New Methods for Signal Compression and Their Relations to Restoration Problems

Yehuda Dar
New Methods for Signal Compression and Their Relations to Restoration Problems

Research Thesis

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy

Yehuda Dar

Submitted to the Senate
of the Technion — Israel Institute of Technology
Tishrei 5779 Haifa September 2018
The research thesis was done under the supervision of Prof. Alfred M. Bruckstein and Prof. Michael Elad in the Faculty of Computer Science.

**Acknowledgements**

I would like to express my deepest gratitude to my advisors, Prof. Alfred Bruckstein and Prof. Michael Elad, for their invaluable guidance and support throughout the exciting journey of my PhD studies. I feel truly fortunate for the opportunity to work with such extraordinary researchers, each in his unique way showed me how to pursue excellence in research while enjoying the process through endless curiosity and enthusiasm for science. I consider Freddy and Miki as my mentors to the academic life and beyond.

I would also like to thank the members of the Geometric Image Processing Lab for creating the ultimate atmosphere for graduate studies and research.

I am greatly thankful to my mother Yona, for all her love and support throughout the years. The values and perspectives I learned from her have been of utmost importance in my road to this achievement.

The generous financial help of the Technion is gratefully acknowledged.
Publications

This research thesis has led to the following publications.

Journal Publications:


Publications in Conference Proceedings:

- Y. Dar, M. Elad, and A. M. Bruckstein, "Image Restoration via Successive Compression", in Picture Coding Symposium, 2016.
# Contents

List of Figures .......................................................... 1

Abstract ............................................................................. 1

Abbreviations and Notations ................................................. 3

1 Introduction ...................................................................... 5
  1.1 Signal Compression, Restoration, and Their Interplay .............. 5
  1.2 Dissertation Contributions ............................................. 6
    1.2.1 An Iterative Optimization Methodology for Intricate
    Rate-Distortion Problems .................................................. 6
    1.2.2 System Optimization from the Compression Standpoint .......... 7
    1.2.3 Extension of Image and Video Compression Standards to Optimize
    Acquisition-Rendering Systems .................................... 8
    1.2.4 The Information-Theoretic Perspective: Relation to the Remote
    Source Coding Problem and New Analysis ............................. 9
    1.2.5 Restoration by Compression: New Methods and Analysis .......... 9
    1.2.6 Compression-Artifact Reduction via Denoising-based Postprocessing 10

2 An Iterative Optimization Approach for Intricate Compression Problems ......................................................... 11
  2.1 Lossy Compression via Operational Rate-Distortion Optimization . 11
  2.2 Standard Compression Designs: Using Squared-Error Distortion Metrics
  and Block-based Coding ....................................................... 13
  2.3 The Proposed Iterative-Optimization Framework for Intricate Compression Problems ................................................. 15

