Combined Adaptive and Averaging Strategies for JPEG-Based Low Bit-Rate Image Coding

Ana-Maria Sevcenco and Wu-Sheng Lu
Dept. of Electrical and Computer Engineering
University of Victoria
Victoria, BC, CANADA V8W 3P6
sevcenco@engr.uvic.ca, wslu@ece.uvic.ca

Abstract — **A commonly recognized weak point of the DCT-based transform coding is its blocking effects which become increasingly visible in the low bit-rate territory. In the first part of this paper, motivated by a recent work of Bruckstein, Elad, and Kimmel (BEK) [1] and by the progress from [3], we propose a combined adaptive technique that can be applied to a BEK type of transform coding system for performance improvement. In the second part of the paper, motivated by a recent work of Tsaig, Elad, Milanfar, and Golub (TEMG) [2], we investigate an averaging technique for the design of optimal interpolation filter that can be utilized in a TEMG type system framework for further performance improvement. Simulation results are presented to demonstrate the effectiveness of the two proposed methods.**

Image coding; adaptive down-sampling; average strategy; local image features and statistics; quality factor of quantization matrix

I. Introduction

In spite of the emerging wavelet-based standard JPEG-2000 for still images [4], the discrete cosine transform (DCT) based JPEG standard [5]-[7] remains to be one of the most successful applications of transform coding methods for still digital images.

A commonly realized disadvantage of the DCT-based transform coding is its blocking artifacts which become increasingly visible when the bit-rate gets lower. In the past, considerable research endeavors have been made to deal with this problem. The work of particular interest and relevant to the methods described below are the algorithms recently proposed in [1]-[3] that are JPEG-based and incorporate an anti-aliasing filtering and down-sampling pre-processing step and an interpolative up-sampling post-processing step, which have demonstrated considerable performance improvement in terms of coding gain and reduced blocking effects. Studies of image coding problems employing similar frameworks can also be found in [8]-[11].

In this paper, two JPEG-based down-scaling techniques for low bit-rate image coding are proposed. The proposed techniques are built on similar system frameworks as in [1]-[3] (see Fig. 1) where effective and simple-to-implement adaptive and averaging schemes are incorporated for performance enhancement at a cost of moderate increase in complexity relative to that of [1] and [2]. Simulation results are presented to demonstrate the effectiveness of the proposed techniques.

II. Background Overview

A. *The Method of Bruckstein, Elad, and Kimmel (BEK)*

As illustrated in Fig. 1, the BEK method [1] (after a routine anti-aliasing filtering of the input image) *down-samples* the image to a lower resolution, applies JPEG to the down-scaled image with the *same* bit budget, and then interpolates the resulting image to obtain an image of original resolution. Since the JPEG is applied to an image of reduced size (for instance, the size is reduced by 4 when the image is downscaled by a factor of 2 in each dimension), using the same bit budget simply means that an increased number of bits can be assigned to each 8-by-8 image blocks, allowing more DCT coefficients to be included in the encoding process. It is intuitively clear that the above step shall help reduce the blocking effects. In addition, the interpolative up-sampling as a low-pass filtering step facilitates a further reduction of the blocking artifacts. A quantitative analysis that substantiates this idea and simulation results that demonstrate the effectiveness of the method especially for low bit-rate coding can be found in [1].

B. *The Method of Tsaig, Elad, Milanfar, and Golub (TEMG)*

For further performance enhancement, the method of TEMG [2] addresses the issue of *optimizing the decimation and interpolation* stages. The algorithm starts by applying an anti-aliasing separable FIR two-dimensional (2-D) filter to an input digital image X of size $m \times n$. In this case, the transfer

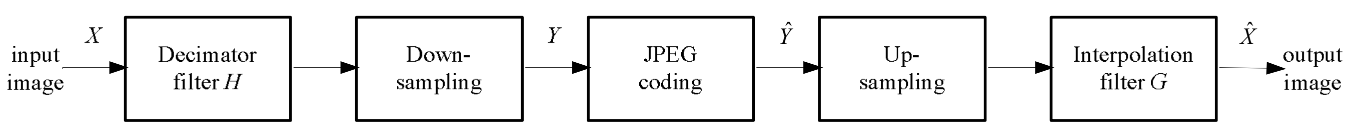

Figure 1. The general system framework.

