Postprocessing of Compressed Images via Sequential Denoising

Yehuda Dar, Alfred M. Bruckstein, Michael Elad, and Raja Giryes

***Abstract*—In this work we propose a novel postprocessing technique for compression-artifact reduction. Our approach is based on posing this task as an inverse problem, with a regularization that leverages on existing state-of-the-art image denoising algorithms. We rely on the recently proposed Plug-and-Play Prior framework, suggesting the solution of general inverse problems via Alternating Direction Method of Multipliers (ADMM), leading to a sequence of Gaussian denoising steps. A key feature in our scheme is a linearization of the compression-decompression process, so as to get a formulation that can be optimized. In addition, we supply a thorough analysis of this linear approximation for several basic compression procedures. The proposed method is suitable for diverse compression techniques that rely on transform coding. Specifically, we demonstrate impressive gains in image quality for several leading compression methods - JPEG, JPEG2000, and HEVC.**

***Index Terms*—Lossy Compression, Postprocessing, Deblocking, Denoising, Image Restoration, Plug-and-Play Prior.**

I. Introduction

BANDWIDTH and memory constraints play a crucial role in transmission and storage systems. Various compression methods are available in order to meet severe constraints on the bit-cost in data representation. While some applications require perfect reconstruction, some may tolerate inaccuracies and can benefit from a reduced representation-cost. The latter approach is known as lossy compression and is widely used for representing a signal under bit-budget constraints while allowing some errors in recovery. Accordingly, a variety of techniques were standardized over the years for the lossy compression of acoustic and visual signals.

Since lossy compression allows discrepancies between the original and the reconstructed signals, the differences being intentionally used in tradeoffs between bit-rate and quality. The nature of the created artifacts depends on the compression architecture. For example, block-based image compression techniques suffer from blockiness effects that increase and degrade the reconstruction as the bit-rate is reduced.

As artifacts are inherent in the lossy compression of signals, a great number of artifact-reduction techniques were proposed over the years (e.g., [1]–[18] for image compression). These methods usually focus on specific signal types (e.g., image, video or audio) and sometimes even on specific artifacts corresponding to certain compression designs (e.g., deblocking procedures for images). Common image compression techniques rely on transform-coding, where image blocks are transformed, and the resultant transform-coefficients are quantized according to their relative importance. The prominent artifacts of this architecture are [19]: blockiness due to the separate treatment of non-overlapping blocks; ringing caused by the effective elimination of high frequency components, expressed as contours spreading along sharp edges; and blurring that results from high-frequency information loss. Postprocessing of compressed images are subcategorized into two approaches [19]: enhancement of the deteriorated signal by smoothing its artifacts (e.g., [1], [3]), and restoration of the original signal samples (e.g., [6], [15]).

In this work we propose a novel postprocessing technique for compression artifact reduction by a regularized restoration of the original (precompressed) signal. Specifically, we formulate the compression postprocessing procedure as a regularized inverse-problem for estimating the original signal given its reconstructed form. We also approximate the (nonlinear!) compression-decompression process by a linear operator, so as to obtain a tractable inverse problem formulation. The intriguing approach of locally linearizing the non-differentiable compression procedures is carefully analyzed, in order to utilize it properly. Whereas many studies focus on corrections of specific artifacts (e.g., image deblocking techniques [1], [3], [5], [13]), our approach attempts to generally restore the signal and thus implicitly repairs multiple artifacts. The major strength of our method comes from the regularization used, as we next explain.

Afonso et al. [20] proposed to efficiently solve regularized inverse-problems in image processing using the Alternating Direction Method of Multipliers (ADMM) [21]. Their approach decouples the inversion and the regularization parts of the optimization problem, which is in turn iteratively solved. Venkatakrishnan et al. [22] further developed the use of the ADMM by showing an equivalence between the regularization step and denoising optimization problems. Their framework, called "Plug-and-Play Priors", is flexible, proposing the replacement of the regularization step by a general-purpose Gaussian image denoiser. In this work we propose a compression postprocessing algorithm by employing this scheme. Furthermore, as denoising algorithms relying on sparse models were found to be highly effective ones (e.g., K-SVD [23], [24], BM3D [25]), we utilize a leading denoiser from this category.

Since we propose a method of postprocessing for a variety of lossy-compression techniques, the algorithm and its analysis are dealt within an abstract and general setting. Following

EDICS: Restoration and Enhancement (TEC-RST), Lossy Coding of Images and Video (COM-LOC).

Y. Dar, A. M. Bruckstein, and M. Elad are with the Department of Computer Science, Technion, Israel. R. Giryes is with the Department of Electrical and Computer Engineering, Duke University, USA.

The research leading to these results has received funding from the European Research Council under European Unions Seventh Framework Program, ERC Grant agreement no. 320649.

that, a thorough demonstration for image compression is provided. Specifically, we show results for the leading image compression standards: JPEG [26], JPEG2000 [27] and the still-image profile of the HEVC [28], [29], offering the state-of-the-art performance [30]. While these three compression methods rely on a block-based architecture and a transform-coding approach, they differ as follows: JPEG operates on 8x8 blocks and applies a discrete cosine transform (DCT); JPEG2000 works on large blocks (tiles) of at least 128x128 pixels and utilizes a discrete wavelet transform (DWT); in HEVC-stills the image is split into coding blocks that are further partitioned using a quadtree structure, then intra-prediction is performed and transform coding is applied on the prediction residuals (where the transform is mainly integer-approximations of the DCT at various sizes). Our method is evaluated for a diversified set of compression algorithms that span the range of the contemporary coding concepts. Moreover, our postprocessing technique achieves significant gains and usually outperforms the cutting-edge methods for the examined compression standards.

This paper is organized as follows. In section II the proposed postprocessing method is presented. In section III, the compression linearization is mathematically analyzed for simplified cases of quantization and transform coding. Section IV presents image-compression experimental results and compares them to those of competitive techniques. Section V concludes this paper.

II. The Proposed Postprocessing Strategy

A. Problem Formulation using ADMM

Let us consider a signal $\mathbf{x} \in \mathbb{R}^N$ that undergoes a compression-decompression procedure, $C : \mathbb{R}^N \to \mathbb{R}^N$, resulting in the reconstructed signal $\mathbf{y} = C(\mathbf{x})$. For lossy compression methods an error is introduced at a size that depends on the bit-budget, the specific-signal characteristics, and the compression algorithm. We aim at restoring the precompressed signal $\mathbf{x}$ from the reconstructed $\mathbf{y}$ using the following regularized inverse-problem:

$$\hat{\mathbf{x}} = \arg\min_{\mathbf{x}} \ \|\mathbf{y} - C(\mathbf{x})\|_2^2 + \beta s(\mathbf{x}), \tag{1}$$

where $s(\cdot)$ is a regularizer, which can be associated with a given Gaussian denoiser, weighted by the parameter β.[1]

Similar to [20] and [22], we develop an iterative algorithm for the solution of (1). We start by applying variable splitting that yields the following equivalent form of (1):

$$\begin{aligned} &\min_{\mathbf{x},\mathbf{v}} \ \|\mathbf{y} - C(\mathbf{x})\|_2^2 + \beta s(\mathbf{v}) \\ &\text{subject to } \mathbf{x} = \mathbf{v}, \end{aligned} \tag{2}$$

where $\mathbf{v} \in \mathbb{R}^N$ is an additional vector due to the split. The constrained problem (2) is addressed by forming an augmented Lagrangian and its corresponding iterative solution (of its scaled version) via the method of multipliers [21, ch. 2], where the i^{th} iteration consists of

$$\begin{aligned} (\hat{\mathbf{x}}_i, \hat{\mathbf{v}}_i) &= \arg\min_{\mathbf{x},\mathbf{v}} \|\mathbf{y} - C(\mathbf{x})\|_2^2 + \beta s(\mathbf{v}) \\ &\qquad + \frac{\lambda}{2}\|\mathbf{x} - \mathbf{v} - \mathbf{u}_i\|_2^2 \\ \mathbf{u}_{i+1} &= \mathbf{u}_i + (\hat{\mathbf{x}}_i - \hat{\mathbf{v}}_i). \end{aligned} \tag{3}$$

Here $\mathbf{u}_i \in \mathbb{R}^n$ is the scaled dual-variable and λ is an auxiliary parameter, both introduced in the Lagrangian.

Please note the following notation remark for a general vector $\mathbf{u}$. First, $\mathbf{u}_i$ stands for vector $\mathbf{u}$ in the i^{th} iteration. On the other hand, u_j represents the j^{th} component (a scalar) of the vector $\mathbf{u}$. Finally, $\mathbf{u}_i^{(j)}$ denotes the j^{th} element of the vector $\mathbf{u}_i$.

Approximating the joint optimization of $\mathbf{x}$ and $\mathbf{v}$ in (3), using one iteration of alternating minimization, results in the iterative solution in the ADMM form, where the i^{th} iteration consists of

$$\hat{\mathbf{x}}_i = \arg\min_{\mathbf{x}} \|\mathbf{y} - C(\mathbf{x})\|_2^2 + \frac{\lambda}{2}\|\mathbf{x} - \tilde{\mathbf{x}}_i\|_2^2 \tag{4}$$

$$\hat{\mathbf{v}}_i = \arg\min_{\mathbf{v}} \frac{\lambda}{2}\|\mathbf{v} - \tilde{\mathbf{v}}_i\|_2^2 + \beta s(\mathbf{v}) \tag{5}$$

$$\mathbf{u}_{i+1} = \mathbf{u}_i + (\hat{\mathbf{x}}_i - \hat{\mathbf{v}}_i). \tag{6}$$

Here $\tilde{\mathbf{x}}_i = \hat{\mathbf{v}}_{i-1} - \mathbf{u}_i$ and $\tilde{\mathbf{v}}_i = \hat{\mathbf{x}}_i + \mathbf{u}_i$.

