

Down-Scaling for Better Transform Compression

Alfred M. Bruckstein, Michael Elad, and Ron Kimmel

Abstract—The most popular lossy image compression method used on the Internet is the JPEG standard. JPEG's good compression performance and low computational and memory complexity make it an attractive method for natural image compression. Nevertheless, as we go to low bit rates that imply lower quality, JPEG introduces disturbing artifacts. It is known that at low bit rates a down-sampled image when JPEG compressed visually beats the high resolution image compressed via JPEG to be represented with the same number of bits. Motivated by this idea, we show how down-sampling an image to a low resolution, then using JPEG at the lower resolution, and subsequently interpolating the result to the original resolution can improve the overall PSNR performance of the compression process. We give an analytical model and a numerical analysis of the down-sampling, compression and up-sampling process, that makes explicit the possible quality/compression trade-offs. We show that the image auto-correlation can provide good estimate for establishing the down-sampling factor that achieves optimal performance. Given a specific budget of bits, we determine the down sampling factor necessary to get the best possible recovered image in terms of PSNR.

Index Terms—Bit allocation, image down-sampling, JPEG compression, quantization.

I. INTRODUCTION

THE most popular lossy image compression method used on the Internet is the JPEG standard [1]. Fig. 1 presents a basic block diagram of the JPEG encoder. JPEG uses the Discrete Cosine Transform (DCT) on image blocks of size 8×8 pixels. The fact that JPEG operates on small blocks is motivated by both computational/memory considerations and the need to account for the nonstationarity of the image. A quality measure determines the (uniform) quantization steps for each of the 64 DCT coefficients. The quantized coefficients of each block are then zigzag-scanned into one vector that goes through a run-length coding of the zero sequences, thereby clustering long insignificant low energy coefficients into short and compact descriptors. Finally, the run-length sequence is fed to an entropy coder, that can be a Huffman coding algorithm with either a known dictionary or a dictionary extracted from the specific statistics of the given image. A different alternative supported by the standard is arithmetic coding.

JPEG's good middle and high rate compression performance and low computational and memory complexity make it an at-

tractive method for natural image compression. Nevertheless, as we go to low bit rates that imply lower quality, the JPEG compression algorithm introduces disturbing blocking artifacts. It appears that at low bit rates a down-sampled image when JPEG compressed and later interpolated, visually beats the high resolution image compressed directly via JPEG using the same number of bits. Whereas this property is known to some in the industry (see, for example, [9]), it was never explicitly proposed nor treated in the scientific literature. One might argue however, that the hierarchical JPEG algorithm implicitly uses this idea when low bit-rate compression is considered [1].

Let us first establish this interesting property though a simple experiment, testing the compression-decompression performance both by visual inspection and quantitative mean-square-error comparisons. An experimental result displayed in Fig. 2 shows indeed that both visually and in terms of the Mean Square Error (or PSNR), one obtains better results using down-sampling, compression, and interpolation after the decompression. Two comments are in order at this stage: i) throughout this paper, all experiments are done using Matlab v.6.1. Thus, simple IJG-JPEG is used with fixed quantization tables, and control over the compression is achieved via the Quality parameter and ii) throughout this paper, all experiments applying down-sampling use an anti-aliasing pre-filter, as Matlab 6.1 suggests, through its standard image resizing function.

Let us explain this behavior from an intuitive perspective. Assume that for a given image we use blocks of 8×8 pixels in the coding procedure. As we allocate too few bits (say 4 bits per block on average), only the DC coefficients are coded and the resulting image after decompression consists of essentially constant valued blocks. Such an image will clearly exhibit strong blocking artifacts. If instead the image is down-sampled by a factor of 2, the coder is now effectively working with blocks of 16×16 and has an average budget of $4 \cdot 2^2 = 16$ bits to code the coefficients. Thus, some bits will be allocated to higher order DCT coefficients as well, and the blocks will exhibit more detail. Moreover, as we up-scale the image at the decoding stage we add another improving ingredient to the process, since interpolation further blurs the blocking effects. Thus, the down-sampling approach is expected to result in better both visually and qualitatively outcomes.

In this paper we propose an analytical explanation to the above phenomenon, along with a practical algorithm to automatically choose the optimal down-sampling factor for best PSNR. Following the method outlined in [4], we derive an analytical model of the compression-decompression reconstruction error as a function of the memory budget, (i.e., the total number of bits) the (statistical) characteristics of the image, and the down-sampling factor. We show that a simplistic

Manuscript received May 29, 2001; revised April 22, 2003. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Trac D. Tran.

A. M. Bruckstein and R. Kimmel are with the Computer Science Department, The Technion—Israel Institute of Technology, Haifa 32000, Israel (e-mail: freddy@cs.technion.ac.il; ron@cs.technion.ac.il).

M. Elad is with the Computer Science Department—SCCM Program, Stanford University, Stanford, CA 94305 USA (e-mail: elad@sccm.stanford.edu).

Digital Object Identifier 10.1109/TIP.2003.816023

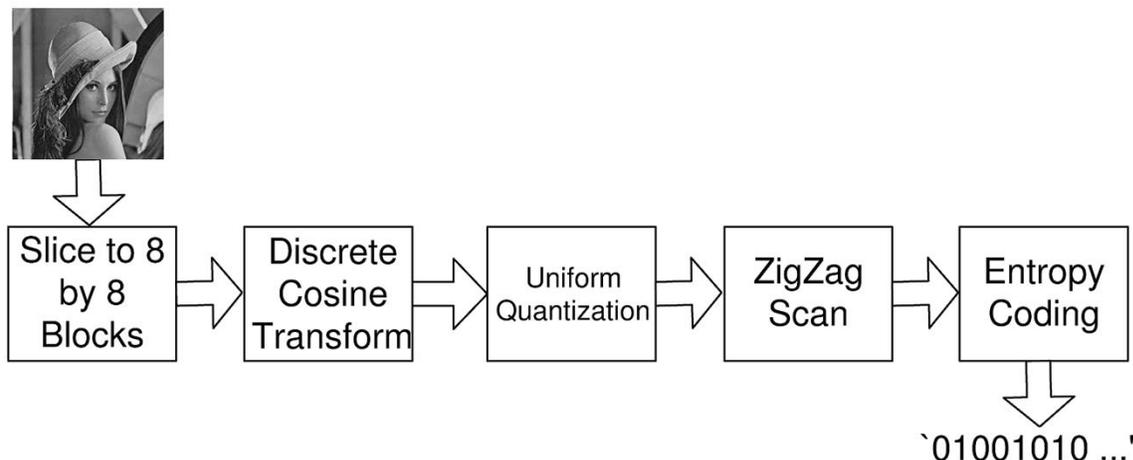


Fig. 1. JPEG encoder block diagram.



Fig. 2. Original image (on the left), JPEG compressed-decompressed image (middle), and down-sampled-JPEG compressed-decompressed and up sampled image (right). The down-sampling factor was 0.5. The compressed 256×256 “Lena” image in both cases used 0.25 bpp inducing MSEs of 219.5 and 193.12, respectively. The compressed 512×512 “Barbara” image in both cases used 0.21 bpp inducing MSEs of 256.04 and 248.42, respectively.

second order statistical model provides good estimates for the down-sampling factors that achieves optimal performance.

This paper is organized as follows. Sections II–IV present the analytic model and explore its theoretical implications. In Section II we start the analysis by developing a model that describes the compression-decompression error based on the quantization error and the assumption that the image is a realization of a Markov random field. Section III then introduces the impact of bit-allocation so as to relate the expected error to the given bit-budget. In Section IV we first establish several important parameters used by the model, and then use the obtained formulation in order to graphically describe the trade-offs between the

total bit-budget, the expected error, and the coding block-size. Section V describes an experimental setup that validates the proposed model and its applicability for choosing best down-sampling factor for a given image with a given bits budget. Finally, Section VI ends the paper with some concluding remarks.

II. ANALYSIS OF A CONTINUOUS “JPEG-STYLE” IMAGE REPRESENTATION MODEL

In this section we start building a theoretical model for analyzing the expected reconstruction error when doing compression-decompression as a function of the total bits budget, the

characteristics of the image, and the down-sampling factor. Our model considers the image over a continuous domain rather than a discrete one, in order to simplify the derivation. The steps we follow are as follows.

- 1) We derive the expected compression-decompression mean-square-error for a general image representation. Slicing of the image domain into M by N blocks is assumed.
- 2) We use the fact that the coding is done in the transform domain using an orthonormal basis, to derive to error induced due to truncation only.
- 3) We extend the calculation to account for quantization error of the nontruncated coefficients.
- 4) We specialize the image transform to the DCT basis.
- 5) We introduce an approximation for the quantization error, as a function of the allocated bits.
- 6) We explore several possible bit-allocation policies and introduce the overall bit-budget as a parameter into our model.

At the end of this process we obtain an expression for the expected error as a function of the bit budget, scaling factor, and the image characteristics. This function eventually allows us to determine the optimal down-sampling factor in JPEG-like image coding.

A. Compression-Decompression Expected Error

Assume we are given images on the unit square $[0, 1] \times [0, 1]$, $f_w(x, y) : [0, 1] \times [0, 1] \rightarrow \mathbf{R}$, realizations of a 2-D random process $\{f_w(x, y)\}$, with second order statistics given by

$$\begin{aligned} E[f_w(x, y)] &= 0, \text{ and } \mathcal{R}(x, y; x + \tau_x, y + \tau_y) \\ &= r_0^2 e^{-\alpha_x |\tau_x|} e^{-\alpha_y |\tau_y|}. \end{aligned} \quad (1)$$

Note that here we assume that the image is stationary. This is a marked deviation from the real-life scenario, and this assumption is done mainly to simplify our analysis. Nevertheless, as we shall hereafter see, the obtained model succeeds in predicting the down-sampling effect on compression-decompression performance.