3 Optimized Pre-Compensating Compression .............................................. 19
  3.1 Introduction .................................................................... 19
  3.2 The Proposed Method .................................................... 22
    3.2.1 The Basic Rate-Distortion Optimization .......................... 22
    3.2.2 Practical Iterative Procedure ....................................... 23
  3.3 Adjusting HEVC Image Compression to Blurry Decompression ........ 26
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.3.1</td>
<td>Competing Methods</td>
<td>26</td>
</tr>
<tr>
<td>3.3.2</td>
<td>Our Method: Experiment Settings</td>
<td>30</td>
</tr>
<tr>
<td>3.4</td>
<td>Application to LCD Motion-Blur Reduction</td>
<td>32</td>
</tr>
<tr>
<td>3.4.1</td>
<td>The LCD Motion Blur and Its Modeling</td>
<td>32</td>
</tr>
<tr>
<td>3.4.2</td>
<td>The Proposed Method for Motion-Blur Reduction</td>
<td>33</td>
</tr>
<tr>
<td>3.4.3</td>
<td>Experimental Results</td>
<td>35</td>
</tr>
<tr>
<td>4</td>
<td>System-Aware Compression: Optimizing Systems from the Compression Standpoint</td>
<td>41</td>
</tr>
<tr>
<td>4.1</td>
<td>Introduction</td>
<td>41</td>
</tr>
<tr>
<td>4.2</td>
<td>The Proposed Method</td>
<td>43</td>
</tr>
<tr>
<td>4.3</td>
<td>Theoretic Analysis for the Gaussian Case</td>
<td>47</td>
</tr>
<tr>
<td>4.3.1</td>
<td>Problem Formulation and Solution</td>
<td>47</td>
</tr>
<tr>
<td>4.3.2</td>
<td>Problem Solution in Theory and Practice</td>
<td>49</td>
</tr>
<tr>
<td>4.4</td>
<td>Experimental Results</td>
<td>50</td>
</tr>
<tr>
<td>4.5</td>
<td>Proofs for the Rate-Distortion Theoretic Analysis</td>
<td>55</td>
</tr>
<tr>
<td>4.5.1</td>
<td>Minimal Expected Distortion</td>
<td>55</td>
</tr>
<tr>
<td>4.5.2</td>
<td>Equivalence of Optimization Problems (4.18) and (4.20)</td>
<td>55</td>
</tr>
<tr>
<td>4.5.3</td>
<td>Equivalence of Optimization Problems (4.20) and (4.21)</td>
<td>59</td>
</tr>
<tr>
<td>5</td>
<td>Compression for Multiple Reconstructions</td>
<td>63</td>
</tr>
<tr>
<td>5.1</td>
<td>Introduction</td>
<td>63</td>
</tr>
<tr>
<td>5.2</td>
<td>The Proposed Method</td>
<td>64</td>
</tr>
<tr>
<td>5.2.1</td>
<td>Problem Formulation</td>
<td>64</td>
</tr>
<tr>
<td>5.2.2</td>
<td>Practical Iterative Procedure</td>
<td>66</td>
</tr>
<tr>
<td>5.3</td>
<td>Experimental Results</td>
<td>67</td>
</tr>
<tr>
<td>6</td>
<td>Restoration by Compression</td>
<td>71</td>
</tr>
<tr>
<td>6.1</td>
<td>Introduction</td>
<td>71</td>
</tr>
<tr>
<td>6.2</td>
<td>Complexity-Regularized Restoration: Problem Settings</td>
<td>73</td>
</tr>
<tr>
<td>6.2.1</td>
<td>Regularized Restoration of Signals</td>
<td>73</td>
</tr>
<tr>
<td>6.2.2</td>
<td>Operational Rate-Distortion Optimization</td>
<td>74</td>
</tr>
<tr>
<td>6.2.3</td>
<td>Complexity-Regularized Restoration: Basic Optimization Formulation</td>
<td>75</td>
</tr>
<tr>
<td>6.3</td>
<td>Proposed Methods</td>
<td>76</td>
</tr>
<tr>
<td>6.3.1</td>
<td>Regularize Total Complexity of Non-Overlapping Blocks</td>
<td>76</td>
</tr>
<tr>
<td>6.3.2</td>
<td>Regularize Total Complexity of All Overlapping Blocks</td>
<td>78</td>
</tr>
<tr>
<td>6.4</td>
<td>Rate-Distortion Theoretic Analysis for the Gaussian Case</td>
<td>83</td>
</tr>
<tr>
<td>6.4.1</td>
<td>Prevalent Restoration Strategies</td>
<td>85</td>
</tr>
<tr>
<td>6.4.2</td>
<td>The Complexity-Regularized Restoration Problem and its Equivalent Forms</td>
<td>86</td>
</tr>
<tr>
<td>6.4.3</td>
<td>Demonstration of The Explicit Results</td>
<td>89</td>
</tr>
</tbody>
</table>
6.4.4 Conceptual Relation to The Proposed Approach 90
6.5 Experimental Results 94
  6.5.1 Image Deblurring 95
  6.5.2 Image Inpainting 99
6.6 Proofs for the Rate-Distortion Theoretic Analysis 102
  6.6.1 Equivalence of Problems 6.1 and 6.2 102
  6.6.2 Equivalence of Problems 6.2 and 6.3 108
  6.6.3 Solution of Problem 6.4 109
  6.6.4 Equivalent Form of Stage 6 of Algorithm 6.1 110
7 Compression Postprocessing via Sequential Denoising 111
  7.1 Introduction 111
  7.2 The Proposed Postprocessing Strategy 113
    7.2.1 Problem Formulation using ADMM 113
    7.2.2 Linear Approximation of the Compression-Decompression Procedure 115
  7.3 Linear Approximation – A Closer Look 116
    7.3.1 Local Linear-Approximation of a Quantizer 117
    7.3.2 The Case of Two-Level Quantization 118
    7.3.3 The Case of Multi-Level Uniform Quantization 121
    7.3.4 Transform Coding 123
  7.4 Experimental Results 128
    7.4.1 Simplistic Compression Procedures 133
    7.4.2 JPEG 134
    7.4.3 JPEG2000 137
    7.4.4 HEVC 137
  7.5 Extension to Postprocessing of Intra-Frame Video Coding 139
    7.5.1 The Proposed Postprocessing Method 142
    7.5.2 Problem Formulation using ADMM 143
    7.5.3 Experimental Results 147
8 Conclusion 151
  8.1 Thesis Summary 151
  8.2 Future Research Directions 153
A A Tree-based Coding Method for One-Dimensional Signals 155
Hebrew Abstract i
List of Figures

1.1  The general system structure addressed by the system-aware compression framework. .......................... 7

1.2  The general imaging system structure addressed by the proposed compression framework. .......................... 8

3.1  The considered flow of an image/video that is first compressed and finally perceived by a human observer. .......................... 20

3.2  Demonstration of the conceptual problem settings for compression that is oriented to post-decompression degradation. .......................... 22

3.3  Interpretation of the proposed compression method as a preprocessing stage followed by a single standard compression. The demonstration here assumes that our procedure runs $T$ iterations. .......................... 25

3.4  PSNR-bitrate curves comparing our approach to competing methods for three grayscale images (see also Table 3.1). The post-decompression deterioration is a $15 \times 15$ Gaussian blur kernel (standard deviation 0.6). .......................... 27

3.5  The images used in the experiments. .......................... 29

3.6  Demonstrating the intermediate and resultant images of the regular and the proposed compression methods for a post-decompression deterioration of a Gaussian blur kernel. The image is ’Starfish’ (a segment of $256 \times 256$ pixels is shown). .......................... 31

3.7  LCD motion blur experiment for the ’Shields’ sequence. The performance of our method in its PSNR-oriented (solid blue line) and smoothness-oriented (dashed blue line) modes is compared to preprocessing using total-variation deblurring [63] in its PSNR-oriented (solid red line) and smoothness-oriented (dashed red line) parameter settings, and to a regular compression-decompression procedure without any additional processing (solid black line). The average frame-PSNR and average frame-SSIM are evaluated in (a) and (b), respectively. .......................... 37
3.8 LCD motion blur experiment for the 'Stockholm' sequence. The performance of our method in its PSNR-oriented (solid blue line) and smoothness-oriented (dashed blue line) modes is compared to preprocessing using total-variation deblurring [63] in its PSNR-oriented (solid red line) and smoothness-oriented (dashed red line) parameter settings, and to a regular compression-decompression procedure without any additional processing (solid black line). The average frame-PSNR and average frame-SSIM are evaluated in (a) and (b), respectively.