The regularization step (5) is of the form of a Gaussian denoising optimization-problem (of a noise level determined by β/λ) and therefore can be viewed as applying a denoising algorithm to the signal $\tilde{\mathbf{v}}_i$. Indeed, the Plug-and-Play Priors framework [22] suggests exactly this strategy, replacing (5) with an independent denoiser; even one that does not explicitly have in its formulation a minimization problem of the form of (5). The deployment of a favorable denoiser introduces valuable practical benefits to the design of the proposed post-processing procedure, and yields a powerful generic method.

B. Linear Approximation of the Compression-Decompression Procedure

Due to the high nonlinearity of $C(\mathbf{x})$, we further simplify the forward-model step (4) using a first-order Taylor approximation of the compression-decompression function around $\hat{\mathbf{x}}_{i-1}$, i.e.,

$$C_{lin}(\mathbf{x}) = C(\hat{\mathbf{x}}_{i-1}) + \left.\frac{dC(\mathbf{z})}{d\mathbf{z}}\right|_{\mathbf{z}=\hat{\mathbf{x}}_{i-1}} \cdot (\mathbf{x} - \hat{\mathbf{x}}_{i-1}) \tag{7}$$

where $\left.\frac{dC(\mathbf{z})}{d\mathbf{z}}\right|_{\mathbf{z}=\hat{\mathbf{x}}_{i-1}}$ is the $N \times N$ Jacobian matrix of the compression-decompression at the point $\hat{\mathbf{x}}_{i-1}$.

Since the approximation of the Jacobian, $\frac{dC(\mathbf{z})}{d\mathbf{z}}$, deeply influences the restoration result and the computational cost, this is a quite delicate task. First, C is a non-linear and even non-differentiable function as the compression often relies on quantization and/or thresholding. Second, we provide here a generic technique, and therefore do not explicitly consider the compression-decompression formulation.

[1] Throughout this paper we shall assume for simplicity that the distortion is modeled as a white additive Gaussian noise, leading to the ℓ_2 term used here. We should note, however, that our scheme could be improved by using a better modeling of the compression-decompression error, such as an ℓ_∞ on the transform coefficients w.r.t. the quantization step-size.

Theoretically, the fact that C is non-differentiable would prevent us from using its Jacobian for the optimization. However, as its Jacobian has only a finite number of singularities, in practice we can rely on it as is done in other fields, e.g., in the training of neural networks that are composed of concatenations of non-differentiable non-linear operations [31].

For calculating the entries of the Jacobian, we rely on the standard definition of the derivative, assuming that C is locally linear. We justify this approach in the next section. As we might be approximating the derivative in the neighborhood of a non-differential point, we take several step-sizes in the calculation of the derivative and average over all of them. This leads to the following approximation to the k^{th} column of the Jacobian:

$$\frac{dC\left(\mathbf{z}\right)}{dz_k} = \frac{1}{|S_\delta|}\sum_{\delta\in S_\delta}\frac{C(\mathbf{z}+\delta\cdot\mathbf{e}_k)-C(\mathbf{z}-\delta\cdot\mathbf{e}_k)}{2\delta}, \quad (8)$$

where $\mathbf{e}_k$ is the k^{th} standard direction vector, and S_δ is a set of step lengths for approximating the derivative using the standard definition (the set size is denoted as $|S_\delta|$).

Due to the high nonlinearity of C, the linear approximation (7) is reasonable in a small neighborhood around the approximating point $\hat{\mathbf{x}}_{i-1}$. Accordingly, we further constrain the distance of the solution from the linear-approximation point by modifying (4) to

$$\hat{\mathbf{x}}_i = \arg\min_{\mathbf{x}} \left\|\mathbf{y}-\tilde{C}\left(\mathbf{x}\right)\right\|_2^2 + \frac{\lambda}{2}\left\|\mathbf{x}-\tilde{\mathbf{x}}_i\right\|_2^2 + \mu\left\|\mathbf{x}-\hat{\mathbf{x}}_{i-1}\right\|_2^2. \quad (9)$$

The proposed generic method is summarized in Algorithm 1.

Algorithm 1 The Proposed Postprocessing Method

$\hat{\mathbf{x}}_0 = \mathbf{y}$, $\hat{\mathbf{v}}_0 = \mathbf{y}$
$i = 1$, $\mathbf{u}_1 = \mathbf{0}$
repeat
  Approximate $C_{lin}(\cdot)$ around $\hat{\mathbf{x}}_{i-1}$ using (7) and (8)
  $\tilde{\mathbf{x}}_i = \hat{\mathbf{v}}_{i-1} - \mathbf{u}_i$
  $\hat{\mathbf{x}}_i = \arg\min_{\mathbf{x}} \|\mathbf{y}-C_{lin}\left(\mathbf{x}\right)\|_2^2 + \frac{\lambda}{2}\|\mathbf{x}-\tilde{\mathbf{x}}_i\|_2^2 + \mu\|\mathbf{x}-\hat{\mathbf{x}}_{i-1}\|_2^2$
  $\tilde{\mathbf{v}}_i = \hat{\mathbf{x}}_i + \mathbf{u}_i$
  $\hat{\mathbf{v}}_i = \text{Denoise}_{\beta/\lambda}\left(\tilde{\mathbf{v}}_i\right)$
  $\mathbf{u}_{i+1} = \mathbf{u}_i + (\hat{\mathbf{x}}_i - \hat{\mathbf{v}}_i)$
  $i \leftarrow i+1$
until stopping criterion is satisfied

III. Linear Approximation – A Closer Look

The wide variety of lossy-compression methods yield a range of diverse compression performances and features. These often rely on the fundamental procedure of quantization, which enables to trade-off representation-precision and cost (in bits). The quantization concept is employed in various forms, e.g., as a scalar/vector operation, using uniform/non-uniform representation levels, and also in the extreme case of thresholding where some data elements are completely discarded (and the remaining are regularly quantized). Furthermore, the statistical properties of the data affect the quantizer performance, and indeed, the prevalent transform-coding concept first considers the data in different orthogonal basis that enables more efficient quantization.

In this section we study the linear approximation of the scalar-quantization procedure, starting at its use for a single variable and proceeding to transform coding of vectors, where the transform-domain coefficients are independently quantized. The analysis provided here sheds some clarifying light on our linearization strategy that is generically applied in the proposed technique to more complicated compression methods.

A. Local Linear-Approximation of a Quantizer

Let us consider a general scalar quantization function $q(x)$ that maps the real-valued input x onto a discrete set of real-valued representation levels. As $q(x)$ is a non-differentiable function we examine its linear approximation around the point x_0 in a limited interval defined by δ as

$$\eta(x_0,\delta) = \left[x_0-\delta, x_0+\delta\right].^2 \quad (10)$$

The studied approximation takes the general linear form of

$$\tilde{q}\left(x\right) = ax+b \quad (11)$$

where a and b are the linearization parameters. The approximation in (11) introduces an error that can be measured via the Mean-Squared-Error (MSE) over the local interval $\eta(x_0,\delta)$ [3]:

$$LMSE^{SQ}\left(a,b;\eta\left(x_0,\delta\right)\right) \triangleq \frac{1}{2\delta}\int_{x_0-\delta}^{x_0+\delta}\left(q\left(x\right)-\tilde{q}\left(x\right)\right)^2 dx \quad (12)$$

Substituting (11) in (12), then demanding parameter optimality by

$$\frac{\partial}{\partial a}LMSE^{SQ}\left(a,b;\eta\left(x_0,\delta\right)\right)=0 \quad (13)$$
$$\frac{\partial}{\partial b}LMSE^{SQ}\left(a,b;\eta\left(x_0,\delta\right)\right)=0,$$

leads to the following optimal parameters:

$$a^* = \frac{3}{\delta^2}\left(L_a - L_b x_0\right) \quad (14)$$

$$b^* = L_b - \frac{3x_0}{\delta^2}\left(L_a - L_b x_0\right) \quad (15)$$

[2]Here we mathematically study the problem for a given quantization function that is used for calculating the approximation, and therefore a single δ value is sufficient. However, in our generic algorithm we empirically utilize a set of δ values in Equation (8) since the compression function is unknown.

[3]We study the linearization error as a function of the approximation-interval size, which is determined by δ. Clearly, by setting a sufficiently small δ we get a zero approximation-error as we shall see hereafter. However, note that the linearization error is not the only factor to consider for the selection of δ. Therefore, we should bear in mind throughout the following derivation that we do not present here an explicit method for selecting the value of δ but an analysis of the local approximation-error of the quantizer as a function of δ. Nevertheless, this mathematical analysis demonstrates the important principles of linear approximation of quantizers and motivates the algorithmic design and experimental settings that are presented in the following sections.

where we defined

$$L_a \triangleq \frac{1}{2\delta}\int_{x_0-\delta}^{x_0+\delta} xq\left(x\right)dx \tag{16}$$

$$L_b \triangleq \frac{1}{2\delta}\int_{x_0-\delta}^{x_0+\delta} q\left(x\right)dx. \tag{17}$$

To better understand the values of these parameters, we shall consider several simple cases.