We assume that the image domain $[0, 1] \times [0, 1]$ is sliced into $M \cdot N$ regions of the form

$$\begin{aligned} \Delta_{ij} &\equiv \left[\frac{i-1}{M}, \frac{i}{M} \right] \times \left[\frac{j-1}{N}, \frac{j}{N} \right] \\ \text{for } i &= 1, 2, \dots, M; \quad j = 1, 2, \dots, N. \end{aligned}$$

Assume that due to our coding of the original image $f_w(x, y)$ we obtain the compressed-decompressed result $\hat{f}_w(x, y)$, which is an approximation of the original image. We can measure the error in approximating $f_w(x, y)$ by $\hat{f}_w(x, y)$ as follows:

$$\begin{aligned} \mathcal{E}_w^2 &= \iint_{[0,1] \times [0,1]} (f_w(x, y) - \hat{f}_w(x, y))^2 dx dy \\ &= \sum_{i=1}^M \sum_{j=1}^N \iint_{\Delta_{ij}} (f_w(x, y) - \hat{f}_w(x, y))^2 dx dy \\ &= \sum_{i=1}^M \sum_{j=1}^N \frac{1}{M \cdot N} \{MSE_{f_w}(\Delta_{ij})\} \end{aligned} \quad (2)$$

where we define

$$\begin{aligned} MSE_{f_w}(\Delta_{ij}) &\equiv \frac{1}{\text{Area}(\Delta_{ij})} \\ &\quad \times \iint_{\Delta_{ij}} (f_w(x, y) - \hat{f}_w(x, y))^2 dx dy. \end{aligned} \quad (3)$$

We shall, of course, be interested in the expected mean square error of the digitization, i.e.,

$$\begin{aligned} E[\mathcal{E}_w^2] &= \sum_{i=1}^M \sum_{j=1}^N \frac{1}{M \cdot N} \\ &\quad E \left(\frac{1}{\text{Area}(\Delta_{ij})} \iint_{\Delta_{ij}} (f_w(x, y) - \hat{f}_w(x, y))^2 dx dy \right). \end{aligned} \quad (4)$$

Note that the assumed wide-sense stationarity of the image process results in the fact that the expression $E[MSE_f(\Delta_{ij})]$ is independent of (i, j) , i.e., we have the *same* expected mean square error over each slice of the image. Thus, we can write

$$\begin{aligned} E[\mathcal{E}_w^2] &= M \cdot N \frac{1}{M \cdot N} E[MSE_{f_w}(\Delta_{11})] \\ &= E \left[\frac{1}{\text{Area}(\Delta_{11})} \iint_{\Delta_{11}} (f_w(x, y) \right. \\ &\quad \left. - \hat{f}_w(x, y))^2 dx dy \right]. \end{aligned} \quad (5)$$

Up to now we considered the quality measure to evaluate the approximation of $f_w(x, y)$ in the digitization process. We shall next consider the set of basis functions needed for representing $f_w(x, y)$ over each slice.

B. Bases for Representing $f_w(x, y)$ Over Slices

In order to represent the image over each slice Δ_{ij} , we have to choose an orthonormal basis of functions. Denote this basis by $\{\Phi_{kl}(x, y)\}_{k,l=0,1,2,\dots}$. We must have

$$\begin{aligned} \iint_{\Delta_{ij}} \Phi_{kl} \Phi_{k'l'} dx dy &= \delta_{k-k'} \delta_{l-l'} \\ &= \begin{cases} 1, & \text{if } (k, l) \equiv (k', l') \\ 0, & \text{otherwise.} \end{cases} \end{aligned}$$

If $\{\Phi_{kl}\}$ is indeed an orthonormal basis then we can write

$$f_w(x, y) = \sum_{k=0}^{\infty} \sum_{l=0}^{\infty} \langle f_w(x, y), \Phi_{kl}(x, y) \rangle \Phi_{kl}(x, y), \quad (6)$$

as a representation of $f_w(x, y)$ over Δ_{ij} in terms of an infinite set of coefficients

$$\begin{aligned} F_{kl} &\equiv \langle f_w(x, y), \Phi_{kl}(x, y) \rangle \\ &= \iint_{\Delta_{ij}} f_w(x, y) \Phi_{kl}(x, y) dx dy. \end{aligned} \quad (7)$$

Suppose now that we approximate $f_w(x, y)$ over Δ_{ij} by using only a finite set Ω of the orthonormal functions $\{\Phi_{kl}(x, y)\}$, i.e., consider

$$\hat{f}_w(x, y) = \sum_{(k,l) \in \Omega} \langle f_w(x, y), \Phi_{kl}(x, y) \rangle \Phi_{kl}(x, y). \quad (8)$$

The optimal coefficients in the approximation above turn out to be the corresponding F_{kl} 's from the infinite representation. The mean square error of this approximation, over Δ_{11} say, will be

$$\begin{aligned} MSE_{f_w}(\Delta_{11}) &= \frac{1}{\frac{1}{M} \cdot \frac{1}{N}} \\ &\cdot \iint_{\Delta_{11}} \left(f_w(x, y) - \hat{f}_w(x, y) \right)^2 dx dy \\ &= M \cdot N \left[\iint_{\Delta_{11}} f_w^2(x, y) dx dy \right. \\ &\quad - 2 \iint_{\Delta_{11}} f_w(x, y) \hat{f}_w(x, y) dx dy \\ &\quad \left. + \iint_{\Delta_{11}} \hat{f}_w^2(x, y) dx dy \right]. \quad (9) \end{aligned}$$

Hence,

$$\begin{aligned} MSE_{f_w}(\Delta_{11}) &= M \cdot N \\ &\times \left[\iint_{\Delta_{11}} f_w^2(x, y) dx dy \right. \\ &\quad - 2 \sum_{(k,l) \in \Omega} \langle f_w(x, y) \phi_{kl}(x, y) \rangle \\ &\quad \times \iint_{\Delta_{11}} f_w(x, y) \phi_{kl}(x, y) dx dy \\ &\quad \left. + \sum_{(k,l) \in \Omega} \langle f_w(x, y) \phi_{kl}(x, y) \rangle^2 \right] \\ &= M \cdot N \\ &\times \left[\iint_{\Delta_{11}} f_w^2(x, y) dx dy \right. \\ &\quad \left. - \sum_{(k,l) \in \Omega} \langle f_w(x, y), \phi_{kl}(x, y) \rangle^2 \right]. \quad (10) \end{aligned}$$

Now the expected $MSE_{f_w}(\Delta_{11})$ will be

$$\begin{aligned} E[MSE_{f_w}(\Delta_{11})] &= M \cdot N \\ &\times \left[\iint_{\Delta_{11}} E[f_w^2(x, y)] dx dy \right. \\ &\quad - \sum_{(k,l) \in \Omega} E[\langle f_w(x, y), \\ &\quad \left. \phi_{k,l}(x, y) \rangle^2] \right]. \quad (11) \end{aligned}$$

Hence

$$\begin{aligned} E[\mathcal{E}_w^2] &= M \cdot N \cdot r_0^2 \cdot \frac{1}{M \cdot N} - M \cdot N \cdot \sum_{(k,l) \in \Omega} E[F_{kl}^2] \\ &= r_0^2 - M \cdot N \cdot \sum_{(k,l) \in \Omega} E[F_{kl}^2]. \quad (12) \end{aligned}$$

C. Effect of Quantization of the Expansion Coefficient F_{kl}

Suppose that in the approximation

$$\hat{f}_w(x, y) = \sum_{(k,l) \in \Omega} F_{kl} \Phi_{k,l}(x, y) \quad (13)$$

we can only use a finite number of bits in representing the coefficients $\{F_{kl}\}$ that take values in \mathbb{R} . If F_{kl} is represented/encoded with b_{kl} -bits we shall be able to describe it via F_{kl}^Q that takes on $2^{b_{kl}}$ values only, i.e., $F_{kl}^Q = Q_{b_{kl}}(F_{kl}) : \mathbb{R} \rightarrow$ set of $2^{b_{kl}}$ representation levels. The error in representing F_{kl} in this way is $\Gamma_{kl}^2 = (F_{kl} - F_{kl}^Q)^2$. Let us now see how the quantization errors affect the $MSE_{f_w}(\Delta_{11})$. We have

$$\begin{aligned} MSE_{f_w}^Q(\Delta_{11}) &= \frac{1}{\frac{1}{M} \cdot \frac{1}{N}} \\ &\times \iint_{\Delta_{11}} \left(f_w(x, y) - \hat{f}_w^Q(x, y) \right)^2 dx dy \quad (14) \end{aligned}$$

where $\hat{f}_w^Q(x, y) = \sum_{(k,l) \in \Omega} F_{k,l}^Q \phi_{k,l}(x, y)$. Some algebraic steps leads to the following result for the expected $MSE_{f_w}^Q(\Delta_{11})$. The expected $MSE_{f_w}^Q(\Delta_{11})$ is therefore given by

$$\begin{aligned} E[MSE_{f_w}^Q(\Delta_{11})] &= E[MSE_{f_w}(\Delta_{11})] \\ &\quad + M \cdot N \cdot \sum_{(k,l) \in \Omega} E[\Gamma_{kl}^2] \\ &= r_0^2 - M \cdot N \cdot \sum_{(k,l) \in \Omega} E[F_{kl}^2] \\ &\quad + M \cdot N \sum_{(k,l) \in \Omega} E \\ &\quad \times [(F_{kl} - F_{kl}^Q)^2]. \quad (15) \end{aligned}$$

Hence, in order to evaluate $E[(\mathcal{E}_w^Q)^2]$ in a particular representation when the image is sliced into $M \cdot N$ pieces and over each piece we use a subset Ω of the possible basis functions (i.e., $\Omega \subset \{(k, l) | k, l = 0, 1, 2, \dots\}$) and we quantize the coefficients with B_{kl} -bits we have to evaluate

$$\begin{aligned} &r_0^2 - M \cdot N \cdot \sum_{(k,l) \in \Omega} \{ \text{variance of } F_{kl} \} \\ &+ M \cdot N \cdot \sum_{(k,l) \in \Omega} \{ \text{expected error in quantizing} \\ &\quad F_{kl} \text{ with } (b_{kl}) \text{ bits} \}. \end{aligned}$$

D. An Important Particular Case: Markov Process With Separable Cosine Bases

We now return to the assumption that the statistics of $\{f_w(x, y)\}$ is given by (1), namely,

$$E[f_w(x, y)] = 0, \text{ and}$$

$$E[f_w(x, y) f_w(x + \tau_x, y + \tau_y)] = r_0^2 e^{-\alpha_x |\tau_x|} e^{-\alpha_y |\tau_y|}$$

and we choose a separable cosine basis for the slices, i.e., over $[0, 1/M] \times [0, 1/N]$, $\Phi_{kl}(x, y) = \varphi_k(x) \varphi_l(y)$, where

$$\varphi_k(x) = \sqrt{M(2 - \delta_k)} \cos kM\pi x, \quad k = 0, 1, 2, \dots$$

$$\varphi_l(y) = \sqrt{N(2 - \delta_l)} \cos lN\pi y, \quad l = 0, 1, 2, \dots$$

This choice of using the DCT basis is motivated by our desire to model the JPEG behavior. As is well known [6], the DCT offers a good approximation of the KLT if the image is modeled as a 2-D random Markov field with very high correlation factor.