function of the decimation filter can be expressed as $H(z_1, z_2) = h(z_1)\cdot h(z_2)$, where $h(z)$ is a Hamming-window based linear phase one-dimensional (1-D) FIR lowpass filter. In [2], the normalized cutoff frequency f_c of $H(z_1, z_2)$ is optimized so as to maximize the peak signal-to-noise ratio (PSNR). The output of decimation filter is down-sampled by k in both dimensions and the down-scaled image Y is encoded by DCT plus certain quantization and coding techniques before transmission. The received 2-D signal is decoded by corresponding decoding techniques and the resulting image $\hat{Y}$ is then up-sampled by k and filtered by an interpolation 2-D filter G to produce the reconstructed image $\hat{X}$.

In [2], the scaling factor is set to $k = 2$ and the upsampling and interpolative filtering are treated as a unified process which is performed using one filter G for interpolation. In a scenario such as this, filter G may be explicitly described by four subfilters $G^{(p,q)}$ with p and q assuming the values of 0 or 1, where each subfilter is applied to the decoded image $\hat{Y}$ to produce a subimage of size $(m/2)\times(n/2)$, and the four filtered subimages are used to construct image $\hat{X}$ as its even row/even column, even row/odd column, odd row/even column, and odd row/odd column parts, respectively. Each of the four nonseparable 2-D FIR filters is obtained by minimizing the Euclidean norm of an error vector that measures the difference between a subimage $x^{(p,q)}$ obtained by down-scaling the input image X by 2 and a subimage obtained by filtering Y using subfilter $G^{(p,q)}$. Here $x^{(p,q)}$ denotes a column vector formed by taking each row of the subimage, transposing it to a column, stacking it with the next column vector, and so on. In analytic terms, let the size of each subfilter $G^{(p,q)}$ be $l\times l$ and Φ be a matrix of size $s\times l^2$ with $s = mn/4$, where each row is obtained by associating a mask of size $l\times l$ with each component of image $\hat{Y}$ and stacking the elements of $\hat{Y}$ within the mask as a row vector. If we denote the vector version of the subfilter $G^{(p,q)}$ by $g^{(p,q)}$, then $g^{(p,q)}$ is obtained by minimizing the norm

$$\min\left\|\Phi\cdot g^{(p,q)} - x^{(p,q)}\right\|_2^2 \tag{1}$$

which leads to

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

where Φ^{+} denotes the pseudo-inverse of matrix Φ. Once filters $g^{(p,q)}$ for $p,\ q \in \{0,\ 1\}$ are determined, they are used to produce the even row/even column, even row/odd column, odd row/even column, and odd row/odd column parts of the output image $\hat{X}$ as $\hat{x}^{(p,q)} = \Phi\cdot g^{(p,q)}$.

C. *The Adaptive Down-Scaling Algorithms*

Adopting a BEK type framework, two adaptive down-scaling algorithms are proposed in [3] in order to further improve the coding performance in terms of the PSNR, and this is accomplished at a cost of moderate increase in complexity relative to that of [1].

Roughly speaking, the process of down-scaling an input image is made adaptive in two ways, both of which are based on the knowledge of the image's local features or statistics. In one way, the *down-scaling rate* is made adaptive to the image's local features of each $s\times s$ block, where the block size s depends on the set of down-scaling rates employed in the process; while in the other, the down-scaling rate is fixed but each down-scaled 8×8 block is associated with a *quality factor* of quantization matrix that is made adaptive to the image's features in that block using a linear interpolation over a given range of levels. In either adaptive framework, for a given bit budget, an increased number of bits is assigned to those blocks with rich features such as edges, while a reduced number of bits are given to those blocks that are less dynamic. This kind of bit allocation further eases off the blocking artifacts improving the coding performance. The adaptive down-scaling rate algorithm turns out to be effective for large size images containing regions with fewer details, while the adaptive quality factor algorithm is found to be more suitable for medium-size images, as it requires considerably more computing power than the BEK method.