B. The Case of Two-Level Quantization

We start by studying the elementary two-level quantizer that takes the form of a step function (Fig. 1a) as follows:

$$q_2\left(x\right) = \begin{cases} -1/2 & ,\text{for } x \leqslant 0 \\ 1/2 & ,\text{for } x > 0 \end{cases} \tag{18}$$

where the two output levels, $r_0 = -1/2$ and $r_1 = 1/2$, are assigned according to the input sign. This canonic form is useful to our discussion here, since it is an asymmetric function around the origin, and thus, will simplify the mathematical analysis. Nevertheless, the form in (18) can be extended to any two-level quantizer using shifts and scaling that adjust the step-location and the two representation levels. Accordingly, the results in this section are easily extended, e.g., by considering the quadratic effect of the step-size scaling on the local MSE.

When the local interval is completely contained within a single decision region, i.e. $\eta(x_0,\delta) \subset [-\infty, 0]$ or $\eta(x_0,\delta) \subset (0,\infty]$, then $q_2\left(x\right)$ is locally fixed on r_0 or r_1, respectively, and therefore

$$L_a = \frac{1}{2\delta}\int_{x_0-\delta}^{x_0+\delta} xr_i dx = x_0 r_i \tag{19}$$

$$L_b = \frac{1}{2\delta}\int_{x_0-\delta}^{x_0+\delta} r_i dx = r_i \tag{20}$$

for the respective $i \in \{0,1\}$. Then setting (19) and (20) in (14) and (15), respectively, induces the optimal values $a^* = 0$ and $b^* = r_i$, that of course accurately represent the locally flat function with a corresponding zero local-MSE.

Now we turn to the more interesting case where the local interval spans over the two decision regions, i.e., $x_0 - \delta < 0$ and $x_0 + \delta > 0$. Calculating again the optimal parameter set (14)-(15) for this scenario requires to decompose the integrals (16)-(17) to the two decision regions, yielding the following optimal linearization parameters:

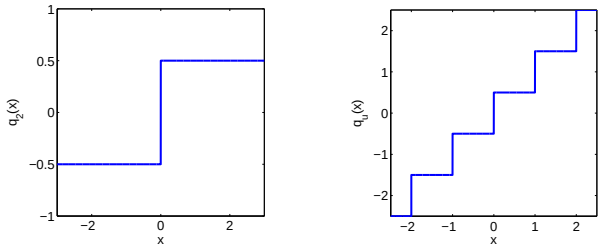
$$a^* = \frac{3}{4\delta}\left(1 - \left(\frac{x_0}{\delta}\right)^2\right) \tag{21}$$

$$b^* = \frac{3x_0}{4\delta}\left(\left(\frac{x_0}{\delta}\right)^2 - \frac{1}{3}\right) \tag{22}$$

and, using (12), the corresponding error (for $\delta > |x_0|$) is[4]

$$LMSE^{SQ}\left(a^*, b^*; \eta\left(x_0,\delta\right)\right) = \frac{1}{16}\left(1 + 3\left(\frac{x_0}{\delta}\right)^2\right)\left(1 - \left(\frac{x_0}{\delta}\right)^2\right). \tag{23}$$

[4]Recall that for $\delta < |x_0|$ the approximation local-MSE is zero.



(a) Two-level (b) Uniform

Fig. 1. Examples of scalar quantizers.

One should note that on the limit of the global linear approximation, i.e., when $\delta \to \infty$, the optimal parameters are

$$\lim_{\delta\to\infty} a^* = 0 \tag{24}$$

$$\lim_{\delta\to\infty} b^* = 0. \tag{25}$$

This asymptotic fitting to a constant-valued function is also expressed in the numerical results in Fig. 2a-2b.

Let us study the optimal approximation for the non-trivial case of $\delta > |x_0|$. First, we notice that the error tends to zero as δ gets closer to $|x_0|$. Second, The maximal error is obtained for $\delta = \sqrt{3}|x_0|$ and its value is $\frac{1}{12}$. Moreover, for approximation around the non-differentiable point, i.e. $x_0 = 0$, the error is a constant and, therefore, independent of δ. This interesting observation is a special case of a more general behavior where a constant error value is achieved for any (x_0, δ) pair that is on the line $\delta = c|x_0|$ for some $c \in [0,1]$. This constant local-MSE is due to the fixed ratio between the lengths of the subintervals $[x_0 - \delta, 0]$ and $[0, x_0 + \delta]$, determining the optimal approximation in this case. The latter analysis is clearly exhibited in the numerical results in Fig. 2.

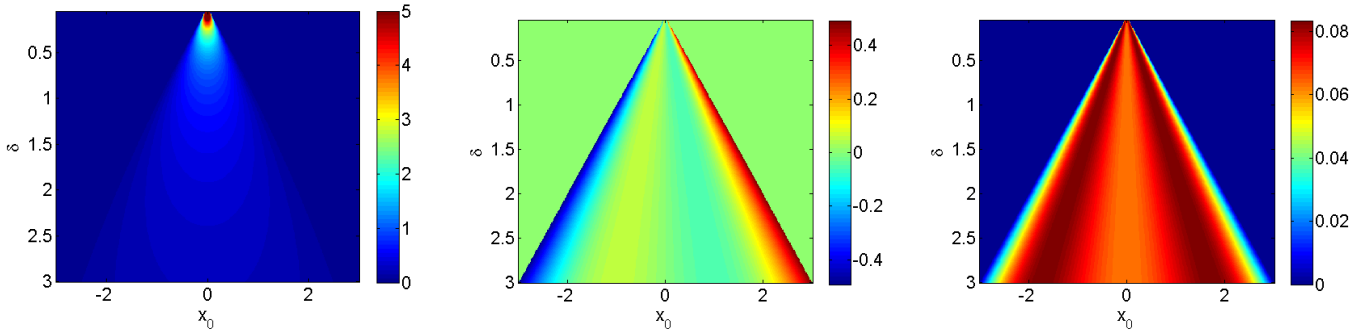
The numerical results also demonstrate the following behavior of the approximation as function of δ. At the beginning, the solution gradually considers the step by having an increasingly steeper slope, then, the approximation begins to approach the asymptotic solution of a flat line. It is also observed that the approximation is useful (in terms of relatively low error) when the interval size tends to be the minimal that contains the discontinuity point, located here at 0. Furthermore, in some sense, finding the best interval for approximating around $x_0 \neq 0$ is like measuring the distance of x_0 from the step.

C. The Case of Multi-Level Uniform Quantization

Let us extend the above analysis to a multi-level uniform quantizer in the mid-riser form [32, p. 137] (Fig. 1b):

$$q_u\left(x\right) = \lfloor x \rfloor + \frac{1}{2}. \tag{26}$$

Here the quantization step is of unit length, and accordingly the i^{th} decision region, $[d_i, d_{i+1}) = [i, i+1)$, maps the input to the i^{th} representation level $r_i = i + \frac{1}{2}$. Note that i is an integer that may be positive or negative. As in the previous case, this normalized quantizer form yields a simplified analysis that is, however, extendable to any uniform quantizer by shifts and scaling.



(a) a^* (b) b^* (c) Local MSE

Fig. 2. Optimal linearization of a normalized two-level quantizer as function of the local interval center (x_0) and length (δ).

The optimal local linear approximation for this uniform quantizer is obtained by calculating (14)-(15) for the formula in (26). Again, the solution depends on the interval layout. In the simplest case, the considered interval is completely contained within a single decision region, i.e., $\eta(x_0,\delta) \subset [d_i, d_{i+1}]$ for some i. Here $q(x) = r_i$ for any $x \in \eta(x_0,\delta)$. Clearly, the corresponding discussion for the two-level quantizer (see section III-B) also holds here, meaning that $a^* = 0$ and $b^* = r_i$ with a zero local-MSE.

Another scenario that coincides with the two-level quantizer is when $x_0 - \delta \in [d_{i-1}, d_i]$ and $x_0 + \delta \in [d_i, d_{i+1}]$, i.e., the interval is spread over only two adjacent decision regions. Indeed, the optimal parameters here are obtained by appropriately shifting the results in (21)-(22). However, note that the multi-level quantizer has two levels only for $\delta < min\{d_{i+1} - x_0, x_0 - d_{i-1}\} < 1$.

Now we proceed to the main case, where the approximation interval spans over more than two decision regions, i.e., $x_0 - \delta \in [d_i, d_{i+1}]$ and $x_0 + \delta \in [d_j, d_{j+1}]$ for $j - i > 1$. First, we express the uniform quantization function as a sum of shifted two-level quantizers:

$$q_u(x) = \sum_{\tau=-\infty}^{\infty} q_2(x-\tau), \tag{27}$$

where $q_2(\cdot)$ was defined in (18). Then, using (27) we can develop (16)-(17) to the following forms:

$$L_a^u = \sum_{\tau=-\infty}^{\infty} L_a^\tau \tag{28}$$

$$L_b^u = \sum_{\tau=-\infty}^{\infty} L_b^\tau, \tag{29}$$

where L_a^τ and L_b^τ are the corresponding values for the two-level quantizer $q_2(x-\tau)$. These allow us to write the optimal linearization parameters of the uniform quantizer as the summation of the optimal parameters of the shifted two-level quantizers, namely

$$a_u^* = \sum_{\tau=-\infty}^{\infty} a_\tau^* \tag{30}$$

$$b_u^* = \sum_{\tau=-\infty}^{\infty} b_\tau^*, \tag{31}$$

where a_τ^* and b_τ^* are the optimal linearization parameters for $q_2(x-\tau)$ and are obtainable by shifting the expressions in (21)-(22). This analytic relation between the linearization of the uniform and the two-level quantizers is clearly exhibited in the numerical results (see Fig. 3) in the form of a periodic structure.

