blockiness. In terms of PSNR, in all three cases, when coded with 0.2 bpp, the gain over standard JPEG is around 1–1.5 dB.

It would be interesting to examine the filters obtained in the optimization process. Fig. 8 shows the 2-D frequency response of the optimal interpolation filters obtained for *Barbara* (recall that 4 filters are needed). As we can see, all four filters exhibit a low-pass behavior with some high-pass components. In other words, the filters attempt to smooth out the blockiness while preserving edge and texture information.

The overhead of sending the filter coefficients to the decoder is not included in the rate calculation, yet it is in the order of a few hundred bits per filter, which is negligible even for very low bit rates (we found that for the parameters used in our experiments, each filter would occupy around 200 bits when properly entropy coded). We would like to comment that the 4-filters approach we have taken is completely equivalent to a single filter approach, in which case zero-padding is used prior to filtering, and only one vector $\mathbf{x}$ is constructed. Indeed, it is not hard to see that the LS problem $\mathbf{\Phi g} = \mathbf{x}$, with $\mathbf{\Phi}$ a row-stacked

Fig. 6. Compression results for *Goldhill*, 0.2 bpp. (a) JPEG, $\mathrm{PSNR} = 27.43$ dB. (b) Bruckstein *et al.*, $\mathrm{PSNR} = 27.95$ dB. (c) Optimal interpolation, $\mathrm{PSNR} = 28.91$ dB.

Fig. 7. Compression results for *Boats*, 0.2bpp. (a) JPEG, $\mathrm{PSNR} = 26.82$ dB. (b) Bruckstein *et al.*, $\mathrm{PSNR} = 27.42$ dB. (c) Optimal interpolation, $\mathrm{PSNR} = 28.58$ dB.

form of the image Y after zero-padding, can be decoupled into four subproblems of the form of (3). The advantage of using the 4-filters approach is apparent when the matrix $\mathbf{\Phi}$ needs be constructed explicitly, in which case it is four times smaller than if zero-padding was used. However, in instances where the sampling factor is greater than two, or when it is a rational number, we note that such an approach may not be feasible, and so the problem should be formulated using a single filter.

III. Optimal Decimation and Interpolation by Variable Projection

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 (1) which origi-

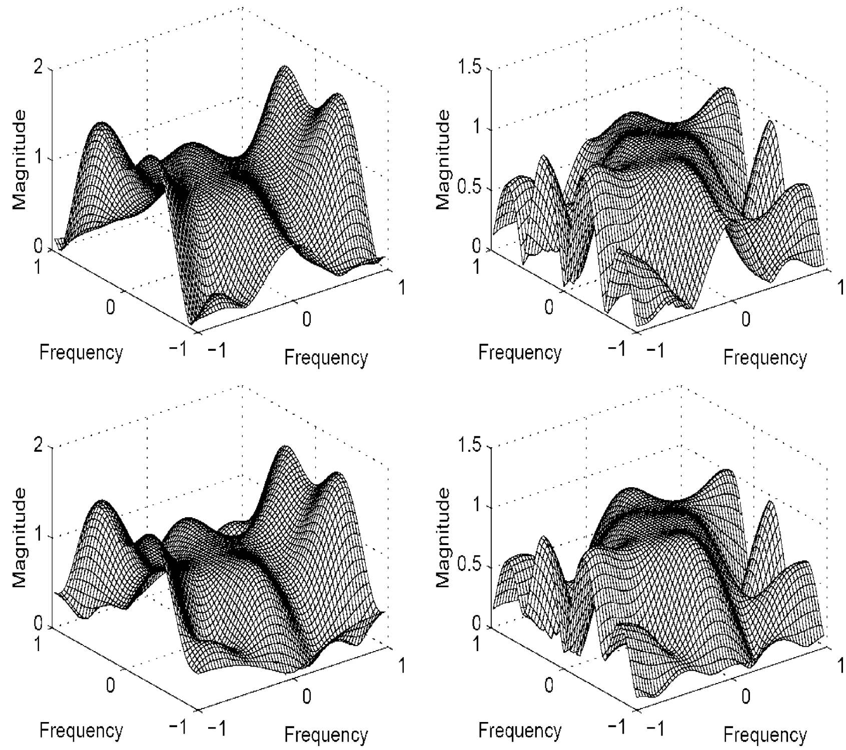

Fig. 8. Optimal interpolation filters obtained for *Barbara*.