Motivated by the line of thoughts in [1]-[3], we propose two improved strategies for JPEG-based low bit-rate image coding.

III. Combined Adaptive and Averaging Strategies

A. *A Combined Adaptive Strategy*

Based on the two adaptive algorithms that are separately presented in [3], here we propose a combined algorithm that takes the advantages of the two adaptive strategies. To describe this algorithm, we first recall several concepts and definitions from reference [3].

Let X be the input gray-scale image of size $N_1\times N_2$ with 8-bit pixel vales in the range [0, 255], and let the image be divided into a total of K blocks of size $s\times s$. Easy-to-compute and reasonable measures for the richness of the image details and "dynamics" in a given image block include its standard deviation and edge density measure [3]. The standard deviation associated with the k-th block is given by

$$\sigma_k = \frac{1}{s}\sqrt{\sum_{i=1}^{s}\sum_{j=1}^{s}\left(x_{i,j}^{(k)} - \bar{x}_k\right)^2} \tag{3}$$

where $x_{i,j}^{(k)}$ is the light intensity of the pixel at position (i, j) of the k-th image block of size $s\times s$, and $\bar{x}_k$ is the mean value of the block.

Let E be the binary edge map of the input image, which has the same size as the image itself and assumes value 1 at position (i, j) if the point happens to be on an edge or value 0 otherwise. Let E_k be the $s\times s$ submatrix of E associated with the k-th image block. An alternative measure, namely the edge density measure, for the richness of the k-th block can then be defined as

$$d_k = \frac{\text{sum}[E_k(:)]}{s^2} \quad (4)$$

where sum[E_k(:)] denotes the total number of 1's in E_k. This gives $0 \le d_k < 1$, and typically $d_k \ll 1$.

For a given set of integers sampling rates $\{R_k\}$, with $k = 1,2,...,K$, and minimum rate $R_{\min} = R_1 > 1$, the appropriate block size *s* is determined using

$$s = 8 \cdot lcm(R_1, R_2, ..., R_K) \quad (5)$$

where *lcm* represents the least common multiple of $\{R_k, k = 1,2,...,K\}$.

With the above preliminaries, the combined adaptive algorithm can be described as follows.

First, similar to the BEK method, an anti-aliasing filtering stage is applied to an input image *X*. Next, the filtered image is divided into blocks of size $s \times s$, where *s* is determined by (5). Using either (3) or (4), the local image statistics are generated for each block and certain thresholds δ_i with $i = 1, 2, \ldots, K - 1$ are selected. Based on this information, the down-sampling process can be performed for each $s \times s$ block by a specific rate R_k. Those blocks not down-sampled by the minimum rate $R_{\min}$ are subsequently bordered by blocks with constant value 128. In this way, the size of the combined block becomes $s_{down} = s / R_{\min}$. The resulting image is denoted by *Y*. As an illustrative example, image *peppers* so processed using two sampling rates 2 and 4 is shown in Fig. 2.

For the coding part, a modified JPEG transform is applied to compress and decompress the intermediate image *Y*, where the quality factor of the quantization matrix is made adaptive using again the local image statistics. Specifically, for a given intermediate image *Y* and a given range of values for the quality factors, q_{low}, q_{median}, and q_{high}, the local image statistics e_j are computed for each 8×8 block employing (3) or (4), and $e_{\min}$, e_{median}, and $e_{\max}$ are determined among the set of $\{e_j, j = 1,2,...,(N_1 / R_{\min}) \times (N_2 / R_{\min})\}$. Using these values, the quality factor q_j for a current block is obtained as

$$\text{if } e_j > e_{median}, \text{ then } q_j = q_{median} + (e_j - e_{median}) \cdot \frac{q_{high} - q_{median}}{e_{\max} - e_{median}}$$

$$\text{else } q_j = q_{low} + (e_j - e_{\min}) \cdot \frac{q_{median} - q_{low}}{e_{median} - e_{\min}}.$$

As a result, for a given bit budget, an intermediate 8×8 image block with rich details is allowed to use more bits to code its DCT coefficients than the less dynamic image blocks.

We note that a small overhead is required for the quality factors $\{q_k\}$ to be coded and transmitted along with the information data.

Figure 2. The intermediate image *Y* for image *peppers*.