The numerical calculations (Fig. 3) also show convergence to the global approximation parameters

$$\lim_{\delta\to\infty} a_u^* = 1 \tag{32}$$

$$\lim_{\delta\to\infty} b_u^* = 0, \tag{33}$$

which imply $\lim_{\delta\to\infty} C_{lin}(x) = x$. In order to explain the results in Fig. 3, we return to the interpretation of a multi-level quantizer as a sum of shifted two-level quantizers (as expressed in Eq. (27)). First, examining the case of approximation around a decision level, shows that at each point of $\delta = k$ (for integer k values), two additional representation levels are included in the approximation (one on each side of the interval) and affect the optimal approximation. Comparing Fig. 3 to Fig. 2 reveals that the effect of each of these added representation levels is like approximating a two-level quantizer around a point that differs from its threshold level. Evaluating the approximation around a point that is not a decision level (see Fig. 3 while considering non-integer x_0 values) extends the previous behavior by combining two unsynchronized periodic patterns, each of them stems from a recurrent addition of representation levels from a different side.

The MSE plot (Fig. 3c) shows that, for nontrivial intervals that contain at least one non-differentiable point, the minimal MSE is obtained for approximation over a small interval that includes only the nearest decision level. This somewhat resembles the underlying principle of the dithering procedure [33], where the points within a quantization-cell are differentiated by an added noise that statistically maps them to neighboring cells according to their relative proximity. Moreover, maximal MSE of 0.106 is obtained for $\delta = 0.67$ and $x_0 = \frac{1}{2} + i$ (for $i = 0, \pm1, \pm2, ...$), where only the two adjacent non-differentiable points affect the linearization. This can also be shown analytically by setting the decompositions in (27) and (30)-(31) into (12), resulting in

$$LMSE_u^{SQ}\left(a^*, b^*; \eta(x_0,\delta)\right) = \tag{34}$$

$$\sum_{\tau=-\infty}^{\infty} LMSE_{\tau}^{SQ}\left(a_{\tau}^{*}, b_{\tau}^{*}; \eta\left(x_0, \delta\right)\right) + \frac{1}{2\delta} \sum_{\substack{\tau,\nu=-\infty \\ \tau\neq\nu}}^{\infty} \int_{x_0-\delta}^{x_0+\delta} \left(q_2(x-\tau) - a_{\tau}^{*}x - b_{\tau}^{*}\right) \times \times \left(q_2(x-\nu) - a_{\nu}^{*}x - b_{\nu}^{*}\right) dx,$$

where $LMSE_{\tau}^{SQ}\left(a_{\tau}^{*}, b_{\tau}^{*}; \eta\left(x_0, \delta\right)\right)$ is the optimal LMSE for $q_2(x-\tau)$ as available by shifting the expression in (23).

D. Transform Coding

We now turn to generalize the discussion to compression of multidimensional signals by considering the widely used concept of transform coding, where scalar quantization is applied in the transform domain. We examine coding of an N-length signal vector using a unitary transform, that can be formulated as the vector-valued function

$$C\left(\mathbf{x}\right) = \mathbf{U}Q\left(\mathbf{U}^T\mathbf{x}\right), \tag{35}$$

where $\mathbf{x}$ is the $N\times 1$ signal to compress, $\mathbf{U}$ is an $N\times N$ unitary matrix, and $Q\left(\cdot\right)$ is a vector-valued quantization function that scalarly quantizes the input components, i.e.,

$$Q\left(\mathbf{x}\right) = \begin{bmatrix} q\left(x_1\right) \\ \vdots \\ q\left(x_N\right) \end{bmatrix} \tag{36}$$

where $q\left(\cdot\right)$ is a single-variable scalar quantization function as studied above, and x_i is the i^{th} component of the vector $\mathbf{x}$. Moreover, as the last definition exhibits, the discussion is simplified by assuming identical quantization rules to all vector components.

As scalar quantization is a building block of the transform coding procedure (35), it imposes its non-differentiable nature on $C(\mathbf{x})$. Let us consider the linear approximation of $C(\mathbf{x})$ around the point $\mathbf{x}_0 \in \mathbb{R}^N$ in a limited neighborhood of a high-dimensional cube defined by δ as

$$\eta(\mathbf{x_0}, \delta) = \left\{\mathbf{x} \mid \|\mathbf{x}-\mathbf{x_0}\|_{\infty} \leq \delta\right\}. \tag{37}$$

The approximation takes the general multidimensional linear form of

$$\tilde{C}\left(\mathbf{x}\right) = \mathbf{A}\mathbf{x} + \mathbf{b} \tag{38}$$

where $\mathbf{A} \in \mathbb{R}^{N\times N}$ and $\mathbf{b} \in \mathbb{R}^N$ are the linearization parameters. The local MSE of approximating the transform-coding procedure around $\mathbf{x}_0$ is defined as

$$LMSE^{TC}\left(\mathbf{A}, \mathbf{b}; \eta(\mathbf{x_0}, \delta)\right) \triangleq \frac{1}{|\eta(\mathbf{x_0}, \delta)|} \int_{\eta(\mathbf{x_0}, \delta)} \left\| C(\mathbf{x}) - \tilde{C}\left(\mathbf{x}\right)\right\|_2^2 d\mathbf{x} \tag{39}$$

By substituting (38) in (39) and using the energy-preservation property of unitary transforms, we get the equivalent error expression in the transform-domain

$$\frac{1}{|\hat{\eta}(\mathbf{x}_0, \delta)|} \int_{\hat{\eta}(\mathbf{x}_0, \delta)} \left\| Q(\hat{\mathbf{x}}) - \hat{\mathbf{A}}\hat{\mathbf{x}} - \hat{\mathbf{b}}\right\|_2^2 d\hat{\mathbf{x}} \tag{40}$$

where

$$\begin{aligned} \hat{\mathbf{x}} &= \mathbf{U}^T\mathbf{x} \\ \hat{\mathbf{x}}_0 &= \mathbf{U}^T\mathbf{x}_0 \\ \hat{\mathbf{A}} &= \mathbf{U}^T\mathbf{A}\mathbf{U} \\ \hat{\mathbf{b}} &= \mathbf{U}^T\mathbf{b} \\ |\hat{\eta}(\mathbf{x}_0, \delta)| &= |\eta(\mathbf{x}_0, \delta)|. \end{aligned} \tag{41}$$

and the rotated approximation area (or volume), $\hat{\eta}$, is defined around $\hat{\mathbf{x}}_0$ and may not have sides that are aligned with the axes.

Let us generally define the local linearization error for compression of a signal vector, $\mathbf{x}$, by identical scalar quantization of its components, x_i:

$$\begin{aligned} LMSE^{SQV}\left(\mathbf{A}, \mathbf{b}; \bar{\eta}\right) &\triangleq \\ &= \frac{1}{|\bar{\eta}|} \int_{\bar{\eta}} \|Q(\mathbf{x}) - \mathbf{A}\mathbf{x} - \mathbf{b}\|_2^2 d\mathbf{x} \\ &= \frac{1}{|\bar{\eta}|} \sum_{i=1}^{N} \int_{\bar{\eta}} \left(q(x_i) - \mathbf{a}_i^T\mathbf{x} - b_i\right)^2 d\mathbf{x} \end{aligned} \tag{42}$$

where $\bar{\eta}$ is an arbitrary shaped approximation area, and the last equality relies on the separability of $Q\left(\cdot\right)$ and the L_2-norm definition.

Equations (40) and (42) clearly show that the MSE of approximating transform-coding (40) reduces to the linearization-error of scalar quantization of the transform coefficients, namely,

$$\begin{aligned} LMSE^{TC}\left(\mathbf{A}, \mathbf{b}; \eta(\mathbf{x}_0, \delta)\right) &= LMSE^{SQV}\left(\hat{\mathbf{A}}, \hat{\mathbf{b}}; \hat{\eta}(\mathbf{x}_0, \delta)\right) \\ &= \frac{1}{|\hat{\eta}(\mathbf{x}_0, \delta)|} \sum_{i=1}^{N} \int_{\hat{\eta}(\mathbf{x}_0, \delta)} \left(q(\hat{x}_i) - \hat{\mathbf{a}}_i^T\hat{\mathbf{x}} - \hat{b}_i\right)^2 d\hat{\mathbf{x}} \end{aligned} \tag{43}$$

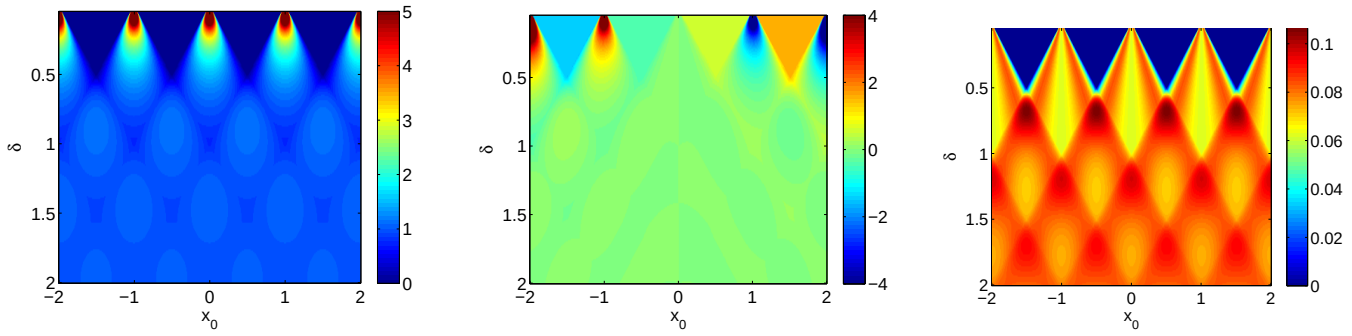
where $\hat{\mathbf{a}}_i^T$ is the i^{th} row of $\hat{\mathbf{A}}$, and $\hat{b}_i$ is the i^{th} element of the vector $\hat{\mathbf{b}}$.