3.9 LCD motion blur experiment for the sequence 'Shields' (grayscale, 120 frames at 60fps, 480x480 pixels) where a global horizontal motion of -3 pixels/frame causes the blur. An exemplary frame segment is presented. (a) the original frame segment. (b)-(f) are the displayed frame segments using each of the five examined methods. (g)-(k) are the simulated perceived frame segments corresponding to each of the displayed frames. The presented PSNR and SSIM evaluations are for the complete perceived frame, and the bit-rates are those measured for the compression of the complete sequence of 120 frames.

3.10 LCD motion blur experiment for the sequence 'Shields'. Here we present the difference images between the perceived and the original frame segments, for each of the methods presented in Fig. 3.9: (a) regular, (b) TV smoothness-oriented, (c) TV PSNR-oriented, (d) proposed smoothness-oriented, and (e) proposed PSNR-oriented. The difference images are presented in grayscale by scaling the value range of $[-50, 50]$ for the regular approach, and the value range of $[-15, 15]$ for the other methods.

4.1 The general system structure considered in this chapter.

4.2 Experiment for one-dimensional signal and adaptive tree-based coding in an acquisition-rendering system. The amplitude-modulated chirp source signal is the blue curve in (a) and (b). The red curves in (a) and (b) are the system output (rendered) signals resulting from regular compression and the proposed method, respectively. (c) Comparison of PSNR-bitrate curves.

4.3 HEVC coding of 10 video frames. The curves present the PSNR of the system rendered output with respect to the high-resolution source.

4.4 Coding a group of 10 frame from the 'Stockholm' sequence (spatial portion of 480x480 pixels). (a) The third source frame. (b) the rendered frame using regular compression (28.29 dB at 2.37 bpp). (c) the rendered frame using the proposed compression (29.45 dB at 1.34 bpp).
4.5 Coding a group of 10 frame from the ‘Shields’ sequence (spatial portion of 480x480 pixels). (a) The third source frame. (b) the rendered frame using regular compression (27.93 dB at 2.41 bpp). (c) the rendered frame using the proposed compression (29.31 dB at 1.31 bpp).

5.1 The general network structure considered in this chapter.

5.2 The regular and proposed methods applied for the ‘Flower and Bugs’ image. The PSNR values in this figure are for the individual reconstructions, i.e., using the regular MSE and not the expected one from (5.2) that is used in the rest of the chapter.

5.3 Methods evaluation for the image ‘Flower and Bugs’.

6.1 The autocorrelation of the cyclo-stationary Gaussian signal used in the demonstration. (a) The circulant autocorrelation matrix in the signal domain, and (b) the corresponding eigenvalues obtained using the DFT decomposition.

6.2 Demonstrating the theoretic results for a denoising problem with a noise level of $\sigma^2_n = 5$. (a) The degenerated degradation operator in the signal domain $H = I$. (b) DFT-domain magnitude of the degradation filter. (c) Optimal waterfilling solution in DFT domain. (d) Optimal rate allocation in DFT domain.

6.3 Demonstrating the theoretic results for a restoration problem with a noise level of $\sigma^2_n = 5$. (a) The degradation operator in the signal domain $H$. (b) DFT-domain magnitude of the degradation filter. (c) Optimal waterfilling solution in DFT domain. (d) Optimal rate allocation in DFT domain.

6.4 Empirical analysis for deblurring the Cameraman image using Algorithm 6.1 (i.e., without overlapping blocks and shifted grids). The parameter settings here are: $\mu = (6.67 \times 10^{-6})$ and $\beta = 0.01$.

6.5 Empirical analysis for deblurring the Cameraman image using Algorithm 6.2 considering 9 shifted block grids. The parameter settings here are: $\mu = (6.67 \times 10^{-6})/\text{number of shifts}$ and $\beta = 0.01$.

6.6 Empirical analysis for deblurring the Cameraman image using Algorithm 6.2 with JPEG2000 considering 9 shifted block grids. The compression parameter is the compression ratio given to the JPEG2000 compression. The parameter settings here are: 9 shifts, $\mu = (1.33 \times 10^{-4})/\text{number of shifts}$, $\beta = 25 \times 10^{-4}$. 

Technion - Computer Science Department - Ph.D. Thesis PHD-2018-13 - 2018
6.7 The deblurring experiment (settings #2) for the Cameraman image (256 × 256). (a) The underlying image. (b) Degraded image (20.76 dB). (c) Restored image using Algorithm 3 with JPEG2000 compression (28.10 dB). (d) Restored image using Algorithm 3 with HEVC compression (30.14 dB).

6.8 The deblurring experiment (settings #2) for the Barbara image (512 × 512). (a) The underlying image. (b) Degraded image (22.49 dB). (c) Restored image using Algorithm 3 with JPEG2000 compression (25.18 dB). (d) Restored image using Algorithm 3 with HEVC compression (27.72 dB).

6.9 The inpainting experiment (80% missing pixels) for the Barbara image (512 × 512). (a) The original image. (b) Deteriorated image. (c) Restored image using Algorithm 3 with JPEG2000 compression (24.83 dB) (d) Restored image using Algorithm 3 with HEVC compression (28.80 dB).

6.10 The inpainting experiment (80% missing pixels) for the House image (256 × 256). (a) The original image. (b) Deteriorated image. (c) Restored image using Algorithm 3 with JPEG2000 compression (30.50 dB) (d) Restored image using Algorithm 3 with HEVC compression (33.10 dB).