At the receiver's end, the decoded image $\hat{Y}$ is selectively up-sampled and linearly interpolated for each $s_{down} \times s_{down}$ block by corresponding sampling rate of R_k or $R_{\min}$. Finally, the resulting image with size $(N_1 / R_{\min}) \times (N_2 / R_{\min})$ is up-sampled by $R_{\min}$ and linearly interpolated, like in the BEK method, to reconstruct image $\hat{X}$ with the same resolution as the input image *X*.

An alternative version of this combined algorithm can be obtained if one takes the advantage of the specific block structures of intermediate image *Y* (see Fig. 2) and compresses the blocks with different down-sampling rates by using different pre-defined quality factors. For a given set of down-sampling rates $\{R_k\}$, with $k = 1,2,...,K$, one can choose a corresponding set of quality factors $\{q_k\}$ to compress each $s \times s$ block from image *Y*. In the decoding stage, it is necessary to perform an additional intermediate step that decompresses the entire image using a fix known quality factor, which usually results in a poor quality intermediate image $\hat{Y}$. This $\hat{Y}$ is used only for identifying the necessary up-sampling rate R_k for each $s \times s$ block based on the specific block structures containing constant value 128, and then finding the corresponding quality factor q_k for the entire $s \times s$ block. Having done this, the decompression process is repeated using the appropriate quality factors, and the resulting image $\hat{Y}$ is adaptively up-sampled and interpolated, as previously described, to reconstruct image $\hat{X}$. Therefore, if fixed sets of down-scaling rates and quality factors are employed, there will be no need to code and transmit the map of quality factors.

B. An Averaging Strategy for Performance Improvement

Unlike the original TEMG algorithm which only utilizes the first subimage for further processing and neglecting the information contained in the other three subimages, we propose to take advantage of all available data.

Let Y_1, Y_2, Y_3, and Y_4 denote, respectively, the filtered and down-sampled even row/even column, even row/odd column, odd row/even column, and odd row/odd column subimages of input image *X*. Note that Y_1 here is taken as *Y* in the original TEMG algorithm. We propose to construct a new *Y* by computing the average of the four subimages and then rounded it to integer values, namely, $\overline{Y} = \text{round}[(Y_1 + Y_2 + Y_3 + Y_4)/4]$. It is intuitively clear that $\overline{Y}$ contains information from all four subimages, hence may serve as a better representative for input image *X*. Therefore, it is reasonable to hope that replacing *Y* with $\overline{Y}$ in our algorithms would lead to further performance improvement at the cost of a very moderate increase in complexity. In doing so, the matrix Φ depends on the average subimage $\overline{Y}$, as well as the interpolation filters vectors $g^{(p,q)}$, with $p,q=\{0,1\}$. The optimization problem in (1) and its solution given by (2) remain the same, with matrix Φ modified accordingly.

IV. Simulation Results

The proposed algorithms were applied to several medium size (512×512) test images including *peppers*, *lena*, *boat*, and *goldhill*. Due to the space limitation, below only the simulation results concerning image *peppers* are presented.

A. The Combined Adaptive Algorithm

The performance of the proposed combined adaptive (AdaComb) algorithm was compared with the performance of several existing algorithms: the BEK method [1], the adaptive down-scaling rate (AdaRate) algorithm [3] and the adaptive quality factor (AdaQFactor) algorithm [3].

Using a decimation filter with $f_c=0.7$ and linear interpolation in all four algorithms, a fixed quality factor q for the BEK, an adaptive quality factor which has been adjusted in the range $[-0.8+q,0.8+q]$ for the AdaQFactor and AdaComb, and 2 and 4 as down-scaling rates for the AdaRate and AdaComb, the results for image *peppers* are depicted in Fig. 3. Our simulation experiences also indicated that the AdaComb algorithm may obtain increased performance gain over the AdaRate algorithm if the designer has the patience in tuning of the input parameters carefully.

B. The Averaging Algorithm

By using the average of all four down-sampled subimages instead of just one subimage as the input to the encoder module in conjunction with the original TEMG algorithm, we obtained superior results in term of PSNR for medium size images.