While the separability of $Q\left(\cdot\right)$ was utilized to have integrals in (43) that consider quantization of single transform-coefficients, the integration is still over a multidimensional area that is not necessarily separable (i.e., not aligned with the axes). We can remedy this by starting from an appropriately rotated area in the signal-domain, $\eta_U(\mathbf{x}_0, \delta)$, such that its transform-domain counterpart is aligned with the axes (see Fig. 4):

$$\hat{\eta}_U(\mathbf{x}_0, \delta) = \left\{\hat{\mathbf{x}} \mid \|\hat{\mathbf{x}} - \hat{\mathbf{x}}_0\|_{\infty} \leq \delta\right\}. \tag{44}$$

Note that $\eta_U(\mathbf{x}_0, \delta)$ is not necessarily the optimally shaped approximation area as it is used here for the analytic simplicity of having full separability in the transform domain. We continue our transform-domain analysis by adopting this separable integration-area.

Recall that we look for the optimal linear approximation of the signal-domain function $C\left(\mathbf{x}\right)$. This is obtainable by finding the optimal transform-domain parameters $\hat{\mathbf{A}}^*$ and $\hat{\mathbf{b}}^*$ and then transforming them back to the signal domain. Following this



(a) a_u^* (b) b_u^* (c) Local MSE

Fig. 3. Optimal linearization of a normalized uniform quantizer as function of the local interval center (x_0) and length (δ).

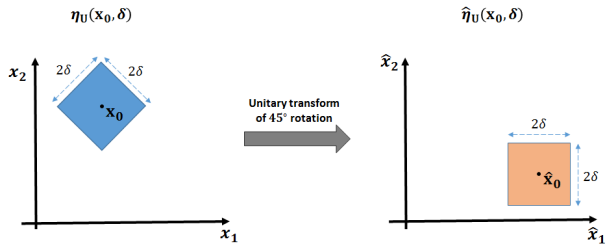


Fig. 4. Transformation of the approximation area. Exemplified in $\mathbb{R}^2$ for the unitary transform of 45°-rotation and a signal-domain area that is rotated in accordance to the transform.

strategy we first pose the componentwise optimality demands in the transform domain:

$$\frac{\partial}{\partial \hat{a}_{ij}} LMSE^{SQV}\left(\hat{\mathbf{A}},\hat{\mathbf{b}};\hat{\eta}_U(\mathbf{x}_0,\delta)\right)=0 \text{ for } i,j=1,...,N$$
$$\frac{\partial}{\partial \hat{b}_{i}} LMSE^{SQV}\left(\hat{\mathbf{A}},\hat{\mathbf{b}};\hat{\eta}_U(\mathbf{x}_0,\delta)\right)=0 \text{ for } i=1,...,N \quad (45)$$

Some calculations show that the solution satisfying the optimality conditions consists of a diagonal matrix $\hat{\mathbf{A}}^*$ (i.e., $\hat{a}_{ij}^*=0$ for $i\neq j$) such that the parameter pair $\left(\hat{a}_{ii}^*,\hat{b}_i^*\right)$ is the one obtained for optimal approximation of a single-variable quantizer over the interval $\left[\hat{\mathbf{x}}_0^{(i)}-\delta,\hat{\mathbf{x}}_0^{(i)}+\delta\right]$ as generally given in (14)-(15). Then, the signal-domain parameters are given as

$$\mathbf{A}^*=\mathbf{U}\hat{\mathbf{A}}^*\mathbf{U}^T=\sum_{i=1}^{N}\hat{a}_{ii}^*\mathbf{u}_i\mathbf{u}_i^T \quad (46)$$

$$\mathbf{b}^*=\mathbf{U}\hat{\mathbf{b}}^*. \quad (47)$$

where the last equality in (46) is due to the diagonality of $\hat{\mathbf{A}}^*$ and $\mathbf{u}_i$ denotes the i^{th} column of $\mathbf{U}$. The corresponding optimal error is equivalent in the signal and transform domains, hence can be expressed in a simplified form as

$$LMSE^{TC}\left(\mathbf{A}^*,\mathbf{b}^*;\eta_U(\mathbf{x}_0,\delta)\right) \quad (48)$$
$$=LMSE^{SQV}\left(\hat{\mathbf{A}}^*,\hat{\mathbf{b}}^*;\hat{\eta}_U(\mathbf{x}_0,\delta)\right)$$
$$=\frac{1}{2\delta}\sum_{i=1}^{N}\int_{\hat{\mathbf{x}}_0^{(i)}-\delta}^{\hat{\mathbf{x}}_0^{(i)}+\delta}\left(q(\hat{x}_i)-\hat{a}_{ii}^*\hat{x}_i-\hat{b}_i^*\right)^2 d\hat{\mathbf{x}}$$
$$=\sum_{i=1}^{N}LMSE^{SQ}\left(\hat{a}_{ii}^*,\hat{b}_i^*;\eta(\hat{\mathbf{x}}_0^{(i)},\delta)\right)$$

The last expression exhibits the approximation error of transform-coding as the sum of the errors of the separate linearization of the scalar quantization of the transform-domain coefficients. Although the assumed scenario includes equal quantization procedure for all the coefficients, the contributed errors by the various elements are different as each has its own scalar approximation-point $\hat{\mathbf{x}}_0^{(i)}$ located differently with respect to the quantization lattice.

Let us exemplify the latter analysis on a transform coder of two-component signals (i.e., $\mathbf{x}\in\mathbb{R}^2$), that scalary applies the normalized two-level quantizer that was studied above (see Eq. (18)) on the two components in the domain of the 45°-rotation matrix that takes the 2x2 form of $\mathbf{U}_{\pi/4}=\frac{1}{\sqrt{2}}\begin{bmatrix}1&-1\\1&1\end{bmatrix}$. The approximation is around $\mathbf{x}_0=\mathbf{U}_{\pi/4}\hat{\mathbf{x}}_\mathbf{0}$ in a 45°-rotated square neighborhood defined by the $\eta_U(\mathbf{x}_0,\delta)$ (see Fig. 4). We further define the first component of $\hat{\mathbf{x}}_\mathbf{0}$ to vary and fix the second on the value of 15, i.e., $\hat{\mathbf{x}}_\mathbf{0}=\begin{bmatrix}\hat{\mathbf{x}}_\mathbf{0}^{(1)}\\15\end{bmatrix}$. The overall linearization error as function of δ and $\hat{\mathbf{x}}_\mathbf{0}^{(1)}$ (Fig. 5) shows that it combines the errors of the scalar linearization of the transform coefficients (Figs. 6e-6f). The corresponding parameters in the signal domain (where the matrix $\mathbf{A}$ is not necessarily diagonal) are given in Fig. 7. Again, the results generalize the previous observations by demonstrating that minimal MSE is obtained for approximation over the minimal area that includes the nearest non-differentiable point of the compression function (see Fig. 5).

We now further develop the discussed transform-coder to the common procedure where the transform-coefficients are uniformly quantized according to different step sizes, $\{\Delta_i\}_{i=1}^N$, getting coarser for higher frequencies, i.e., $\Delta_i\leq\Delta_j$ for $i<j$. Let us consider N-length signal vectors and analyze the linear approximation over the $\eta_U(\mathbf{x}_0,\delta)$ neighborhood, where $\hat{\mathbf{A}}^*$ is diagonal. Accordingly, in this case, quantization of each transform-coefficient is linearized separately over a

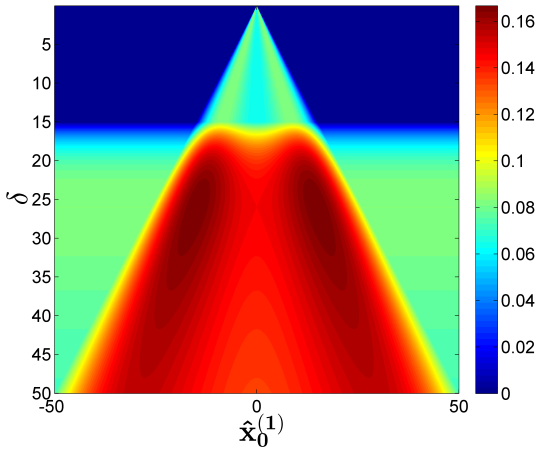


Fig. 5. Overall MSE of optimal linearization of the exemplary transform coding procedure (for signals in $\mathbb{R}^2$ and equal transform-domain quantizers).