7.1 Examples of scalar quantizers.

7.2 Optimal linearization of a normalized two-level quantizer as function of the local interval center \((x_0)\) and length \((\delta)\).

7.3 Optimal linearization of a normalized uniform quantizer as function of the local interval center \((x_0)\) and length \((\delta)\).

7.4 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.

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

7.6 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 \(\hat{A}\), and (c)-(d) show the values of \(\hat{b}\)'s components. (e)-(f) show the corresponding approximation errors of the two-transform domain elements.

7.7 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 \(A\), and (e)-(f) show the values of \(b\)'s components.

7.8 Interpretation of a diagonal \(\hat{A}^*\) as a transform-domain filter that depends on \(\delta\). Here \(N = 32\) and the \(i^{th}\) quantization step is \(\Delta_i = 2^{i/4}\).
7.9 Interpretation of $A^*$ as a signal-domain filter that depends on $\delta$. Presented here for the case of $\delta = 50$ from Fig. 7.8c, incorporated in a DCT-based coding. .......................................................... 131

7.10 Reconstruction of Lena (256x256) from scalar quantization at 4bpp. . . 134

7.11 Reconstruction of Lena (512x512) from JPEG compression at 0.363bpp. 135

7.12 Reconstruction of Barbara (512x512) from JPEG2000 compression at 0.40bpp. .......................................................... 138

7.13 Reconstruction of Lena (128x128 Portion) from HEVC compression at 0.639bpp. .......................................................... 142

7.14 Reconstruction of the $5^{th}$ frame of Highway (CIF) from Motion-JPEG2000 compression at 0.08bpp. .......................................................... 149

7.15 Reconstruction of News (CIF) from Motion-JPEG2000 compression at 0.27bpp. .......................................................... 150
Abstract

Compression and restoration are two fundamental data processing tasks. Lossy compression represents the given data using a binary description that trades off the reconstructed signal quality with the representation bit-cost to satisfy storage or transmission constraints. Restoration methods estimate an unknown signal from its degraded version resulting from inevitable imperfections of practical devices. Since typical systems involve various degradations and also require data storage and transmission, the restoration and compression problems have various interesting connections of great importance. This thesis examines several challenging problems at the intersection of the compression and restoration topics.

In the main part of this thesis we propose a compression framework for rate-distortion optimizations that are intricate due to unusual distortion metrics. Using the alternating direction method of multipliers (ADMM) technique we develop an optimization approach addressing the difficult problem by iterative solution of easier tasks, including standard compression applications. We believe that the general iterative optimization framework proposed here entails a great potential for a variety of challenging compression problems associated with unusual distortion metrics. The usefulness of our general design is established here using a thorough study of its implications to problems connecting compression, degradation operators, and restoration processes.

We study a typical system structure where a signal is first acquired, then compressed for transmission or storage, and eventually rendered using an imperfect device. While the resulting quality of the system output signal may severely be affected by the acquisition and rendering processes, these degradations are usually ignored in the compression stage, leading to an overall sub-optimal system performance. Using our ADMM-based compression methodology we optimize the system’s performance with respect to end-to-end reconstruction error versus the compression bit-cost, showing that the design of the new globally-optimized compression reduces to a standard compression of a ”system adjusted” signal. Essentially, we propose a new practical framework for the information-theoretic problem of remote source coding. The main ideas of our method are further illustrated using rate-distortion theory for Gaussian signals. We experimentally demonstrate our framework for image and video compression using the state-of-the-art HEVC standard, adjusted to several system layouts including acquisition and rendering models. The experiments establish our method as the best approach for
optimizing the system performance at high bit-rates from the compression standpoint.

Moreover, we explore the topic of signal restoration using complexity regularization, quantifying the compression bit-cost of the signal estimate. Based on our ADMM-based approach, we present new restoration methods relying on repeated applications of standard compression techniques. Thus, we restore signals by leveraging state-of-the-art models designed for compression. Our approach utilizes a new shift-invariant complexity regularizer, measuring the bit-cost of all the shifted forms of the estimate, promoting restoration using averaging of decompressed outputs gathered from compression of shifted signals. On the theoretical side, we present a rate-distortion theoretic analysis for restoration of a Gaussian signal. The presented experiments show good results for image deblurring and inpainting using the JPEG2000 and HEVC compression standards.

Another research branch thoroughly demonstrates that existing denoising methods can be directly employed in a post-decompression processing stage to improve compression. This representative case emphasizes further this thesis’ findings on the symbiotic relations between compression and restoration methods.
Abbreviations and Notations

List of Notations:

- $C(\cdot)$: in Chapters 2-6: a general lossy compression function
  - in Chapter 7: a general lossy compression-decompression function
- $F(\cdot)$: a general decompression function
- $B$: the discrete set of binary compressed representations supported by $C$
- $S$: the discrete set of decompressed signals supported by $C$ and $F$
- $D(\cdot, \cdot)$: a general distortion metric
- $R(\cdot)$: a general bit-cost evaluation function
- $C_b(\cdot)$: a block-level lossy compression function
- $F_b(\cdot)$: a block-level decompression function
- $B_b$: the discrete set of binary compressed representations supported by $C_b$
- $S_b$: the discrete set of decompressed signal blocks supported by $C_b$ and $F_b$
- $R_b(\cdot)$: the block-level bit-cost evaluation function
- $\mathcal{I}$: the set of indices corresponding to the non-overlapping blocks of the signal
- $P_i$: the matrix extracting the $i^{th}$ block from the complete signal
- $P_A$: the projection matrix onto the range of the linear operator $A$
- $I$: the identity matrix
- $t$: iteration number
- $I(\cdot, \cdot)$: mutual information
- $E\{\cdot\}$: expectation
- $A^*$: the conjugate transpose of the matrix $A$
- $A^+$: the pseudoinverse of the matrix $A$
- $F$: the discrete Fourier transform (DFT) matrix
- $x^F$: the DFT-domain representation of the vector $x$
- $x_k^F$: the $k^{th}$ component of the DFT-domain representation of the vector $x$
- $s(\cdot)$: a general regularization function (for restoration inverse problems)
List of Abbreviations:

ADMM : alternating direction method of multipliers
AFPSNR : average frame peak signal-to-noise ratio
BD-PSNR : Bjontegaard delta peak signal-to-noise ratio
bpp : bits per pixel
DFT : discrete Fourier transform
HEVC : high efficiency video coding
LCD : liquid crystal display
MSE : mean squared error
PSNR : peak signal-to-noise ratio
SSIM : structural similarity index
Chapter 1

Introduction

1.1 Signal Compression, Restoration, and Their Interplay

Compression and restoration are two fundamental signal processing problems. Each of them is a field of research rich in theoretical and practical studies. The main goal of lossy compression is to represent a given signal using a binary description satisfying the bandwidth or memory constraints of the respective transmission or storage system. To attain the representation bit-cost constraint, inaccurate reconstruction is allowed leading to more significant distortions of the decompressed signal as the bit budget decreases. Accordingly, the trade-off between the bit cost (or bit rate) and the resulting distortion has been extensively studied in various theoretical and practical frameworks for rate-distortion optimization, establishing it as a central problem in the signal processing and information theory fields.

Essentially, any digital system involves some unwanted degradation affecting the processed data. One classical example is the signal acquisition process, required for digitization of "real world" analog signals in order to be further processed/stored in a digital manner, introducing undesired deteriorations due to the practically imperfect sampling and quantization processes. Other examples from the image processing field relate to various degradation models involving additive noise and/or image distortions resulting from inevitable blur, pixel erasure, or resolution-reduction processes. Correspondingly, restoration methods estimate the unknown original signal given its degraded form and possibly based on some assumed general properties of the underlying signal. This main problem has been widely studied from various deterministic and statistical perspectives and based on a diversity of signal models and optimization strategies.

Since digital systems usually include a compression component in addition to other processes possibly degrading the data, there is a great interest in problems relating the compression and restoration topics. For example, a well-established research line suggests to denoise a signal corrupted by an additive white Gaussian noise using compression aspects embodied in complexity regularization [1, 2, 3, 4, 5]. Other studies suggested combined compression and denoising techniques (for examples, see [6, 7, 8, 9]). Another
interesting research path proposes an inpainting-based approach for image and video compression, where only representative pixels are coded and the reconstruction relies on restoring a complete signal via inpainting [10, 11, 12]. Additionally, the information-theoretic problem of remote source coding (originally known as noisy source coding [13, 14]) studies the lossy compression process of a degraded input and a possible deterioration of the decompressed data.

In this thesis we study various problems at the intersection of the compression and restoration topics. We propose new compression and restoration frameworks addressing challenging and broad problem settings that were previously considered impractical. Our designs, relying on contemporary optimization methods and signal models, are developed for general signals and experimentally demonstrated for images and videos. The main principles of our new methods are further discussed in analytic frameworks combining signal processing and information theoretic aspects.

1.2 Dissertation Contributions

1.2.1 An Iterative Optimization Methodology for Intricate Rate-Distortion Problems

We propose a compression framework for addressing complex rate-distortion optimization forms via iterative solution of easier problems emerging from the alternating direction method of multipliers (ADMM) [15] or other variable-splitting optimization techniques. The original compression problems are challenging due to unusual distortion metrics that, for example, do not allow a block-based treatment and, by that, impose significant computational difficulties on a direct treatment of the optimization problem for high dimensional signals. Remarkably, the utilization of variable splitting leads to an optimization process including a sequential application of standard compression techniques. One may assess the vast potential of the proposed compression methodology for various applications by contemplating the wide use of ADMM [16, 17, 18, 19, 20, 21] and the similar half quadratic splitting method [22, 23] for diverse challenging signal restoration tasks. In this thesis we present several manifestations of this general idea, each of them being a general framework addressing important problem settings that were previously neglected, possibly due to the formerly believed impracticability. We consider that our design philosophy is further applicable to compression problems beyond the cases presented in this thesis. In this vein, future compression architectures may consider joint optimization with respect to perceptual distortion metrics or enhancement filters. Moreover, the ideas presented in this thesis can be further developed to form a significant alternative to the Lagrangian rate-distortion optimization methods that originated in [24, 25] and dominated the image and video coding field for the last two decades (see, e.g., [26, 27, 28, 29]).
1.2.2 System Optimization from the Compression Standpoint

Lossy compression has a central role in information systems where the data may be inaccurately represented in order to meet storage-space or transmission-bandwidth constraints. While the compression is a crucial system-component, it is only an intermediate stage among data processing procedures that determine the eventual output of the system. For example, consider a common audio/visual system structure where the source signal is acquired and compressed for its storage/transmission, then the decompression is followed by a rendering stage producing the ultimate system output. Evidently, in this example, the quality of the system output is determined by the acquisition-rendering chain, and not only by the lossy compression stage. Nevertheless, the compression is usually designed independently of the system structure, thus inducing a sub-optimal rate-distortion performance for the overall system.