To compute $E[\mathcal{E}_w^2]$ for this case we need to evaluate the variances of F_{kl} defined as

$$F_{kl} \equiv \iint_{\Delta_{11}} f_w(x, y) \varphi_k(x) \varphi_l(y) dx dy. \quad (16)$$

We have

$$\begin{aligned} E[F_{kl}^2] &= E \left[\iint_{\Delta_{11}} \iint_{\Delta_{11}} f_w(x, y) f_w(\xi, \eta) \varphi_k(x) \right. \\ &\quad \left. \times \varphi_l(y) \varphi_k(\xi) \varphi_l(\eta) dx dy d\xi d\eta \right] \\ &= \iint_{\Delta_{11}} \iint_{\Delta_{11}} r_0^2 e^{-\alpha_x |x-\xi|} e^{-\alpha_y |y-\eta|} \\ &\quad \cdot M(2 - \delta_k) \cos(k\pi Mx) \cos(k\pi M\xi) \\ &\quad \cdot N(2 - \delta_l) \cos(l\pi Ny) \cos(l\pi N\eta) \\ &\quad \cdot dx dy d\xi d\eta. \end{aligned} \quad (17)$$

Therefore, by separating the integrations we obtain

$$\begin{aligned} E[F_{kl}^2] &= r_0^2 (2 - \delta_k) \int_0^{1/M} \int_0^{1/M} e^{-\alpha_x |x-\xi|} \\ &\quad \times \cos(k\pi Mx) \cos(k\pi M\xi) M dx d\xi \\ &\quad \cdot (2 - \delta_l) \int_0^{1/N} \int_0^{1/N} e^{-\alpha_y |y-\eta|} \\ &\quad \times \cos(l\pi Ny) \cos(l\pi N\eta) N dy d\eta. \end{aligned} \quad (18)$$

Changing variables of integration to

$$\begin{aligned} \tilde{x} &= Mx, & \tilde{x} &\in [0, 1] & \tilde{\xi} &= M\xi, & \tilde{\xi} &\in [0, 1] \\ \tilde{y} &= Ny, & \tilde{y} &\in [0, 1] & \tilde{\eta} &= N\eta, & \tilde{\eta} &\in [0, 1] \end{aligned}$$

yields

$$\begin{aligned} E[F_{kl}^2] &= r_0^2 (2 - \delta_k) (2 - \delta_l) \cdot \frac{1}{M} \cdot \int_0^1 \int_0^1 e^{-(\alpha_x/M)|\tilde{x}-\tilde{\xi}|} \\ &\quad \times \cos(k\pi \tilde{x}) \cos(k\pi \tilde{\xi}) d\tilde{x} d\tilde{\xi} \\ &\quad \cdot \frac{1}{N} \int_0^1 \int_0^1 e^{-(\alpha_y/N)|\tilde{y}-\tilde{\eta}|} \cos(l\pi \tilde{y}) \\ &\quad \times \cos(l\pi \tilde{\eta}) d\tilde{y} d\tilde{\eta}. \end{aligned} \quad (19)$$

Let us define, for compactness, the following integral:

$$\int_0^1 \int_0^1 e^{-A|x-\xi|} \cos(k\pi x) \cos(k\pi \xi) dx d\xi \equiv \mathcal{M}(A; k, l). \quad (20)$$

Then we see that

$$\begin{aligned} E[F_{kl}^2] &= r_0^2 (2 - \delta_k) \cdot \frac{1}{M} \cdot \mathcal{M}\left(\frac{\alpha_x}{M}; k, k\right) \\ &\quad \cdot (2 - \delta_l) \cdot \frac{1}{N} \cdot \mathcal{M}\left(\frac{\alpha_y}{N}; l, l\right). \end{aligned} \quad (21)$$

An Appendix at the end of the paper derives $\mathcal{M}(A; k, l)$, leading to the following expression:

$$\begin{aligned} \mathcal{M}(A; k, l) &= (1 + \delta_{k-or-l}) \delta_{|k-l|} \frac{A}{A^2 + k^2 \pi^2} \\ &\quad - (1 + (-1)^{l+k}) \frac{A^2}{(A^2 + (l\pi)^2)(A^2 + (k\pi)^2)} \\ &\quad \cdot \frac{(2 - e^{-A}[(-1)^k + (-1)^l])}{2}. \end{aligned} \quad (22)$$

Hence

$$\begin{aligned} E[F_{kl}^2] &= \frac{4r_0^2}{M \cdot N} \left[\frac{\left(\frac{\alpha_x}{M}\right)}{\left(\frac{\alpha_x}{M}\right)^2 + k^2 \pi^2} \right] \\ &\quad \times \left(1 - (2 - \delta_k)(1 - (-1)^k e^{-(\alpha_x/M)}) \right) \\ &\quad \times \frac{\left(\frac{\alpha_x}{M}\right)}{\left(\frac{\alpha_x}{M}\right)^2 + k^2 \pi^2} \\ &\quad \cdot \left[\frac{\left(\frac{\alpha_y}{N}\right)}{\left(\frac{\alpha_y}{N}\right)^2 + l^2 \pi^2} \right] \\ &\quad \times \left(1 - (2 - \delta_l)(1 - (-1)^l e^{-(\alpha_y/N)}) \right) \\ &\quad \times \frac{\left(\frac{\alpha_y}{N}\right)}{\left(\frac{\alpha_y}{N}\right)^2 + l^2 \pi^2} \end{aligned} \quad (23)$$

E. Incorporating the Effect of Coefficient Quantization

According to rate-distortion theory, if we either assume uniform or Gaussian random variables, there is a formula for evaluating the Mean-Square-Error due to quantization. This formula, known to be accurate at high rates, is given by [3]

$$E \left[(F_{kl} - F_{kl}^Q)^2 \right] \sim \mathcal{K} \cdot \frac{\text{Var}\{F_{kl}\}}{2^{2b_{kl}}} \quad (24)$$

where \mathcal{K} is a constant in the range [1, 3] and b_{kl} represents the number of bits allocated for representing F_{kl} . Putting the above results together, we get that the expected mean square error in representing images from the process $\{f_w(x, y)\}$ with Markov statistics, by slicing the image plane into $M \cdot N$ slices and using, over each slice, a cosine basis is given by

$$\begin{aligned} E[(\mathcal{E}_w^Q)^2] &= r_0^2 - \sum_{k, l \in \Omega} r_0^2 (2 - \delta_k) (2 - \delta_l) \\ &\quad \times \mathcal{M}\left(\frac{\alpha_x}{M}; k, k\right) \mathcal{M}\left(\frac{\alpha_y}{N}; l, l\right) \\ &\quad + \sum_{k, l \in \Omega} r_0^2 (2 - \delta_k) (2 - \delta_l) \\ &\quad \times \mathcal{M}\left(\frac{\alpha_x}{M}; k, k\right) \mathcal{M}\left(\frac{\alpha_y}{N}; l, l\right) \cdot \frac{\mathcal{K}}{2^{2b_{kl}}} \\ &= r_0^2 \left\{ 1 - \sum_{k, l \in \Omega} (2 - \delta_k) (2 - \delta_l) \right. \\ &\quad \times \mathcal{M}\left(\frac{\alpha_x}{M}; k, k\right) \mathcal{M}\left(\frac{\alpha_y}{N}; l, l\right) \\ &\quad \left. \times \left[1 - \frac{\mathcal{K}}{2^{2b_{kl}}} \right] \right\}. \end{aligned} \quad (25)$$

This expression gives $E[(\mathcal{E}_w^Q)^2]$ in terms of r_0 , $\{\alpha_x, \alpha_y\}$ and $\{b_{kl}\}$ -the bits allocated to the coefficients F_{kl} where the subset of the coefficient is given via Ω .

III. SLICING AND BIT-ALLOCATION OPTIMIZATION PROBLEMS

Suppose we consider

$$\begin{aligned} E[(\mathcal{E}_w^Q)^2] &= r_0^2 \left\{ 1 - \sum_{k, l \in \Omega} (2 - \delta_k) \mathcal{M}\left(\frac{\alpha_x}{M}; k, k\right) \right. \\ &\quad \left. \times (2 - \delta_l) \mathcal{M}\left(\frac{\alpha_y}{N}; l, l\right) \left[1 - \frac{\mathcal{K}}{2^{2b_{kl}}} \right] \right\} \end{aligned} \quad (26)$$

as a function of $M, N, \{b_{kl}\}$. We have that the total bit usage in representing the image is

$$B_{\text{TOT}} = \left(\sum_{k,l \in \Omega} b_{kl} \right) \cdot M \cdot N.$$

Now we can solve a variety of bit-allocation and slicing optimization problems [3].

It is important to note that in adopting a bit-allocation procedure we effectively deviate from the way the JPEG assigns bits to the different DCT coefficients. In the JPEG algorithm, a uniform quantizer is used, which at a first glance may appear to be inferior to any reasonable bit-allocation method. However, due to the entropy coding (Huffman, and RLE) following the quantization stage, the overall bit assignment effect seems to be similar to a global procedure of bit-allocation based on variances, and for that matter, approximates well the rate-distortion behavior.