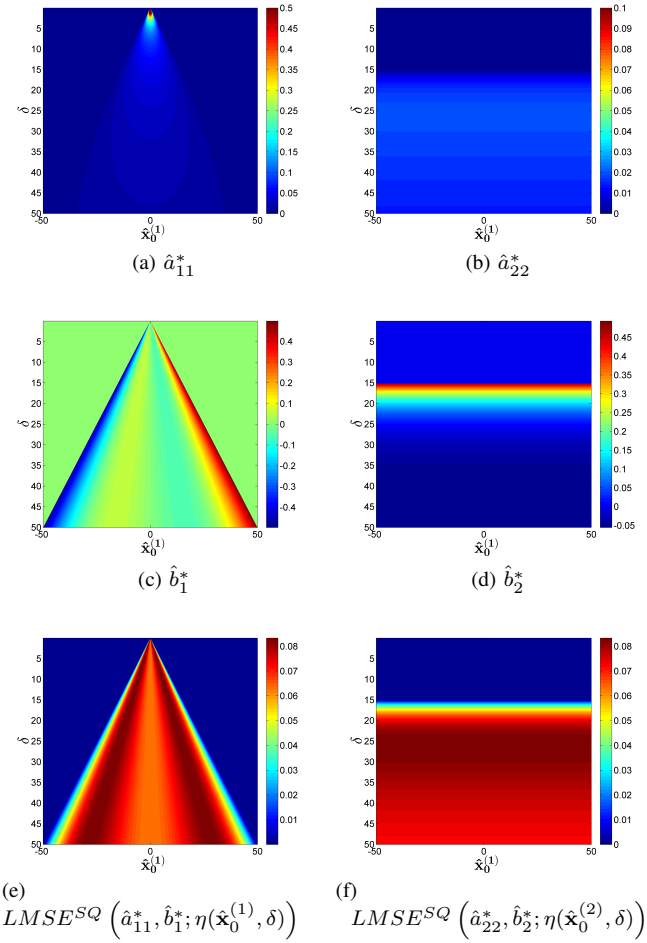


Fig. 6. Transform domain parameters of the optimal linear approximation of the exemplary transform coding procedure (for signals in $\mathbb{R}^2$ and equal transform-domain quantizers). (a)-(b) describe the diagonal elements of the 2x2 matrix $\hat{\mathbf{A}}$, and (c)-(d) show the values of $\hat{\mathbf{b}}$'s components. (e)-(f) show the corresponding approximation errors of the two-transform domain elements.

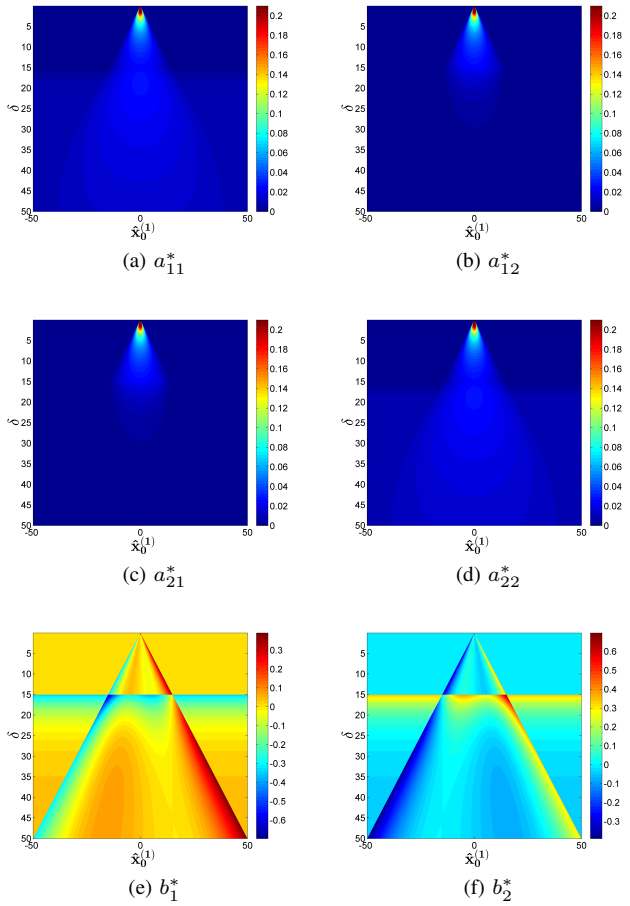
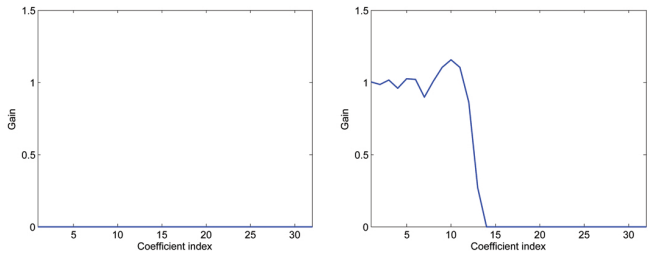


Fig. 7. Signal domain parameters of the optimal linear approximation of the exemplary transform coding procedure (for signals in $\mathbb{R}^2$ and equal transform-domain quantizers). (a)-(d) describe the components of the 2x2 matrix $\mathbf{A}$, and (e)-(f) show the values of $\mathbf{b}$'s components.

one-dimensional interval of size 2δ. However, due to different quantization-steps and approximation-points, the $\{\hat{a}_{ii}^*\}_{i=1}^N$ values vary. We simplify the discussion by approximating around a vector $\mathbf{x}_0$ with components residing near the middle of the scalar decision-regions of the transform-domain quantizers. Then, relying on the above analysis of the one-dimensional uniform quantizer, we note the following behavior of the sequence $\{\hat{a}_{ii}^*\}_{i=1}^N$ for $2\delta \approx \Delta_K$ $(K > 1)$. First, for some integer L $(1 \leq L < K)$, the relation $\Delta_i \ll \Delta_K \approx 2\delta$ holds for any $i \leq L$, and therefore, the optimal parameters are $\hat{a}_{ii}^* \approx 1$ and $\hat{b}_i^* \approx 0$. Second, for $L < i \leq K$, the 2δ value is still greater than Δ_i, however relatively closer, hence, the corresponding $\hat{a}_{ii}^*$ values fluctuate. Finally, the $i > K$ coefficients have quantization steps that are greater than 2δ, and accordingly, $\hat{a}_{ii}^* \approx 0$. The latter qualitative analysis lets us to interpret the local-approximation of the transform-coder, $\hat{\mathbf{A}}^*$, as a low-pass filter that depends on δ and the approximation point. Furthermore, the numerical results (Fig. 8) demonstrate the above by showing preservation of low frequencies, an unstable transition phase, and attenuation of high-frequency components. Note that for too low or too high values of δ the filter has a all-stop (Fig. 8a) or all-pass (Fig. 8d) behavior, respectively.

We conclude by considering the signal-domain filter $\mathbf{A}^*$



(a) $\delta = 0.5$ (b) $\delta = 5$

(c) $\delta = 50$ (d) $\delta = 500$

Fig. 8. Interpretation of a diagonal $\hat{\mathbf{A}}^*$ as a transform-domain filter that depends on δ. Here $N = 32$ and the i^{th} quantization step is $\Delta_i = 2^{i/4}$.

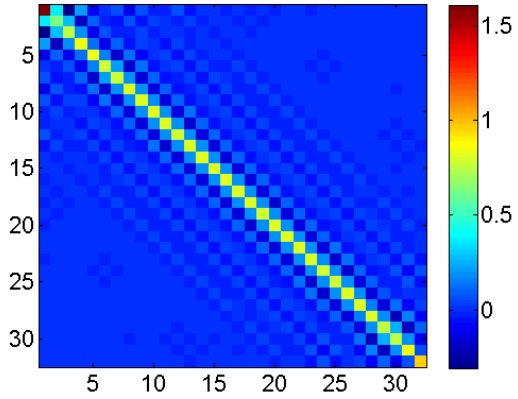


Fig. 9. Interpretation of $\mathbf{A}^*$ as a signal-domain filter that depends on δ. Presented here for the case of $\delta = 50$ from Fig. 8c, incorporated in a DCT-based coding.

related to $\hat{\mathbf{A}}^*$ by the inverse-transformation in (46). When the compression utilizes the Discrete Fourier Transform (DFT) and the approximation is over $\eta_U(\mathbf{x}_0, \delta)$, then the diagonal matrix $\hat{\mathbf{A}}^*$ yields a circulant $\mathbf{A}^*$. While the latter involves complex-valued calculations, coding using Discrete Cosine Transform (DCT) keeps the procedure over the reals. The signal-domain filter, $\mathbf{A}^*$, of the DCT-based coding is exemplified in Fig. 9 showing an approximately Toeplitz structure.

E. Postprocessing of Simplistic Compression Procedures

We turn now to demonstrate the proposed postprocessing method by applying it on the simplistic compression procedures of scalar quantization and one-dimensional transform coding. The experimental settings (e.g., the utilized denoiser, the stopping criterion, and the definition of the oracle denoiser) are thoroughly explained in the beginning of section IV.

TABLE I
PSNR COMPARISON FOR SCALAR QUANTIZATION

Image	Bit-Rate	No Postprocessing	Oracle Denoiser	Proposed Method
Lena 256x256	2	22.86	24.54	**24.84**
	3	28.90	30.80	**31.02**
	4	34.64	37.19	**37.41**
	5	40.72	42.82	**43.02**
Barbara 256x256	2	23.47	25.66	**25.93**
	3	28.48	30.94	**31.11**
	4	34.72	37.29	**37.38**
	5	40.74	42.55	**42.64**

(a) Scalar Quantization (34.64dB) (b) Postprocessing Result (37.41dB)

Fig. 10. Reconstruction of Lena (256x256) from scalar quantization at 4bpp.