In this part of the thesis, we propose a compression approach defined by an operational rate-distortion optimization considering a given system structure. Specifically, we study a general flow (Fig. 1.1) where the compression is preceded by a linear operator distorting the input along with an additive white noise, and the decompression is followed by another linear operation. We formulate a rate-distortion optimization based on a quadratic distortion metric involving the system’s linear operators. For general linear operators and high dimensional signals, this intricate rate-distortion optimization is too hard to be directly solved. Consequently, we address this challenge using the ADMM technique [15], suggesting an iterative procedure that relies on the simpler tasks of standard compression (which is system independent!) and $\ell_2$-constrained deconvolution for the linear operators of the system.

Optimizing the system output quality from the compression standpoint is an attractive answer to the inherent tradeoff among distortion, bit-cost, and computational complexity. First, bits are wisely spent for representing signal components that will be important at the output of the overall system (for example, one should obviously not

![Figure 1.1: The general system structure addressed by the system-aware compression framework.](image-url)
code signal components belonging to the null space of the post-decompression operator). Second, the added computational load can be well accommodated at the compression stage that usually has access to rich computational resources and unlimited time, unlike the decompression stage.

Furthermore, the problem settings we address here resemble the framework of (lossy) compression aimed to facilitate computations applied on the decompressed data (see, e.g., [30, 31]). The ability of our method to adapt standard compression to post-decompression processing may be utilized also for such goals too.

1.2.3 Extension of Image and Video Compression Standards to Optimize Acquisition-Rendering Systems

In this part of the thesis we demonstrate our ADMM-based compression methodology for image and video compression using the state-of-the-art HEVC standard, adjusted to several system layouts (defined as particular cases of the general structure in Fig. 1.2) including acquisition and rendering models. We start with the case of a display device following the decompression and modeled as a known blur degradation, then we extend it to the particular problem of motion-blur perceived while viewing video on LCD devices (for more details on this degradation type see, e.g., [32, 33, 34]). As another example we consider an extended model for a multimedia distribution network where several display types are possible, each associated with its probability to be used and represented as a different known blur degradation. We also provide demonstrations for an acquisition-rendering system layout where the compression is preceded by an acquisition process modeled as a linear low-pass and sub-sampling filter distorting the
input along with an additive white noise, and the decompression is followed by a linear up-sampling and blur processes. The experiments established our method as the best approach for optimizing the system performance at high bit-rates from the compression standpoint.

1.2.4 The Information-Theoretic Perspective: Relation to the Remote Source Coding Problem and New Analysis

The above contributions essentially provide new viewpoints on the information theoretic problem of remote source coding (see, for examples, its origins in [13, 14] and their successors). Our contribution is in the deterministic setting versus the statistical perspective used in previous works. Specifically, we consider an operational rate-distortion problem for the compression of a given signal, based on a new distortion metric imposed by the lack of any explicit statistical model for the unknown source. Our settings lead to the remarkable result that, using the ADMM technique, one can employ a standard compression method to address remote source coding problems in much more complicated instances than were previously feasible (e.g., [35]).

Using rate-distortion theory, we then also study the examined problem in statistical settings, in conjunction with the distortion metric we proposed for the deterministic framework, for the case of a cyclo-stationary Gaussian source signal and linear shift-invariant system operators. Our results show that the initial rate-distortion optimization reduces to a reverse water-filling procedure adjusted to the system operators and considering the pseudoinverse-filtered version of the input signal. We use these theoretical results to explain concepts appearing in the practical method intended for non-Gaussian signals and general linear system operators.

1.2.5 Restoration by Compression: New Methods and Analysis

In this part of the thesis we study the topic of signal restoration using complexity regularization, quantifying the compression bit-cost of the signal estimate. The restoration problem considered here is formulated as a rate-distortion optimization and, consequently, directly relates to the optimization methodology we proposed and demonstrated for several compression problems. Interestingly, while complexity-regularized restoration has been an established concept prior to our work, solid practical methods were suggested only for the Gaussian denoising task, leaving more complicated restoration problems lacking a general constructive approach.

Here we present practical methods for complexity-regularized restoration of signals, accommodating deteriorations caused by a known linear degradation operator of an arbitrary form. Our iterative procedure, obtained using the ADMM approach, addresses the restoration task as a sequence of simpler problems involving $\ell_2$-regularized deconvolutions and standard compression techniques. Thus, we restore the signal by leveraging underlying models designed for compression. Additionally, we propose a
shift-invariant complexity regularizer, measuring the bit-cost of all the shifted forms of the estimate, extending our method to use averaging of decompressed outputs gathered from compression of shifted signals. On the theoretical side, we present an analysis of complexity-regularized restoration of a cyclo-stationary Gaussian signal from deterioration by a linear shift-invariant operator and an additive white Gaussian noise. The theory shows that optimal complexity-regularized restoration relies on an elementary restoration filter and compression that spreads reconstruction quality unevenly based on the energy distribution of the degradation filter. Remarkably, these ideas are realized also in the proposed practical methods. We also present experiments showing good results for image deblurring and inpainting using the JPEG2000 and HEVC compression standards.