A. Optimal Local Bit Allocation and Slicing Given Total Bit Usage

The problem here is: Given the constraint $MN \sum_{(k,l) \in \Omega} b_{kl} = B_{\text{TOT}}$, find $\{b_{kl}^*\}$ that minimize the $E[\mathcal{E}_w^2]$. Thus, the following expression is to be minimized with respect to $\{b_{kl}\}$

$$\sum_{k,l \in \Omega} \sum_{\overbrace{(2 - \delta_k) \mathcal{M}\left(\frac{\alpha_x}{M}; k, k\right) (2 - \delta_l) \mathcal{M}\left(\frac{\alpha_y}{N}; l, l\right) \mathcal{K}}^{\tilde{\sigma}_{kl}^2}} \times 2^{-2b_{kl}}. \quad (27)$$

This is a classical bit allocation process and we have that the optimal bit allocation yields (theoretically) the same error for all terms in

$$\sum_{k,l \in \Omega} \sum_{\tilde{\sigma}_{kl}^2} \frac{\tilde{\sigma}_{kl}^2}{2^{2b_{kl}}} = \sum_{k,l \in \Omega} \sum_{\Lambda_{kl}^2} \frac{\tilde{\sigma}_{kl}^2}{\Lambda_{kl}^2} \quad (28)$$

where we defined Λ_{kl} as the number of quantization levels, see [6]. Hence, we need

$$\frac{\tilde{\sigma}_{kl}^2}{\Lambda_{kl}^2} = \text{Const} \Rightarrow \Lambda_{kl}^2 = \frac{\tilde{\sigma}_{kl}^2}{\text{Const}} \quad (29)$$

and we should have

$$2^2 \sum_{k,l \in \Omega} b_{kl} = \prod_{k,l \in \Omega} \Lambda_{kl}^2 = 2^{2B_{\text{TOT}}/(MN)}. \quad (30)$$

The result is

$$\prod_{k,l \in \Omega} \frac{\tilde{\sigma}_{kl}^2}{(\text{Const})} = 2^{2B_{\text{TOT}}/(MN)} \quad (31)$$

or

$$\prod_{k,l \in \Omega} \tilde{\sigma}_{kl}^2 = (\text{Const})^{|\Omega|} \cdot 2^{2B_{\text{TOT}}/(MN)}. \quad (32)$$

Hence

$$\Lambda_{kl}^2 = \tilde{\sigma}_{kl}^2 \frac{|\Omega| \sqrt{2^{2B_{\text{TOT}}/(MN)}}}{\left(\prod_{(k,l) \in \Omega} \tilde{\sigma}_{kl}^2\right)^{1/|\Omega|}} \quad (33)$$

or

$$2^{2b_{kl}} = \tilde{\sigma}_{kl}^2 \cdot 2^{2B_{\text{TOT}}/(MN) - 1/|\Omega|} \cdot \left(\prod_{(k,l) \in \Omega} \tilde{\sigma}_{kl}^2 \right)^{-(1/|\Omega|)} \quad (34)$$

yielding

$$b_{kl} = \frac{1}{2} \log_2 \tilde{\sigma}_{kl}^2 + \frac{B_{\text{TOT}}}{M \cdot N} \cdot \frac{1}{|\Omega|} - \frac{1}{2} \log_2 \left(\prod_{(k,l) \in \Omega} \tilde{\sigma}_{kl}^2 \right)^{1/|\Omega|}. \quad (35)$$

With this optimal bit allocation the expression

$$\sum_{(k,l) \in \Omega} \sum_{\tilde{\sigma}_{kl}^2} \frac{\tilde{\sigma}_{kl}^2}{2^{2b_{kl}}} \quad (36)$$

is minimized to

$$|\Omega| \cdot \text{Const} = |\Omega| \cdot 2^{(-2B_{\text{TOT}}/M \cdot N \cdot |\Omega|)} \left(\prod_{(k,l) \in \Omega} \tilde{\sigma}_{kl}^2 \right)^{1/|\Omega|}. \quad (37)$$

Hence,

$$E[(\mathcal{E}_w^Q)^2]_{OPT} = r_0^2 \left\{ 1 - \sum_{k,l \in \Omega} (2 - \delta_k)(2 - \delta_l) \times \mathcal{M}\left(\frac{\alpha_x}{M}; k, k\right) \mathcal{M}\left(\frac{\alpha_y}{N}; l, l\right) + |\Omega| 2^{-2B_{\text{TOT}}/rM \cdot N \cdot 1/|\Omega|} \times \left(\prod_{(k,l) \in \Omega} \mathcal{K} \cdot (2 - \delta_k)(2 - \delta_l) \times \mathcal{M}\left(\frac{\alpha_x}{M}; k, k\right) \mathcal{M}\left(\frac{\alpha_y}{N}; l, l\right) \right)^{1/|\Omega|} \right\} \quad (38)$$

an error expression in terms of $(B_{\text{TOT}}, M, N, \Omega)$ and the second-order—statistics parameters r_0, α_x, α_y of the $\{f_w(x, y)\}$ -process.

B. Effect of Slicing With Rigid Relative Bit Allocation

An alternative bit allocation strategy, perhaps more in the spirit of the classical JPEG standard, can also be thought of. Consider that Ω is chosen and the b_{kl} 's are also chosen *a priori* for all $(k, l) \in \Omega$. Then, we have

$$E[(\mathcal{E}_w^Q)^2] = r_0^2 \left\{ 1 - \sum_{k,l \in \Omega} (2 - \delta_k)(2 - \delta_l) \times \mathcal{M}\left(\frac{\alpha_x}{M}; k, k\right) \mathcal{M}\left(\frac{\alpha_y}{N}; l, l\right) \left[1 - \frac{\mathcal{K}}{2^{2b_{kl}}} \right] \right\} \quad (39)$$

as a function of M and N . This function clearly decreases with increasing M and N since more and more bits are allocated to the image, and here $B_{\text{TOT}} = M \cdot N \cdot \sum_{k,l \in \Omega} b_{kl}$. Suppose now that for $M = N = 1$, we choose a certain bit allocation for a given Ω (say $\Omega = \{(k, l) | k + l \leq \text{Limit}, k, l = 0, 1, 2, \dots\}$), i.e., we chose \bar{b}_{kl} but now as we increase the number of slices (i.e., increase M and N) we shall modify the b_{kl} 's to keep B_{TOT} a constant by choosing $b_{kl}(M, N) = \bar{b}_{kl} \cdot 1/M \cdot N$. Here B_{TOT} remains a constant and we can again analyze the behavior of $E[(\mathcal{E}_w^Q)^2]$ as M and N vary.

In our experiments we assumed that the above limit was set to 7 (i.e., coefficients with $(k, l) \in \{k, l = 0, 1, 2, \dots, k + l \leq 7\}$ are assigned with bits). As to the choice of the b_{kl} per

each of these coefficients, we used a fixed template based on the following JPEG quantization table (see [1]) that specifies the quantization step per each coefficient

$$\begin{bmatrix} 16 & 11 & 10 & 16 & 24 & 40 & 51 & 61 \\ 12 & 12 & 14 & 19 & 26 & 58 & 60 & 55 \\ 14 & 13 & 16 & 24 & 40 & 57 & 69 & 56 \\ 14 & 17 & 22 & 29 & 51 & 87 & 80 & 62 \\ 18 & 22 & 37 & 56 & 68 & 109 & 103 & 77 \\ 24 & 35 & 55 & 64 & 81 & 104 & 113 & 92 \\ 49 & 64 & 78 & 87 & 103 & 121 & 120 & 101 \\ 72 & 92 & 95 & 98 & 112 & 100 & 103 & 99 \end{bmatrix}.$$

Values of b_{kl} were chosen to be inversely proportional to these values.

C. Soft Bit Allocation With Cost Functions For Error and Bit Usage

We could also consider cost functions of the form

$$C_{MSE} \left(E \left[(\mathcal{E}_W^Q)^2 \right] \right) + C_B \left(M \cdot N \cdot \frac{\text{Bits}}{\text{slice}} \right)$$

where C_{MSE} and C_B are cost functions chosen according to the task in hand, and ask for the bit allocation that minimize the joint functionals, in the spirit of [5].

IV. THEORETICAL PREDICTIONS OF THE MODEL

In Sections II and III, we proposed a model for the compression error as a function of the image statistics (r_0, α_x, α_y), the given total bits budget B_{TOT} , and the number of slicings M and N . Here, we fix these parameters according to the behavior of natural images and typical compression setups and study the behavior of the theoretical model.

Assume we have a gray scale image of size 512×512 with 8 bits/pixel as our original image. JPEG considers 8×8 slices of this image and produces, by digitizing the DCT transform coefficients with a predetermined quantization table, approximate representation of these 8×8 slices. We would like to explain the observation that down-sampling the original image, prior to applying JPEG compression to a smaller image, produces with the same bit usage, a better representation of original image.

Suppose the original image is regarded as the ‘‘continuous’’ image defined over the unit square $[0, 1] \times [0, 1]$, as we have done in the theoretical analysis. Then, the pixel width of a 512×512 image will be $1/512$. We shall assume that the original image is a realization of a zero mean 2-D stationary random process with autocorrelation of the form $\mathcal{R}(|i - I'|, |j - j'|) = r_0^2 |\rho_1|^{|i-i'|} |\rho_2|^{|j-j'|}$, with ρ_1 , and ρ_2 in the range of $[0.65, 0.95]$, as is usually done (see [6]). From a single image, r_0^2 can be estimated via the expression

$$\begin{aligned} r_0^2 &\approx \frac{1}{512 \times 512} \left(\sum_{i,j} (I(i,j) - \bar{I})^2 \right) \\ &= 2 \frac{1}{256} \left(\sum_0^{127} (i + 0.5)^2 \right) = 5461.3 \end{aligned}$$

assuming an equalized histogram. If we consider that

$$\rho^{|i-i'|} \cong e^{-\alpha|i/512-i'/512|} = e^{-(\alpha/512)|i-i'|}$$

we can obtain an estimate for α using $e^{-\alpha/512} = \rho \in [0.6, 0.95]$, This provides

$$-\frac{\alpha}{512} = \log_e \rho \longrightarrow -512 \times \log_e \rho = \alpha \in [25, 220].$$

The total number of bits for the image representation will range from 0.05 bpp to about 2.0 bpp, hence, B_{TOT} will be between $512 \times 512 \times 0.05 = 13107$ to $512 \times 512 \times 2 = 524,288$ bits for 512×512 original images. Therefore, in the theoretical evaluations we shall take $\alpha_x, \alpha_y \in [25, 220]$, $r_o \approx 5000$ for 256 gray level images, with total bit usage between 10 000 and 20 000.