1) Scalar Quantization: We begin with the elementary compression procedure of applying uniform scalar quantization (as formulated in (26)) on the signal samples. Motivated by the analysis in section III-C, we define here the approximation interval to be no longer than the quantization step Δ, as considering only the nearest non-differentiable point yields a useful linearization. Accordingly, the derivative (which is scalar here) is approximated using (8) and $S_\delta = \{0.1\Delta, 0.2\Delta, 0.3\Delta, 0.4\Delta, 0.5\Delta\}$. While λ's value was fixed to 0.01, the β and μ parameters were set as functions of the bit-rate: $\beta = 500 \cdot 2^{-2bpp}$ and $\mu = 5 \cdot 10^{-4} \cdot 2^{0.6bpp}$. Our technique achieved impressive PSNR improvements (Table I) over the entire bit-rate range, and consistently passed the oracle denoiser[5]. Visually, the false-contouring artifacts were significantly reduced (Fig. 10).

2) One-Dimensional Transform Coding: Now we extend the examined compression method by performing scalar quantization in the transform domain. Specifically, we split the image into nonoverlapping one-dimensional vertical vectors of 2 pixels. Then we compress them separately by transform coding them using $\mathbf{U}_{\pi/4} = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & -1 \\ 1 & 1 \end{bmatrix}$, followed by applying uniform quantization with identical step size to all the coefficients. We evaluated here the algorithm performance for the two types of approximation area that were discussed above: a square area aligned with the axes (i.e., $\eta(\mathbf{x}_0, \delta)$), and a $45°$-rotated squared area (i.e., $\hat{\eta}_U(\mathbf{x}_0, \delta)$) that allows to calculate the linearization in the transform domain and then

[5]The oracle denoiser is defined at the beginning of section IV.

transforming the parameters back to the signal domain using (46)-(47). Both options used square areas of the same size, by setting $S_\delta = \{0.1\Delta, 0.2\Delta, 0.3\Delta, 0.4\Delta, 0.5\Delta\}$. The two area types achieved better results than the oracle denoiser. In that sense, our approach is somewhat robust to the area shape (however, not necessarily to its size). In addition, employing area that is aligned with the axes in the signal domain consistently obtained higher PSNR than the rotated area (Table II). The latter observation will motivate us to use aligned-cubic approximation areas also for more complex compression techniques that will follow next.

IV. Experimental Results

In this section we demonstrate the performance of the proposed postprocessing method by presenting results obtained in conjunction with the leading image compression standards: JPEG, JPEG2000 and the recent HEVC.

In all experiments we use the BM3D method [25] as the denoiser. The code was implemented in Matlab. While the settings differ for the various compression methods, a similar stopping criterion is applied. In (3) we introduced the scaled dual-variable of the i^{th} iteration, $\mathbf{u}_i \in \mathbb{R}^n$. We here denote $\Delta\mathbf{u}_i = \frac{1}{N}\|\mathbf{u}_i - \mathbf{u}_{i-1}\|_1$ and set the algorithm termination conditions to be at one of the following $\Delta\mathbf{u}_i < 0.05$, $\Delta\mathbf{u}_i > \Delta\mathbf{u}_{i-1}$ or some maximal number of iterations attained.

Since the proposed technique uses a well established denoiser as a subroutine, the reference postprocessing we compare with is a single application of this denoiser. This approach is further strengthened by endorsing the denoiser with an oracle capability by searching for the best parameter in terms of maximal PSNR result.

The computational requirements of our suggested technique can be relaxed for many compression methods that operate independently on adjacent blocks. Specifically, the Jacobian becomes a block-diagonal matrix and, therefore, its columns can be arranged in independent subsets for concurrent computation. This reduces the number of compression-decompression applications to the order of the block size.

A. JPEG

This well known standard [26] is a relatively straightforward implementation of a two-dimensional transform coding on 8x8 blocks of the image. Specifically, the quantization is performed in the DCT domain where each coefficient has its own quantization step. As the JPEG extends the oversimplified procedure in subsection III-E2, our postprocessing method is expected to provide good results here. Indeed, the experiments show the impressive gains of the suggested method that compete with the leading technique from [34] (Fig. 11). Moreover, while the competitive methods [1], [3], [15], [34] operate on the low bit-rate range, our method handles the entire bit-rate range and usually exceeds the oracle denoiser (see Fig. 11 and Table III).

B. JPEG2000

This efficient standard [27] applies transform coding in the wavelet domain for relatively large signal blocks (also known as tiles) of at least 128x128 size. Not only the tile size affects the compression run-time, it also impairs the suggested parallelism optimization, as it is beneficial for small block sizes. Nevertheless, it is still recommended to reduce the computational cost by concurrent computation of the Jacobian columns in relatively large subgroups that inevitably contain dependent elements. Our experiments included postprocessing of images compressed using JPEG2000 compression without any tiling. However, the Jacobian was estimated by assuming independent

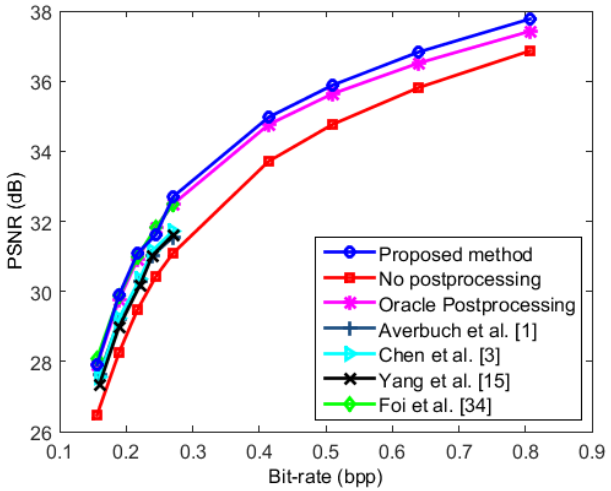


Fig. 11. Reconstruction of Lena (512x512) from JPEG compression.

TABLE III
PSNR Comparison for JPEG

Image	Bit-Rate	No Postprocessing	Oracle Denoiser	Proposed Method
Lena 512x512	0.189	28.24	29.76	**29.90**
	0.271	31.09	32.50	**32.69**
	0.511	34.76	35.64	**35.87**
	0.807	36.86	37.42	**37.79**
Barbara 512x512	0.764	30.89	32.49	**32.59**
	0.938	32.54	33.94	**34.33**
	1.149	34.22	35.34	**35.94**
	1.552	36.88	37.65	**38.31**

(a) JPEG (31.09dB) (b) Postprocessing Result (32.69dB)

Fig. 12. Reconstruction of Lena (512x512) from JPEG compression at 0.271bpp.

TABLE II
PSNR COMPARISON FOR TRANSFORM CODING OF TWO-COMPONENT VECTORS

Image	Quantizer Step Δ	No Postprocessing	Oracle Denoiser	Proposed Method Aligned Approx. Area	Proposed Method Rotated Approx. Area
Lena 256x256	20	31.22	35.83	**37.00**	36.78
	30	27.45	32.21	**33.37**	33.00
	40	24.88	29.92	**30.92**	30.29
Barbara 256x256	15	34.50	38.00	**38.25**	38.12
	20	31.79	35.65	**36.42**	36.31
	30	28.05	32.33	**33.33**	33.28

TABLE IV
PSNR COMPARISON FOR JPEG2000

Image	Bit-Rate	No Postprocessing	Oracle Denoiser	Proposed Method
Lena 256x256	0.273	30.15	30.62	**30.64**
	0.323	31.19	31.75	**31.81**
	0.405	32.48	33.04	**33.08**
	0.510	34.05	34.55	**34.63**
Barbara 256x256	0.329	29.28	29.71	**29.77**
	0.404	30.16	30.67	**30.80**
	0.510	31.40	31.92	**31.98**
	0.778	34.12	34.85	**34.90**

(a) JPEG2000 (30.16dB) (b) Postprocessing Result (30.80dB)

Fig. 13. Reconstruction of Barbara (256x256) from JPEG2000 compression at 0.404bpp.

TABLE V
PSNR COMPARISON FOR HEVC

Image	Bit-Rate	No Postprocessing	Oracle Denoiser	Proposed Method
Lena 128x128	0.334	29.45	29.57	**29.65**
	0.411	30.78	30.92	**30.97**
	0.513	32.32	32.49	**32.56**
	0.650	33.94	34.14	**34.21**
Barbara 128x128	0.401	29.15	29.29	**29.34**
	0.602	31.89	32.10	**32.15**
	0.745	33.47	33.70	**33.78**
	0.911	35.07	35.28	**35.36**

8x8 blocks, where this reduced accuracy yielded considerable relief in the computational burden. The reconstruction PSNR of our method reached up to 0.7dB improvement of the JPEG2000 output, and consistently passed the oracle denoiser result (Table IV). Although the achieved gain over the oracle is modest, these results are encouraging since the oracle denoiser needs the precompressed image and therefore not suitable for the common compression applications. The restoration results visually demonstrated the artifact reduction using our method, specifically, handling of the ringing artifact (Fig. 13).

Whereas other JPEG2000 postprocessing techniques [17], [18] are intended for the (very) low bit-rate range, our method performs well at any bit-rate. Although the experimental settings in [17], [18] are somewhat different, our method evidently outperforms the technique in [17] that achieves maximal gain of 0.2dB (at very low bit-rates). The results in [18], which are also for low bit-rates, demonstrate similar gains to our method, however, we stress again that our approach is intended for a significantly wider bit-rate range.