1.2.6 Compression-Artifact Reduction via Denoising-based Postprocessing

This part of the thesis, deviating from the main stream of work described above, proposes a technique to reduce standard compression artifacts by postprocessing of decompressed images. 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 Plug-and-Play Prior framework [17, 18], suggesting the solution of general inverse problems via 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.
Chapter 2

An Iterative Optimization Approach for Intricate Compression Problems

2.1 Lossy Compression via Operational Rate-Distortion Optimization

Let us consider a signal in the form of an $N$-length real-valued column vector, $\mathbf{x} \in \mathbb{R}^N$, that should be compressed and represented as a binary sequence for its storage or transmission. The lossy compression process is described here using the function

$$ C : \mathbb{R}^N \rightarrow \mathcal{B}, $$

mapping the $N$-dimensional input-signal domain to the discrete set $\mathcal{B}$ of compressed representations in the form of variable-length binary descriptions. The compression of $\mathbf{x}$ is formulated as

$$ b = C(\mathbf{x}), $$

where $b \in \mathcal{B}$ is the compressed data to be stored or transmitted for usage according to the specific application. A corresponding reconstruction process receives the compressed data $b$ as its input and decompresses it via

$$ v = F(b), $$

where

$$ F : \mathcal{B} \rightarrow \mathcal{S} $$

mapping the compressed data to the corresponding reconstructed signal $\mathbf{v} \in \mathbb{R}^N$. 


maps the binary compressed representations in $\mathcal{B}$ to the respective decompressed signals in the discrete set $\mathcal{S} \subset \mathbb{R}^N$. The decompressed signal $v$ can be further processed or provided to a user. For example, in the case of visual signals, $v$ is often processed for its display.

Modern compression designs (see, e.g., [26, 27, 28, 29]) implement the compression mapping $C$ based on operational rate-distortion optimizations, a concept established in [24, 25, 26] and can be described in our settings as follows. For a given deterministic input signal $x$, an optimization problem is defined to determine its best compressed representation $b \in \mathcal{B}$ associated with the decompressed signal $v \in \mathcal{S}$. The optimization is defined to trade-off two contradicting properties of the representation: the required bit-cost and the resulting reconstruction distortions. First, the bit-cost of the binary representation $b \in \mathcal{B}$ is simply its length. Since by (2.3), each $b \in \mathcal{B}$ corresponds to one decompressed signal $v \in \mathcal{S}$, we can associate the bit-cost of a decompressed signal $v \in \mathcal{S}$ as the length of its binary description $b = F^{-1}(v)$. Accordingly, we define the function $R(v)$ to evaluate the bit-cost of the compressed binary representation associated with $v$. The second aspect of the optimization is the reconstruction error evaluation using a distortion metric $D(x, v)$ quantifying the distance between the compression input $x$ and its decompressed version $v$ (where the distortion value is a non-negative real value). A corresponding optimization form, oriented to constraints on the representation bit-cost stemming from storage space or transmission bandwidth limitations, suggests to minimize the distortion with respect to a bit-cost constraint, i.e.,

$$\hat{v} = \arg\min_{v \in \mathcal{S}} D(x, v)$$

subject to $R(v) \leq r$ \hspace{1cm} (2.5)

where $R(v)$ evaluates the length of the binary compressed description $b \in \mathcal{B}$ matched to the decompressed signal $v$, and $r \geq 0$ is the maximal representation bit-cost allowed. Another widely used optimization formulation addresses the minimization of the compression bit-cost under a constrained distortion level, that is,

$$\hat{v} = \arg\min_{v \in \mathcal{S}} R(v)$$

subject to $D(x, v) \leq d$ \hspace{1cm} (2.6)

where $d \geq 0$ is the allowed distortion level. As will be explained in the sequel, the developments and frameworks presented in this thesis are appropriate for each of the two optimization forms presented above – accordingly, without loss of generality, we focus in this chapter on the last optimization form (2.6). The constrained optimization (2.6) is commonly translated in contemporary compression methods (see, e.g., [24, 25, 26, 27, 28, 29]) into its unconstrained Lagrangian form

$$\hat{v} = \arg\min_{v \in \mathcal{S}} R(v) + \lambda D(x, v)$$ \hspace{1cm} (2.7)
where $\lambda \geq 0$ is a Lagrange multiplier matching to a distortion constraint $d_\lambda \geq 0$ (such coding without a pre-specified distortion constraint is prevalent, e.g., in video coding [29]).

Since we consider the compression of high-dimensional signals (i.e., $N$ is large) the discrete set $S$ is prohibitively large. Accordingly, a direct solution of the Lagrangian form in (2.7), by evaluating the optimization cost for all $v \in S$, is impractical for arbitrarily-structured distortion metrics $D(x, v)$. Therefore, compression methods are usually carefully designed to assure that the combination of $D(x, v)$, $S$, and $B$ leads to a computationally practical process. However, these computationally-efficient architectures are too simple and, consequently, limit the compression performance one could wish for, as will be explained next and in Chapters 3-5.

2.2 Standard Compression Designs: Using Squared-Error Distortion Metrics and Block-based Coding

A prevalent architecture promoting a practical instance of the Lagrangian rate-distortion optimization (2.7) relies on the elementary squared-error distortion metric, namely,

$$D(x, v) = \|x - v\|_2^2. \quad (2.8)$$

Consider a segmentation of the signal $x$ into a set of non-overlapping blocks $\{x_i\}_{i \in I}$, where each of them is a column vector of $N_b$ samples and $I$ is the set of indices corresponding to the non-overlapping blocks of the signal. Similarly, $x$ is decomposed into the set of its non-overlapping blocks $\{v_i\}_{i \in I}$. Then, we can rewrite (2.8) as

$$D(x, v) = \sum_{i \in I} \|x_i - v_i\|_2^2, \quad (2.9)$$

showing that, in the case of the basic squared-error metric, the total distortion equals to the sum of distortions corresponding to its non-overlapping blocks. Obviously, this property is satisfied for any segmentation of the signal into non-overlapping blocks. Here we will continue the discussion with respect to the case of equal-size blocks, promoting the usage of a single block-level compression process applied to each of the blocks.