nated from Y is now dependent on $\mathbf{f}$, i.e., $\boldsymbol{\Phi} = \boldsymbol{\Phi}(\mathbf{f})$. Nonetheless, we can still write our problem as a matrix system in the form of (2)

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

and our minimization problem (3) now becomes

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

This is a nonlinear LS problem with respect to the variables $\mathbf{f}, \mathbf{g}^{(p,q)}$. To find a solution for this problem, we apply Golub and Pereyra's *Variable Projection* method [18]. This method is applicable when dealing with separable nonlinear LS problems, i.e., problems for which the model function is a linear combination of nonlinear functions. Since first published, the variable projection method (sometimes coined VARPRO) has been applied in a variety of different fields in engineering and scientific computing (see [19] for an extensive survey).

In our formulation, we have two separable sets of variables, namely $\mathbf{f}$ and $\mathbf{g}^{(p,q)}$. Moreover, the dependence on $\mathbf{g}^{(p,q)}$ is linear. Hence, we can use VARPRO to transform the problem into a nonlinear LS problem in one set of variables. More specifically, assume for the moment we know the optimal $\mathbf{f}$. If we plug this $\mathbf{f}$ into (5), then $\boldsymbol{\Phi}(\mathbf{f})$ is now fixed, hence we again face a linear LS problem, with the solution readily given by

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

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

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

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

As we can see, by using VARPRO, we have essentially eliminated the minimization with respect to $\mathbf{g}^{(p,q)}$, and we are left with a nonlinear LS problem with respect to the decimation filter $\mathbf{f}$. Evidently, this procedure not only reduces the dimension of the parameter space but also results in a better-conditioned problem (see [19] for details). Nonetheless, solving (8) is still a difficult task, due to the nonlinearity of the problem at hand. For our purposes, we shall pursue a near-optimal approach, by restricting 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 window method with a Hamming window to design a separable filter with cutoff frequency ω. Consequently, a suboptimal solution of (5) is given by the lowpass filter $\mathbf{f}_{LP}(\hat{\omega})$, where $\hat{\omega}$ is given by

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

This is a univariate minimization problem, which we solve using Brent's classic algorithm for minimization without derivatives

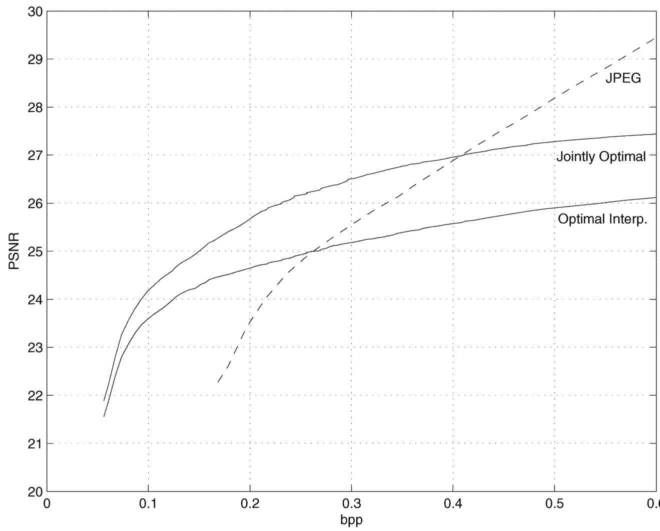

Fig. 9. Optimal interpolation and decimation for *Barbara*.

Fig. 10. Optimal decimation and interpolation for *Barbara*, 0.2 bpp, $\mathrm{PSNR} = 25.5$ dB.

[20]. We note that the optimization problem (9) is solved once at the encoding stage, and the resulting decimation filter need not be sent to the decoder as it is not used in the decoding process. Hence, no additional overhead is introduced by this stage.

Before examining the experimental results, we shall comment about the computational burden involved in each iteration of Brent's algorithm. Each such iteration requires coding the image after applying the filter $\mathbf{f}_{LP}(\omega)$, followed by a decoding stage and computation of the residual in (9). In our experiments on typical 512×512 images, the execution time per iteration was 3.92 s, when executed on a P4 2-GHz machine, using Matlab. As mentioned above, the solution of (9) is carried out once during the encoding stage, and no further computations need be done at the decoder.

Fig. 9 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. 10, which displays the visual results for a bit rate of 0.2 bpp (compare Fig. 5). 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 low-pass 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*.