The symmetric x and y axis slicings considered will be $M, N = 1, 2, \dots, 64$, where we assume that $\alpha_x = \alpha_y$ and $M = N$. Then we shall evaluate [see (26)]

$$E \left[(\mathcal{E}_w^Q)^2 \right] \equiv r_0^2 \left\{ 1 - \sum_{k,l \in \Omega} (2 - \delta_k)(2 - \delta_l) \mathcal{M} \left(\frac{\alpha}{M}, k, k \right) \mathcal{M} \left(\frac{\alpha}{N}, l, l \right) \left[1 - \frac{\mathcal{K}}{\Lambda_{kl}^2} \right] \right\}$$

with Λ_{kl} -s provided by the optimal level allocation

$$\begin{aligned} \Lambda_{kl}^2 &= (2 - \delta_k)(2 - \delta_l) \mathcal{M} \left(\frac{\alpha}{M}, k, k \right) \mathcal{M} \left(\frac{\alpha}{M}, l, l \right) \\ &\quad \times \mathcal{K} (2^{2B_{TOT}/M^2})^{1/|\Omega|} \frac{1}{\left[\prod_{(k,l) \in \Omega} (\tilde{\sigma}_{kl}^2) \right]^{1/|\Omega|}}. \end{aligned}$$

Practically, the optimal level allocation Λ_{kl}^* should be given by $\Lambda_{kl}^* = \max(1, \lfloor \Lambda_{kl} \rfloor)$, a measure that automatically prevents the allocation of negative numbers of bits. Obviously this step must be followed by re-normalization of the bit allocation in order to comply with the bits budget constraint. \mathcal{K} can be taken from 1 to 3, whereas Ω will be $\{(k,l) | k+l \leq 7, k,l = 0,1,\dots,7\}$, simulating the standard JPEG approach which is coding of 8×8 transform coefficients, emphasizing the low frequency range via the precise encoding of only about $|\Omega| = 36$ coefficients.

Using the above described parameter ranges, we plot the predictions of the analytical model for the expected mean square error as a function of the slicings M with bit usage as a parameter.

Figs. 3 and 4 demonstrate the approximated error as a function of the number of slicings for various total number of bits. Fig. 3 displays the predictions of the theoretical model in conjunction with optimal level allocation while Fig. 4 uses the JPEG style rigid relative bit allocation. In both figures the left side shows the results of restricting the number of bits or quantization levels to integers, while the right side shows the results allowing fractional bit and level allocation.

These figures show that for every given total number of bits there is an optimal slicing parameter M indicating the optimal down-sampling factor. For example, if we focus on the bottom right graph (rigid bit allocation with possible fractions), if 50 Kbits are used, the optimal M is found to be 32. This implies that an image of size 512×512 should be sliced to blocks of size $512/32 \times 512/32 = 16 \times 16$. As we move to a budget of only 10 Kbits, optimal M is found to be 18 and the block size to work with becomes $512/18 \times 512/8 \approx 28 \times 28$. Since JPEG works with fixed block size of 8×8 , the first case of 50 Kbits

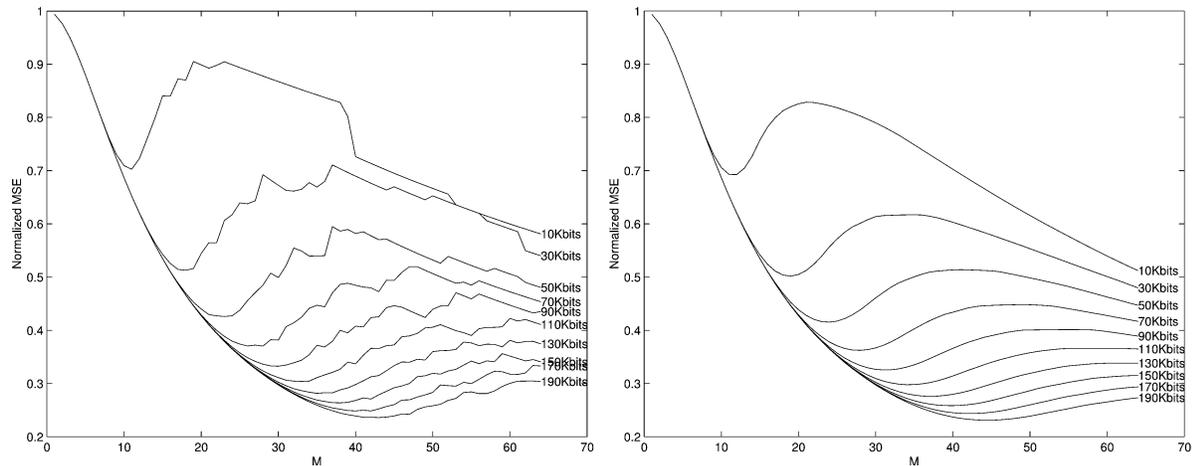


Fig. 3. Theoretical prediction of MSE based on *optimal bit allocation* versus number of slicings M with total bits usage as a parameter. Here, we used the typical values $\alpha = 150$, and $\mathcal{K} = 3$.

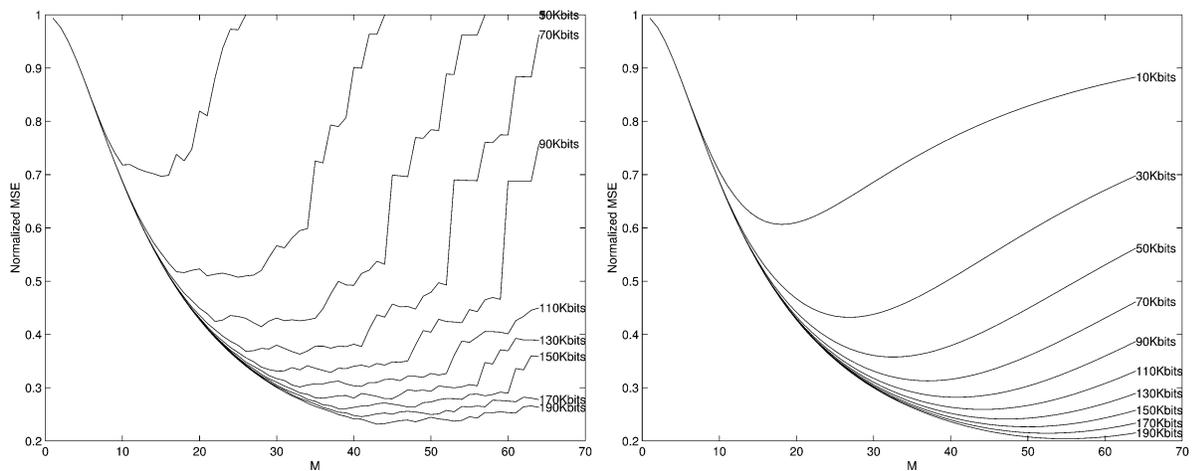


Fig. 4. *Rigid relative bit allocation* based prediction of MSE versus number of slicings M with total bits usage as a parameter. Here, we used the typical values $\alpha = 150$, and $\mathcal{K} = 1$.

calls for a down-sampling by a factor of 2, and the second case with 10 Kbits requires a down-scaling factor of 3.5.

Note that integer bit allocation causes in both cases non-smooth behavior. Also in Fig. 3 it appears that the minimum points are local ones and the error tends to decrease again as M increases. This phenomenon can be explained by the fact that we used an approximation of the quantization error which fails to predict the true error for a small number of bits at large down-scaling factors. Finally, we should note that the parameter \mathcal{K} was chosen differently between the two allocation policies. Within the range $[1, 3]$, we empirically set a value that matched the true JPEG behavior. The overall qualitative behavior however was quite similar for all the range of \mathcal{K} 's.

Fig. 5 shows the theoretical prediction of PSNR versus bits per pixel curves for typical 512×512 images with different down-sampling factors (different values of M , where the down-sampling factor is $= 8M/512$). One may observe that the curve intersections occur at similar locations as those of the experiments with real images shown in Section V. Also, it appears that even though the allocation policies are different, the results are very similar.

An interesting phenomenon is observed in these graphs: For down-sampling factors smaller than 1 an almost flat saturation of the PSNR versus bit-rate is seen at sufficiently large bit-rate. This phenomenon is quite expected, since the down-sampling of the image introduces un-recoverable loss. Thus, even an infinite amount of bits can not recover this induced error, and this is the obtained saturation height. The reason this fluttering happens at lower bit-rates for smaller factors is that the smaller the image, the smaller the amount that will be considered as leading to near-perfect transmission. In terms of quality-scalable coder, this effect is parallel to the attempt to work with an image pyramid representation and allocating too many bits to a specific resolution layer, instead of also allocating bits to the next resolution layer.