C. HEVC

This state-of-the-art coding standard offers a profile of still-image compression [28], [29]. The HEVC applies spatial hybrid-coding on the image by combining a rich prediction capability with transform coding of the prediction residuals. In addition, the image is divided into large blocks (also known as coding units) that are further recursively partitioned into rectangular blocks in various sizes. Therefore, our Jacobian estimation is set to work on independent blocks of 64x64 size, and thus the corresponding run-time was higher than for the previous compression methods. The results here are for HEVC-compression using the software library in [35]. Again, our postprocessing results consistently exceeded the oracle denoiser, as shown in the PSNR comparison in Table V. Since HEVC is a recent standard, other artifact-reduction techniques for its still-image profile have not proposed yet.

To summarize this section, the extensive experiments established the proposed compression-artifact reduction technique as a generic method that achieves cutting-edge results for any relevant image compression and over the entire bit-rate range.

V. CONCLUSION

In this paper we proposed a novel postprocessing method for reducing artifacts in compressed images. The task was

TABLE VI
EXPERIMENTAL SETTINGS FOR THE EXAMINED COMPRESSION METHODS

Compression Method	$S_\delta = \left\{0.1\tilde{\Delta}k\right\}_{k=1}^{5}$	λ	β	μ
JPEG	$\tilde{\Delta} = 135/r$	0.15	$2 \cdot r^{-1}$	$0.01 \cdot 2^r$
JPEG2000	$\tilde{\Delta} = 135/r$	0.15	$5 \cdot r^{-2.3}$	$0.3 \cdot 2^r$
HEVC	$\tilde{\Delta} = 50/r$	0.15	$5 \cdot r^{-1}$	$0.3 \cdot 2^r$

here r is the bit-rate in bpp units.

formulated as a regularized inverse problem, that was subsequently transformed into an iterative form by relying on the ADMM and the Plug-and-Play frameworks. The resulting generic algorithm separately treats the inversion and the regularization, where the latter is implemented by sequentially applying an existing state-of-the-art Gaussian denoiser. For practicality we simplified the inversion step by representing the nonlinear compression-decompression procedure using a linear approximation. Furthermore, we provided a comprehensive mathematical analysis for linear approximation of simplified quantization and transform-coding operations. We demonstrated our approach for image compression and presented experimental-results showing impressive gains, that improve upon state-of-the-art postprocessing results for leading image compression standards.

REFERENCES

[1] A. Averbuch, A. Schclar, and D. L. Donoho, "Deblocking of block-transform compressed images using weighted sums of symmetrically aligned pixels," *IEEE Trans. Image Process.*, vol. 14, no. 2, pp. 200–212, 2005.

[2] Y.-C. Liaw, W. Lo, and J. Z. C. Lai, "Image restoration of compressed image using classified vector quantization," *Pattern Recognition*, vol. 35, no. 2, pp. 329–340, 2 2002.

[3] T. Chen, H. R. Wu, and B. Qiu, "Adaptive postfiltering of transform coefficients for the reduction of blocking artifacts," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 11, no. 5, pp. 594–602, 2001.

[4] K. Lee, D. S. Kim, and T. Kim, "A new postprocessing algorithm based on regression functions," in *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, vol. 4, 2002, pp. IV–3684–IV–3687.

[5] G. A. Triantaffilidis, D. Sampson, D. Tzovaras, and M. G. Strintzis, "Blockiness reduction in JPEG coded images," in *International Conference on Digital Signal Processing*, vol. 2, 2002, pp. 1325–1328 vol.2.

[6] Y.-C. Liaw, W. Lo, and J. Z. Lai, "Image restoration of compressed image using classified vector quantization," *Pattern Recognition*, vol. 35, no. 2, pp. 329–340, 2002.

[7] F. Alter, S. Durand, and J. Froment, "Adapted total variation for artifact free decompression of JPEG images," *Journal of Mathematical Imaging and Vision*, vol. 23, no. 2, pp. 199–211, 2005.

[8] K. Du, J. Lu, H. Sekiya, Y. Sun, and T. Yahagi, "Post-processing for restoring edges and removing artifacts of low bit rates wavelet-based image," in *International Symposium on Intelligent Signal Processing and Communications*, 2006, pp. 943–946.

[9] T. Kartalov, Z. A. Ivanovski, L. Panovski, and L. J. Karam, "An adaptive pocs algorithm for compression artifacts removal," in *International Symposium on Signal Processing and Its Applications*, 2007, pp. 1–4.

[10] T. Tillo and G. Olmo, "Data-dependent pre- and postprocessing multiple description coding of images," *IEEE Trans. Image Process.*, vol. 16, no. 5, pp. 1269–1280, 2007.

[11] P. Weiss, L. Blanc-Feraud, T. Andre, and M. Antonini, "Compression artifacts reduction using variational methods : Algorithms and experimental study," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2008, pp. 1173–1176.

[12] K. Du, H. Han, and G. Wang, "A new algorithm for removing compression artifacts of wavelet-based image," in *IEEE International Conference on Computer Science and Automation Engineering*, vol. 1, 2011.

[13] C. Jung, L. Jiao, H. Qi, and T. Sun, "Image deblocking via sparse representation," *Signal Processing: Image Communication*, vol. 27, no. 6, pp. 663–677, 2012.

[14] A. Zakhor, "Iterative procedures for reduction of blocking effects in transform image coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 2, no. 1, pp. 91–95, 1992.

[15] Y. Yang, N. P. Galatsanos, and A. Katsaggelos, "Regularized reconstruction to reduce blocking artifacts of block discrete cosine transform compressed images," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 3, no. 6, pp. 421–432, 1993.

[16] A. Nosratinia, "Enhancement of JPEG-compressed images by re-application of jpeg," *Journal of VLSI signal processing systems for signal, image and video technology*, vol. 27, no. 1-2, pp. 69–79, 2001.

[17] ——, "Postprocessing of JPEG-2000 images to remove compression artifacts," *IEEE Signal Processing Letters*, vol. 10, no. 10, pp. 296–299, 2003.

[18] Y. Kwon, K. Kim, J. Tompkin, J. Kim, and C. Theobalt, "Efficient learning of image super-resolution and compression artifact removal with semi-local gaussian processes," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 37, no. 9, pp. 1792–1805, Sept 2015.

[19] M.-Y. Shen and C. C. J. Kuo, "Review of postprocessing techniques for compression artifact removal," *Journal of Visual Communication and Image Representation*, vol. 9, no. 1, pp. 2–14, 3 1998.

[20] M. V. Afonso, J. M. Bioucas-Dias, and M. A. T. Figueiredo, "Fast image recovery using variable splitting and constrained optimization," *IEEE Trans. Image Process.*, vol. 19, no. 9, pp. 2345–2356, 2010.

[21] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Found. Trends Mach. Learn.*, vol. 3, no. 1, pp. 1–122, 2011.

[22] S. V. Venkatakrishnan, C. A. Bouman, and B. Wohlberg, "Plug-and-play priors for model based reconstruction," in *IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, 2013, pp. 945–948.

[23] M. Aharon, M. Elad, and A. Bruckstein, "K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation," *IEEE Trans. Signal Process.*, vol. 54, no. 11, pp. 4311–4322, 2006.

[24] M. Elad and M. Aharon, "Image denoising via sparse and redundant representations over learned dictionaries," *IEEE Trans. Image Process.*, vol. 15, no. 12, pp. 3736–3745, 2006.

[25] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, "Image denoising by sparse 3-D transform-domain collaborative filtering," *IEEE Trans. Image Process.*, vol. 16, no. 8, pp. 2080–2095, 2007.

[26] G. K. Wallace, "The JPEG still picture compression standard," *IEEE Trans. Consum. Electron.*, vol. 38, no. 1, pp. xviii–xxxiv, 1992.

[27] C. Christopoulos, A. Skodras, and T. Ebrahimi, "The JPEG2000 still image coding system: an overview," *IEEE Trans. Consum. Electron.*, vol. 46, no. 4, pp. 1103–1127, 2000.

[28] G. J. Sullivan, J. Ohm, W.-J. Han, and T. Wiegand, "Overview of the high efficiency video coding (HEVC) standard," *IEEE Trans. on Circuits Syst. Video Technol.*, vol. 22, no. 12, pp. 1649–1668, 2012.

[29] J. Lainema, F. Bossen, W.-J. Han, J. Min, and K. Ugur, "Intra coding of the HEVC standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 12, pp. 1792–1801, 2012.

[30] T. Nguyen and D. Marpe, "Performance analysis of HEVC-based intra coding for still image compression," in *Picture Coding Symposium*, 2012, pp. 233–236.

[31] J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural Networks*, vol. 61, pp. 85–117, 2015.

[32] A. Gersho and R. M. Gray, *Vector quantization and signal compression*. Kluwer Academic Publishers, 1992.

[33] L. Schuchman, "Dither signals and their effect on quantization noise," *IEEE Transactions on Communication Technology*, vol. 12, no. 4, pp. 162–165, 1964.

[34] A. Foi, V. Katkovnik, and K. Egiazarian, "Pointwise shape-adaptive DCT for high-quality denoising and deblocking of grayscale and color images," *IEEE Trans. Image Process.*, vol. 16, no. 5, pp. 1395–1411, 2007.

[35] F. Bellard, "BPG 0.9.6." [Online]. Available: http://bellard.org/bpg/