For image *peppers*, both the decimation and interpolation filters have length 7. The decimation filter was designed using the Chebyshev window with f_c = 0.7. The simulation results are depicted in Fig. 4.

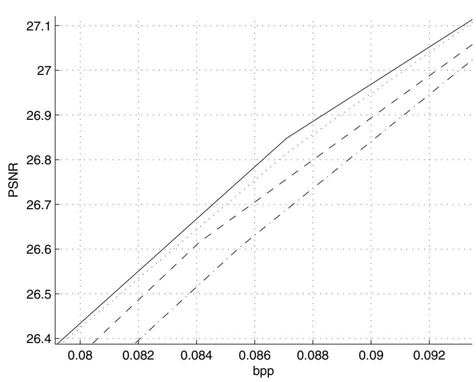

Figure 3. Performance comparison between AdaComb (solid curve), AdaRate (dotted curve), AdaQFactor (dashed curve) and BEK (dash-dotted curve), for image *peppers* using a decimation filter with f_c = 0.7.

V. Concluding Remarks

We have proposed two techniques for low bit-rate transform coding of digital images within the framework of [1]-[3] for further performance improvement. The first one is an adaptive technique that explores the down/up-sampling and coding stages, while the second one is an averaging strategy that is applied to the filtering stage. Simulation results are presented to demonstrate the effectiveness of the proposed methods. Based on the simulation results, it appears to be worthwhile to combine the adaptive and the averaging techniques into one algorithm.

References

[1] A. Bruckstein, M. Eland, and R. Kimmel, "Down-scaling for better transform compression", *IEEE Trans. on Image Processing*, vol. 12, no. 9, Sept. 2003.

[2] Y. Tsaig, M. Eland, P. Milanfar, and G. Golub, "Variable projection for near-optimal filtering in low bit-rate block coders", *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 15, no. 1, Jan. 2005.

[3] A.-M. Sevcenco and W.-S. Lu, "Adaptive down-scaling techniques for JPEG-based low bit-rate image coding", in *Proc. IEEE ISSPIT* 2006, Vancouver, August 27-30, 2006.

[4] "JPEG 2000 Image Coding System WG1N390 REV," ISO/IEC JTC1/SC 29/WG1.

[5] W. B. Pennebaker and J. L. Mitchell, *JPEG Still Image Compression Standard*, Van Nostrand Reinhold, New York, 1992.

[6] "Digital Compression and Coding of Continuous-Tone Still Pictures, Part I, Requirements and Guidelines," ISO/IEC JTC1 Committee Draft 10918-1 Std., Feb. 1991.

[7] "Information Technology – Coding of Moving Picturers and Associated Audio for Digital Storage Media at up to about 1.5 mbits," ISO/IEC 11172 Std., 1993.

[8] B. Zeng and A. N. Venetsanopoulos, "A JPEG-based interpolative image coding scheme," in *Proc. ICASSP*, vol. V, pp. 393-396, Minneapolis, MN., April 1993.

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

[10] D. Mukherjee and S. K. Mitra, "Image resizing in the compressed domain using subband DCT," *IEEE Trans. Circuits Syst. Video Technology*, vol. 6, no. 7, pp. 620-627, July 1996.

[11] R. Dugad and N. Ahuja, "A scheme for spatial scalability using nonscalable encoders," *IEEE Trans. Circuits Syst. Video Technology*, vol.13, no. 10, pp. 993-999, Oct. 2003.

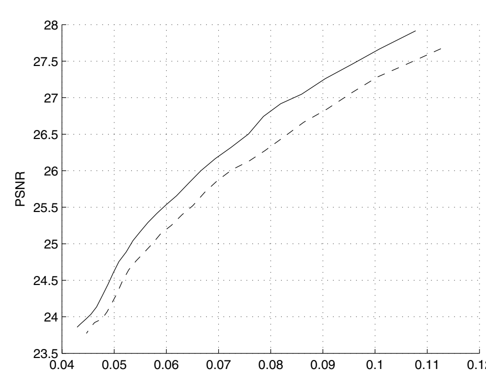

Figure 4. The average TEMG improvement over the original TEMG for image *peppers* using a decimation filter with f_c = 0.7.