V. COMPRESSION RESULTS OF NATURAL AND SYNTHETIC IMAGES

To verify the validity of the analytic model and design a system for image trans-coding we can generate synthetic images for which the autocorrelation is similar to that of a

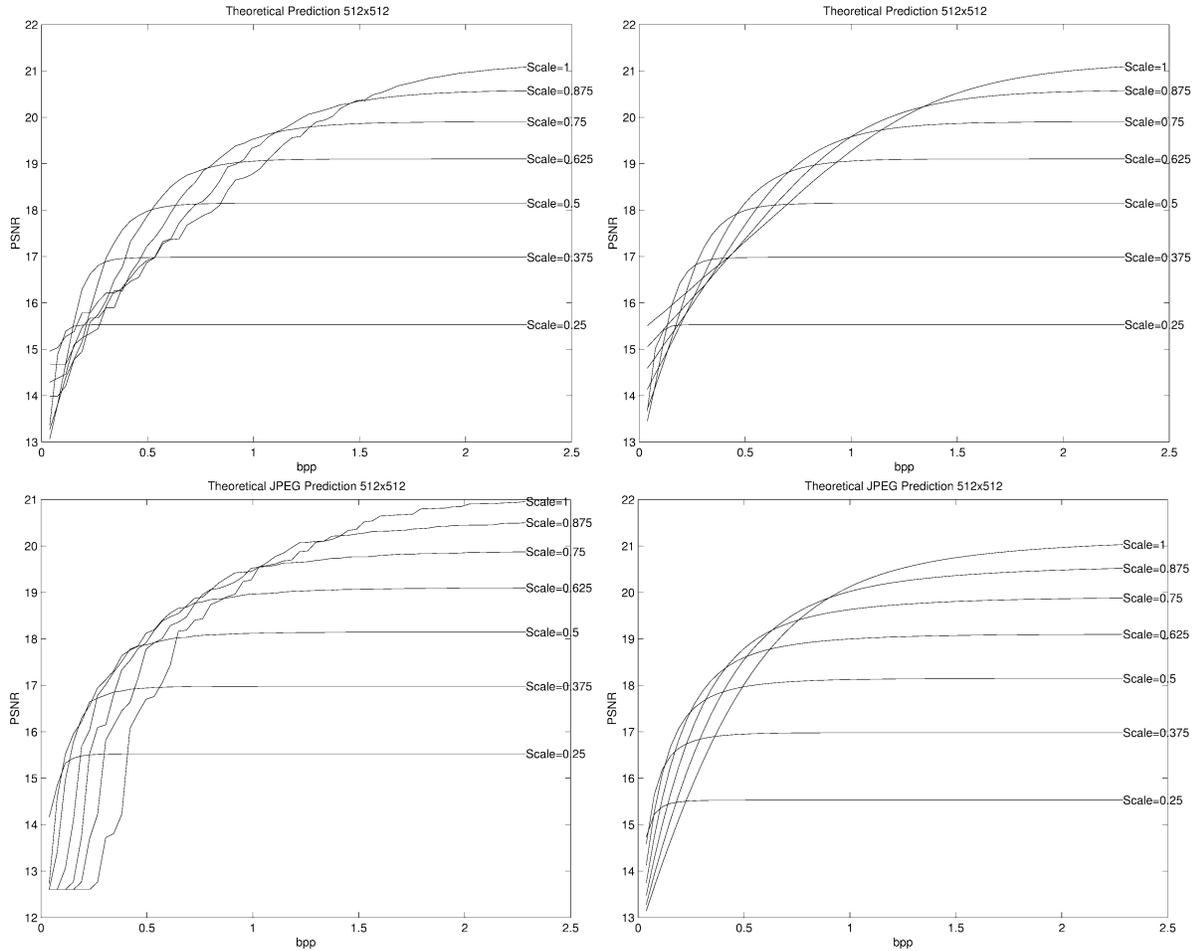


Fig. 5. Optimal (top) and rigid relative (bottom) bit allocation based prediction of PSNR versus bits per pixel with image down-sampling as a parameter. Again, we used the typical values $\alpha = 150$, and $\mathcal{K} = 3$ for the optimal bit allocation case, and $\mathcal{K} = 1$ for the JPEG-style case.

given image. Then we can plot the PSNR/bpp JPEG graphs for all JPEG qualities, one graph for each given down-sampling ratio. The statistical model is considered valid if the behavior is similar for the natural image and the synthesized one.

A. Image Synthesis

Assume that for an image $g(m, n)$ the autocorrelation function is that of a sample of an ergodic homogeneous random field of the form we assumed, hence

$$R_{gg}(m, n) = \frac{1}{MN} \sum_{m'=0}^{M-1} \sum_{n'=0}^{N-1} g(m', n')g(m'+m, n'+n) \\ = e^{-\alpha_x|m| - \alpha_y|n|}.$$

Define the Fourier transform $\hat{g}(k) = \mathcal{F}\{g(x)\}$. Then, the power spectrum of the real signal is given by

$$\mathcal{F}\{R_{gg}(x)\} \triangleq P_{gg}(k) = \hat{g}(k)\hat{g}^*(k) = |\hat{g}(k)|^2.$$

Now, considering a 1D signal with the above statistics, we have

$$\hat{g}(k)\hat{g}^*(k) = \mathcal{F}\{e^{-\alpha|x|}\} = \frac{2\alpha}{\alpha^2 - k^2}$$

Thus, we have that $\hat{g}(k) = \sqrt{2\alpha}\alpha \pm ik/\alpha^2 + k^2$. The solution $\hat{g}(k) = \sqrt{2\alpha}\alpha - ik/\alpha^2 + k^2$ is chosen so as to satisfy $g(x) \neq 0$ for $x > 0$. Therefore

$$g(x) \equiv \mathcal{F}^{-1}\{\hat{g}(k)\} = \mathcal{F}^{-1}\left\{\sqrt{\frac{\alpha}{2}}\frac{2\alpha}{\alpha^2 + k^2}\right\} \\ + \mathcal{F}^{-1}\left\{-\frac{1}{\sqrt{2\alpha}}\frac{ik}{\alpha^2 + k^2}\right\} \\ = \sqrt{\frac{\alpha}{2}}e^{-\alpha|x|} - \frac{1}{\sqrt{2\alpha}}\frac{d}{dx}e^{-\alpha|x|} \\ = \sqrt{\frac{\alpha}{2}}e^{-\alpha|x|}(1 + \text{sgn}(x)) \\ = \begin{cases} \sqrt{2\alpha}e^{-\alpha|x|} & x > 0 \\ 0 & x \leq 0. \end{cases}$$

To generate synthetic images, we can “color” a uniform random (white) noise as follows. Let g_n be an $M \times N$ matrix in which each entry is a uniformly distributed random number. Next, let p be an $M \times N$ matrix with elements

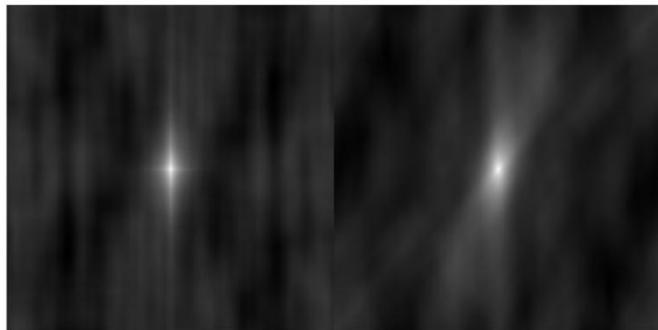
$$p(m, n) = \begin{cases} \sqrt{2\alpha_x}e^{-\alpha_x m}, & n = \frac{N}{2}, m = 1, \dots, M \\ 0, & \text{otherwise} \end{cases}$$

and similarly, q is an $M \times N$ matrix such that

$$q(m, n) = \begin{cases} \sqrt{2\alpha_y}e^{-\alpha_y n}, & m = \frac{M}{2}, n = 1, \dots, N \\ 0, & \text{otherwise.} \end{cases}$$



(a)



(b)

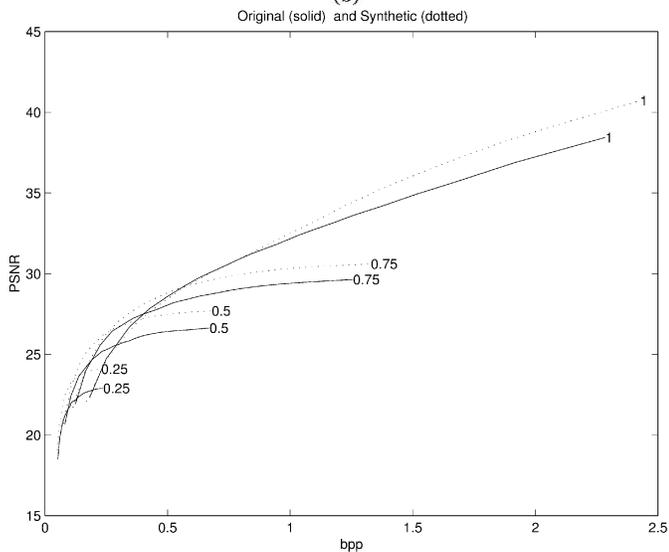


Fig. 6. Comparison between a natural image, and a synthesized one with similar autocorrelation ($\alpha_x = 27$, $\beta = 11$).

A synthetic sample image with desired autocorrelation is then generated by the process

$$[g(m, n)] = \mathcal{F}_{2D}^{-1} \{ \mathcal{F}_{2D}\{g_n\} \cdot \mathcal{F}_{2D}\{p\} \cdot \mathcal{F}_{2D}\{q\} \}.$$

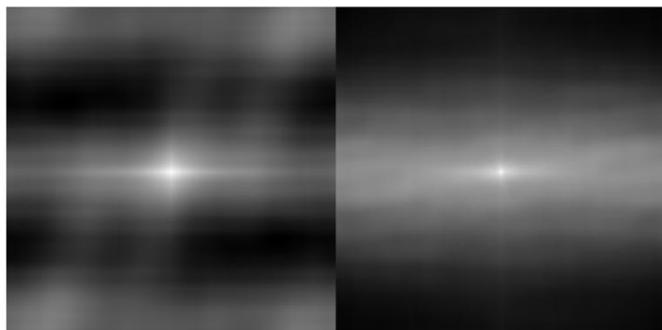
B. Estimating the Image Statistics α_x and α_y

In order to generate a synthetic image with the same statistics as that of the natural one, we have to first estimate the properties of the given image. Let us present a simple method for estimating the image statistics. We already used the relation

$$\mathcal{F}_{2D}\{R_{gg}(x, y)\} \triangleq P_{gg}(k, l) = \hat{g}(k, l)\hat{g}^*(k, l).$$



(a)



(b)

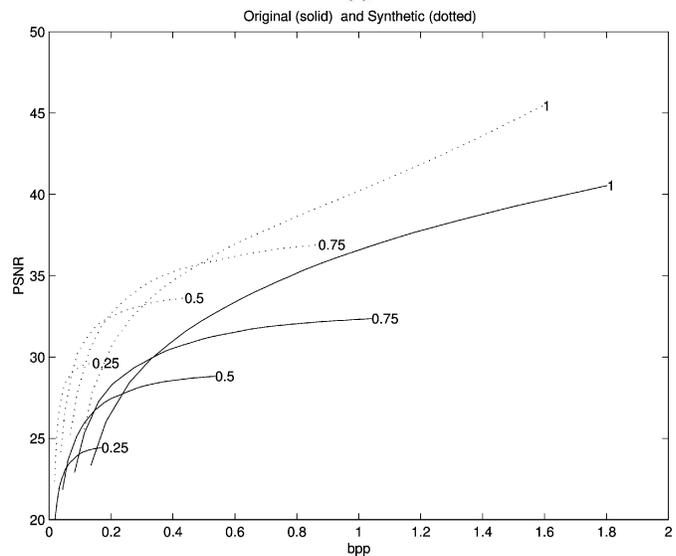


Fig. 7. Comparison between a natural image and a synthesized one with similar autocorrelation ($\alpha_x = 50$, $\alpha_y = 100$).