IV. Discussion and Conclusion

This letter presented an algorithm for finding optimal filters in a sampling/compression scheme. By a LS argument, we have shown that the use of optimal interpolation filters achieves a significant gain in performance, compared to using common filters. We demonstrated that a near optimal approach for finding the decimation filter also results in a marked improvement in quality. In addition, by using optimal filters, we outperform JPEG for a wider range of bit rates compared to [12], making this sampling/compression scheme applicable in more diverse situations.

We note that even greater improvements can be gained by tailoring the filter optimization to the block structure of the coder. By using one set of filters for block boundaries, and another for block interior, we are likely to get better performance, as there is much to be gained by treating these two regions separately. Furthermore, we note that the restriction of the parameter space when solving (9) to separable low-pass filters is quite stringent. Better performance may result by considering a wider variety of decimation filters, while keeping the optimization problem tractable. These issues are left for future research.

References

[1] *Digital Compression and Coding of Continuous-Tone Still Images, Part 1, Requirements and Guidelines*, ISO/IEC JTC1 Committee Draft 10 918-1 Std., Feb. 1991.

[2] *Information Technology – Coding of Moving Pictures and Associated Audio for Digital Storage Media at up to About 1.5 mbit/s*, ISO/IEC 11 172 Std., 1993.

[3] *Information Technology – Generic Coding of Moving Pictures and Associated Audio Information*, ISO/IEC 13 818 Std., 1994.

[4] C. J. Kuo and R. J. Hsieh, "Adaptive postprocessor for block encoded images," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 5, no. 8, pp. 298–304, Aug. 1995.

[5] J. Chou, M. Crouse, and K. Ramchandran, "A simple algorithm for removing blocking artifacts in block-transform coded images," *IEEE Signal Process. Lett.*, vol. 5, no. 1, pp. 33–35, Feb. 1998.

[6] D. Yula and H. Webb, "Image restoration by the method of convex projections: Part I – Theory," *IEEE Trans. Med. Imag.*, vol. MI-1, no. 6, pp. 81–94, Oct. 1982.

[7] A. Zakhor, "Iterative procedures for reduction of blocking effects in transform image coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 2, no. 2, pp. 91–95, Mar. 1992.

[8] Y. Yang, N. P. Galatsanos, and A. K. Katsaggelos, "Regularized reconstruction to reduce blocking artifacts of block discrete cosine transform compressed images," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 3, no. 12, pp. 421–432, Dec. 1993.

[9] T. P. O'Rourke and R. L. Stevenson, "Improved image decompression for reduced transform coding artifacts," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 6, pp. 490–499, 6 1995.

[10] T. Meier, K. N. Ngan, and G. Crebbin, "Reduction of blocking artifacts in image and video coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 9, no. 4, pp. 490–495, Apr. 1999.

[11] M.-Y. Shen and C. C. J. Kuo, "Review of postprocessing techniques for compression artifacts removal," *J. Vis. Commun. Image Represent.*, vol. 9, no. 1, pp. 2–14, Mar. 1998.

[12] A. M. Bruckstein, M. Elad, and R. Kimmel, "Down-sampling for better transform compression," *IEEE Trans. Image Process.*, vol. 12, no. 9, pp. 1132–1144, Sep. 2003.

[13] B. Zeng and A. N. Venetsanopoulos, "A JPEG-based interpolative image coding scheme," in *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing*, vol. V, Minneapolis, MN, Apr. 1993, pp. 393–396.

[14] S. H. Jung, S. K. Mitra, and D. Mukherjee, "Subband DCT: Definition, analysis, and applications," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 6, no. 6, pp. 273–286, Jun. 1996.

[15] J. Mukherjee and S. K. Mitra, "Image resizing in the compressed domain using subband DCT," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 12, no. 7, pp. 620–627, Jul. 1996.

[16] R. Dugad and N. Ahuja, "A scheme for spatial scalability using nonscalable encoders," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 10, pp. 993–999, Oct. 1996.

[17] G. H. Golub and C. F. Van-Loan, *Matrix Computations*, 3rd ed. Baltimore, MD: Johns Hopkins University Press, 1996.

[18] 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.

[19] ——, "Separable nonlinear least squares: The variable projection method and its applications," *Inverse Problems*, vol. 19, no. 2, pp. R1–R26, 2003.

[20] R. P. Brent, *Algorithms for Minimization Without Derivatives*. Englewood Cliffs, NJ: Prentice-Hall, 1973.