Explicitly, for our statistical image model we have that the power spectrum is given by

$$\frac{2\alpha_x}{\alpha_x^2 + k^2} \frac{2\alpha_y}{\alpha_y^2 + l^2} = \hat{g}(k, l)\hat{g}^*(k, l),$$

and the autocorrelation is

$$e^{-\alpha_x|x| - \alpha_y|y|} = \mathcal{F}_{2D}^{-1}\{\hat{g}(k, l)\hat{g}^*(k, l)\}.$$

Thus, all we need to do is to estimate the slopes of the plane given by

$$\begin{aligned} \alpha_x|x| + \alpha_y|y| &= -\ln(\mathcal{F}_{2D}^{-1}\{\hat{g}(k, l)\hat{g}^*(k, l)\}) \\ &= -\ln(\mathcal{F}_{2D}^{-1}\{\mathcal{F}_{2D}\{g(x, y)\}\mathcal{F}_{2D}\{g(x, y)\}^*\}) \end{aligned}$$

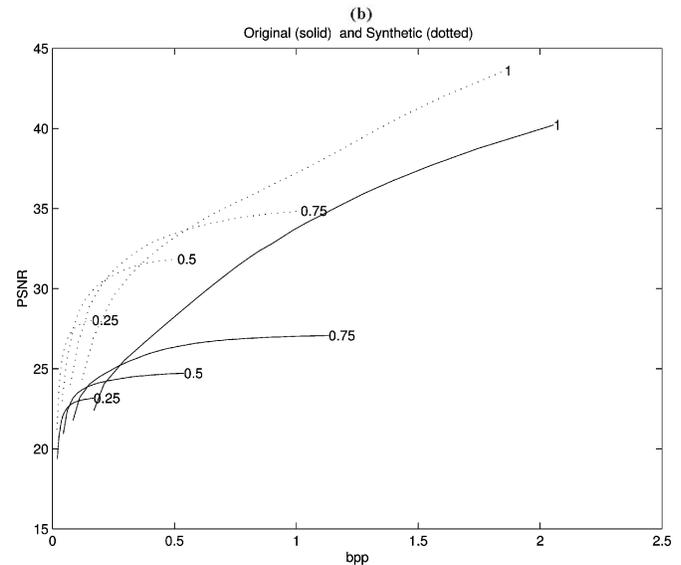
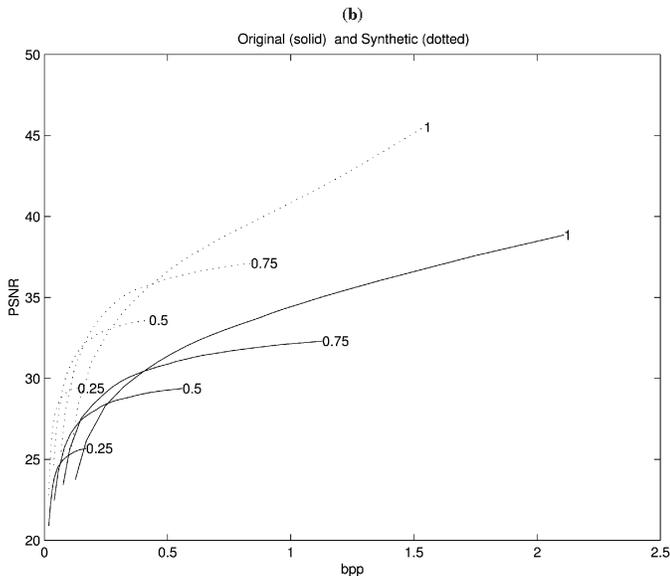
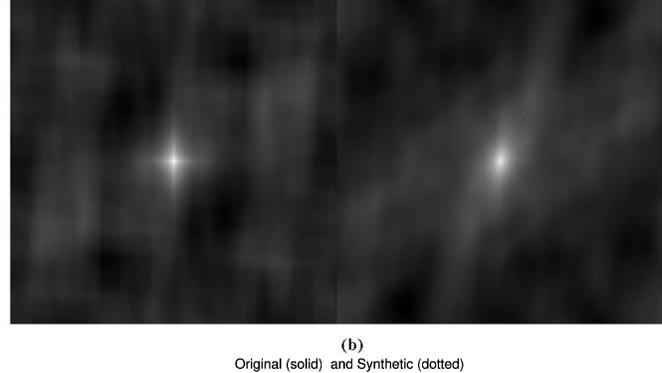
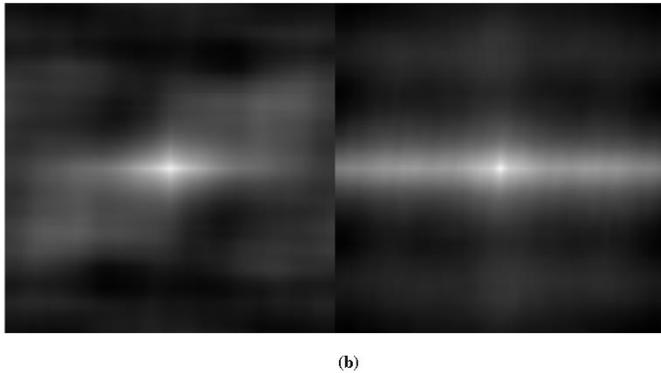
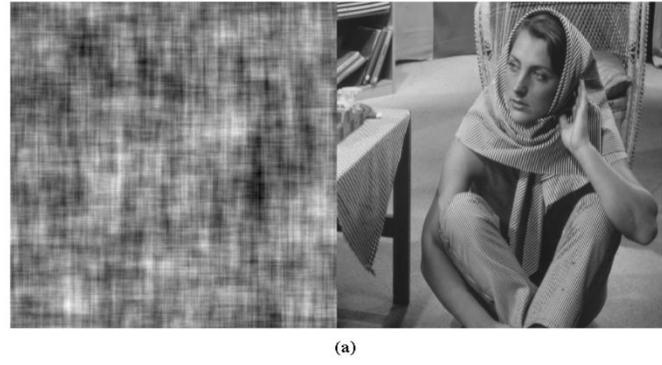


Fig. 8. Comparison between a natural and a synthesized image with similar autocorrelation ($\alpha_x = 39$, $\alpha_y = 91$).

This was the estimation procedure implemented in our experiments.

C. Experimental Results

A JPEG compression performance comparison for a natural image and its random synthesized version is shown in Fig. 6 for a 256×256 image. The examples in Figs. 7–9, are of 512×512 images. The figures show the compression results of synthetic versus natural images with similar statistics. Synthetic and original images and their corresponding autocorrelations are presented with their corresponding JPEG PSNR/bpp compression curves for 4 down-sampling factors. Fig. 9 presents an image for which the statistical model is highly inaccurate due to the

Fig. 9. Comparison between a natural and a synthesized image with similar autocorrelation ($\alpha_x = 42$, $\alpha_y = 26$).

diagonal texture that characterizes a relatively large part of the “Barbara” image. In all these examples we specify the estimated values of α_x and α_y . Note that since these images are of size 256×256 , the values are to be multiplied by 2 if they are to be compared to the range of α suggested in Section V.

The above experiments indicate that the crossing locations between down-sampling factors in the synthetic images appear to be a good approximation of the crossings in the natural images. Thus, based on the second order statistics of the image we can predict the optimal down-sampling factor. Moreover, the nonstationarity nature of images has a relatively minor impact on the optimal down-sampling factor. This is evident from the alignment of the results of the natural and the synthetic images. There appears to be a vertical gap (in PSNR) between the syn-

thetic and the natural images. However, similar PSNR gaps also appear between different synthetic images. Based on the above two observations (ability to use stationary model, and ability to refer to the second moment only), we assert that one can predict the best down-sampling factor based on the model proposed in this paper.

VI. CONCLUSIONS

This paper started with the observation that a the use of down-sampling prior to JPEG coding can improve the overall coding performance both objectively and subjectively. We then presented an analytical model to explain this phenomenon. A set of experiments are shown to verify this model and support the idea of down-sampling before transform coding for optimal compression. Based on the theoretical developments above, a simple algorithm can be developed to set the optimal down-sampling for a given image, based on the image statistics, size, and bit budget available. Further work is required in order explore extensions and implementation issues, such as an efficient estimation for the image statistics, extraction of second order statistics locally and using an hierarchical slicing of the image to various block sizes, and more.

The emerging new standard for image compression, JPEG-2000, is based on coding with a wavelet transform (see [2] and [7]). This new algorithm is known to outperform the regular JPEG. Among the many reasons to this performance improvement, is the fact that wavelet-based coding applies a multiresolution analysis of the underlying image. In this sense, the work presented here proposes the introduction of a simple multiscale feature into the JPEG standard, thereby gaining compression ratio. Further work is required to replace the approach presented here to a locally adaptive one, as is done naturally by the wavelet coders.

APPENDIX

To compute the second order statistics of the coefficients we need to carry out the following integral

$$\iint_0^1 e^{-A|x-\xi|} \cos(k\pi x) \cos(l\pi\xi) dx d\xi \equiv \mathcal{M}(A; k, l). \quad (40)$$

We shall separate the integration into two parts

$$\int_0^1 \left[\int_0^x e^{-A(x-\xi)} \cos(k\pi x) \cos(l\pi\xi) d\xi \right] dx + \int_0^1 \left[\int_0^\xi e^{-A(\xi-x)} \cos(k\pi x) \cos(l\pi\xi) dx \right] d\xi$$

i.e., $\mathcal{M}(A; k, l) = \mathcal{M}_1(A; k, l) + \mathcal{M}_2(A; k, l)$. Note however that $\mathcal{M}_2(A; k, l) = \mathcal{M}_1(A; l, k)$, hence, $\mathcal{M}(A; k, l) = \mathcal{M}_1(A; k, l) + \mathcal{M}_1(A; l, k)$. Hence, after some algebraic steps we obtain

$$\begin{aligned} \mathcal{M}_1(A; k, l) = & \frac{A}{A^2 + (l\pi)^2} \frac{1}{2} (1 + \delta_{k-or-l}) \delta_{|k-l|} \\ & + \frac{l\pi}{A^2 + (l\pi)^2} \\ & \cdot \frac{1}{2} \left[\frac{1}{(l+k)\pi} (1 - (-1)^{k+l}) \right. \\ & \quad \left. + \frac{1}{(l-k)\pi} (1 - (-1)^{l-k}) \right] \\ & - \frac{A^2}{(A^2 + (l\pi)^2)(A^2 + (k\pi)^2)} \\ & \times (1 - e^{-A}(-1)^k). \end{aligned} \quad (41)$$

Returning to the expression for $\mathcal{M}(A; k, l)$ we get

$$\begin{aligned} \mathcal{M}(A; k, l) = & \mathcal{M}_1(A; k, l) + \mathcal{M}_1(A; l, k) \\ = & \frac{A}{A^2 + (l\pi)^2} \frac{1}{2} (1 + \delta_{k-or-l}) \delta_{|k-l|} \\ & + \frac{A}{A^2 + (k\pi)^2} \frac{1}{2} (1 + \delta_{l-or-k}) \delta_{|l-k|} \\ & + \frac{l\pi}{A^2 + (l\pi)^2} \\ & \cdot \frac{1}{2} \left[\frac{1}{(l+k)\pi} (1 - (-1)^{k+l}) \right. \\ & \quad \left. + \frac{1}{(l-k)\pi} (1 - (-1)^{l-k}) \right] \\ & + \frac{k\pi}{A^2 + (k\pi)^2} \\ & \cdot \frac{1}{2} \left[\frac{1}{(k+l)\pi} (1 - (-1)^{k+l}) \right. \\ & \quad \left. + \frac{1}{(k-l)\pi} (1 - (-1)^{k-l}) \right] \\ & + \frac{A^2}{(A^2 + (l\pi)^2)(A^2 + (k\pi)^2)} \\ & \times (1 - e^{-A}(-1)^k + 1 - e^{-A}(-1)^l). \end{aligned} \quad (42)$$

Simplifying this leads to (43), shown at the bottom of the page. To check this formula consider

$$\mathcal{M}(A; 0, 0) = \frac{2}{A} + 0 - \frac{1}{A^2} (2 - 2e^{-A})$$

$$\begin{aligned} \mathcal{M}(A; k, l) = & \frac{A}{A^2 + k^2\pi^2} \cdot (1 + \delta_{k-or-l}) \delta_{|k-l|} \\ & + \begin{cases} 0 & \text{if } l = k \text{ or } l + k \text{ is even} \\ \frac{2A^2}{(A^2 + (l\pi)^2)(A^2 + (k\pi)^2)} & \text{if } l + k \text{ is odd} \end{cases} \\ & - \frac{A^2}{(A^2 + (l\pi)^2)(A^2 + (k\pi)^2)} \left(2 - e^{-A} [(-1)^k + (-1)^l] \right). \end{aligned} \quad (43)$$

and indeed,

$$\begin{aligned}
 \int_0^1 \int_0^1 e^{-A|x-\xi|} dx d\xi &= \int_0^1 \left[\int_0^x e^{-Ax} e^{A\xi} d\xi \right] dx \\
 &\quad + \int_0^1 \left[\int_0^\xi e^{-A\xi} e^{Ax} dx \right] d\xi \\
 &= \int_0^1 e^{-Ax} \left[\frac{1}{A} e^{A\xi} \right]_0^x dx \\
 &\quad + \int_0^1 e^{-A\xi} \left[\frac{1}{A} e^{Ax} \right]_0^\xi d\xi \\
 &= 2 \int_0^1 e^{-Ax} \left[\frac{1}{A} e^{Ax} - \frac{1}{A} \right] dx \\
 &= 2 \cdot \int_0^1 \left(\frac{1}{A} - \frac{1}{A} e^{-Ax} \right) dx \\
 &= 2 \cdot \left(\frac{1}{A} - \frac{1}{A} \left[-\frac{e^{-Ax}}{A} \right]_0^1 \right) \\
 &= 2 \cdot \left(\frac{1}{A} - \frac{1}{A} \left[\frac{1}{A} - \frac{1}{A} e^{-A} \right] \right) \\
 &= \frac{2}{A} - \frac{1}{A^2} (2 - 2e^{-A}).
 \end{aligned}$$

This agrees with the general expression we got above.

ACKNOWLEDGMENT

The authors thank Dr. M. Rudzsky and A. Spira for intriguing discussions. They also thank Y. Katz for translating some of their never-ending equations into LaTeX. Also, they thank the reviewers of this paper and the associate editor for their valuable and detailed comments and suggestions for improvements of the presentation of the paper.

REFERENCES

- [1] W. B. Pennebaker and J. L. Mitchell, *JPEG Still Image Compression Standard*. New York: Van Nostrand Reinhold, 1992.
- [2] D. Taubman, "High performance scalable image compression with EBCOT," *IEEE Trans. Image Processing*, vol. 9, pp. 1158–1170, July 2000.
- [3] A. Gersho and R. M. Gray, *Vector Quantization and Signal Compression*. Norwell, MA: Kluwer, 1992.
- [4] A. Bruckstein, "On optimal image digitization," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 35, no. 4, pp. 553–555, 1987.
- [5] —, "On soft bit allocation," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 35, no. 5, pp. 614–617, 1987.
- [6] A. K. Jain, *Fundamentals of Digital Image Processing*. Englewood Cliffs, NJ: Prentice-Hall, 1989.
- [7] T. Ebrahimi and C. Christopoulos, "JPEG 2000 standard: Still image compression scheme of 21st century," in *JPEG2000 Tutorial*, the European Signal Processing Conference (EUSIPCO 2000), Tampere, Finland, Sept. 5–8, 2000.
- [8] "JPEG-2000 Image Coding System WG1N390 REV.," ISO/IEC JTC 1/SC 29/WG 1.
- [9] [Online]. Available: <http://public.migrator2000.org/J2Kdemonstrator>

Alfred M. Bruckstein received the B.Sc. degree (with honors) and the M.Sc. degree in electrical engineering from The Technion—Israel Institute of Technology, Haifa, and the Ph.D. degree in electrical engineering from Stanford University, Stanford, CA, in 1977, 1980, and 1984, respectively.

Since 1985, he has been a Faculty Member of the Computer Science Department at The Technion—Israel Institute of Technology, where he currently a Full Professor, holding the Ollendorff Chair. During the summers from 1986 to 1995 and from 1998 to 2000 he was a Visiting Scientist at Bell Laboratories. He is in the editorial boards of *Pattern Recognition*, *Imaging Systems and Technology*, *Circuits Systems*, and *Signal Processing*. He also served as a member of program committees of 20 conferences. His research interests are in Image and signal processing, computer vision, computer graphics, pattern recognition, robotics, especially ant robotics, applied geometry, estimation theory and inverse scattering, and neuronal encoding process modeling.

Dr. Bruckstein is a member of SIAM, AMS, and MM. He was awarded the Rothschild Fellowship for Ph.D. Studies at Stanford, Taub Award, Theeman Grant for a scientific tour of Australian Universities, Hershel Rich Technion Innovation Award, and Hershel Rich Innovation Award.

Michael Elad received the Bs.C. (with honors), Ms.C., and Ds.C. degrees from the Electrical Engineering Department of The Technion—Israel Institute of Technology, Haifa, Israel, in 1986, 1988, and 1996, respectively.

From 1988 to 1993, he served as an R&D Officer in the Israeli Air-Force. From 1997 to 1999, he was a Researcher at the Hewlett-Packard Laboratories-Israel (HPL-I). During 2000–2001 he led the research division in Jigami Corporation, Israel. In parallel, during 1997–2001 he was a Lecturer in the Electrical Engineering Department at The Technion. He is now with the Computer Science Department (SCCM program) at Stanford University, Stanford, CA, as a Research Associate. His research interests include inverse problems, signal representations, and numerical algorithms in the areas of signal processing, image processing, and computer vision. He worked on super-resolution reconstruction of images, motion estimation, nonlinear filtering, sparse representations for signals, target detection in images, polar Fourier transform, and image compression.

Dr. Elad was awarded the Wolf, Gutwirth, and Ollendorff Fellowships. He was awarded Best Lecturer in 2000 and 2001.

Ron Kimmel received the B.Sc. degree (with honors) in computer engineering in 1986, the M.S. degree in 1993 in electrical engineering, and the D.Sc. degree in 1995, all from The Technion—Israel Institute of Technology, Haifa.

During 1986–1991 he served as an R&D Officer in the Israeli Air Force. During 1995–1998, he was a Postdoctoral Fellow with the Computer Science Division of Berkeley Labs, and the Mathematics Department, University of California, Berkeley. Since 1998, he has been a Faculty Member of the Computer Science Department at The Technion, where he is currently an Associate Professor. His research interests are in computational methods and their applications in differential geometry, numerical analysis, image processing and analysis, computer aided design, robotic navigation, and computer graphics. He was a Consultant with the HP Research Lab in image processing and analysis during 1998–2000, and to Net2Wireless/Jigami Research Group during 2000–2001. He has been is on the advisory board of MediGuide (biomedical imaging) since 2001, and has been on various program and organizing committees of conferences, workshops, and journal editorial boards, in the fields of image synthesizing, processing, and analysis.

Dr. Kimmel was awarded the Hershel Rich Technion Innovation Award (twice), the Henry Taub Prize for excellence in research, Alon Fellowship, the HTI Postdoctoral Fellowship, and the Wolf, Gutwirth, Ollendorff, and Jury Fellowships.