Similar to the definitions given above for full-signal compression designs, the block-level process is defined using the function $C_b : \mathbb{R}^{N_b} \rightarrow B_b$, mapping the $N_b$-dimensional input signal-block domain to the discrete set $B_b$ of binary compressed representations of blocks. The respective block decompression procedure is defined by the function $F_b : B_b \rightarrow S_b$, mapping the binary compressed representations in $B_b$ to the corresponding decompressed signal blocks in the discrete set $S_b \subset \mathbb{R}^{N_b}$. We also define the bit-cost evaluation function $R_b(v_i)$ returning the number of bits required for the compressed representation associated with the decompressed signal block $v_i \in \mathbb{R}^{N_b}$. Consequently, the compression of the non-overlapping signal-blocks $\{x_i\}_{i \in I}$ leading to the decompressed
blocks $\{v_i\}_{i \in I}$, requires a total amount of bits defined as

$$R(v) = \sum_{i \in I} R_b(v_i).$$

Setting the description of the block-based compression architecture in the Lagrangian form (2.7) yields

$$\{\hat{v}_i\}_{i \in I} = \text{argmin}_{\{v_i\}_{i \in I} \in S_b} \sum_{i \in I} R_b(v_i) + \lambda \sum_{i \in I} \|x_i - v_i\|_2^2,$$

that reduces to a series of block-level rate-distortion Lagrangian optimizations, namely,

$$\text{For } i \in I : \quad \hat{v}_i = \text{argmin}_{v_i \in S_b} R_b(v_i) + \lambda \|x_i - v_i\|_2^2,$$

where these block-level optimizations are independent and refer to the same Lagrangian multiplier $\lambda$. Many compression designs rely on processing of blocks of sufficiently low dimensions, such that the block-level optimizations in (2.12) can be practically employed, for example, by evaluating the Lagrangian cost for all the candidate solutions in $S_b$, which is sufficiently small.

Let us define the following algebraic tools for block treatment. The matrix $P_i$ is defined to provide the $i$th block from the complete signal via the standard multiplication $P_i x = x_i$. Note that $P_i$ can extract any block of the signal, even one that is not in the non-overlapping grid $I$. Accordingly, the matrix $P_i^T$ locates a block in the $i$th block-position in a construction of a full-sized signal and, therefore, lets to express the complete signal as $x = \sum_{i \in I} P_i^T x_i$.

Returning to the block-level compression optimizations in (2.12), producing the decompressed signals $\{\hat{v}_i\}_{i \in I}$ that are used further to form the complete decompressed signal via

$$\hat{v} = \sum_{i \in I} P_i^T \hat{v}_i.$$  

Accordingly, the above block-based design can be associated with an effective discrete set of full-size decompressed signals, defined based on the block level set $S_b$ via

$$S = \left\{ v \mid v = \sum_{i \in I} P_i^T v_i, \quad \{v_i\}_{i \in I} \in S_b \right\}.$$  

This effective set of allowed decompressed signals expresses some aspects of the limited representation ability provided by designs using non-overlapping blocks.
2.3 The Proposed Iterative-Optimization Framework for Intricate Compression Problems

Many important compression problems require more complex distortion metrics than the elementary squared error presented in the former section. Examples for such compression problems may involve optimizations with respect to non-local processing/prediction architectures, consideration of enhancement filters or degradation processes, and perceptual metrics assessing subjective quality of audio/visual signals. In the wide majority of these settings, the Lagrange rate-distortion optimization does not reduce to a sequence of manageable block-level optimizations and, therefore, one cannot carry out the truly desired compression tasks.

In this section we present our general methodology for addressing complicated rate-distortion problems using iterative optimization processes. In Chapters 3-5 we will thoroughly study practical and theoretical aspects of several interesting instances of our general compression approach.

For the construction of our approach, we employ the alternating direction method of multipliers (ADMM) technique \[15\] to address the computationally challenging problem (2.7) when the distortion metric is arbitrarily structured. For a start, we split the optimization variable such that (2.7) becomes

\[
\hat{v} = \arg\min_{v \in S, z \in \mathbb{R}^N} R(v) + \lambda D(x, z) \quad \text{subject to} \quad v = z
\]  

where \( z \in \mathbb{R}^N \) is an auxiliary variable that is not directly constrained to the discrete set \( S \). Then, we apply the scaled form of the augmented Lagrangian and the method of multipliers [15, Ch. 2] on (2.15) and obtain an iterative process formulated as

\[
\begin{align*}
(\hat{v}^{(t)}, \hat{z}^{(t)}) &= \arg\min_{v \in S, z \in \mathbb{R}^N} R(v) + \lambda D(x, z) + \frac{\beta}{2} \left\| v - z + u^{(t)} \right\|^2_2 \\
(\hat{v}^{(t)}, \hat{z}^{(t)}) &= \arg\min_{z \in \mathbb{R}^N} \lambda D(x, z) + \frac{\beta}{2} \left\| z - \hat{v}^{(t)} \right\|^2_2 \\
u^{(t+1)} &= u^{(t)} + \left( \hat{v}^{(t)} - \hat{z}^{(t)} \right)
\end{align*}
\]

where \( t \) denotes the iteration number, \( u^{(t)} \in \mathbb{R}^N \) is the scaled dual variable, and \( \beta \) is an auxiliary parameter introduced by the augmented Lagrangian. We get the ADMM form of the problem by applying one iteration of alternating minimization on (2.16), leading to a sequence of easier optimizations:

\[
\begin{align*}
\hat{v}^{(t)} &= \arg\min_{v \in S} R(v) + \frac{\beta}{2} \left\| v - \hat{z}^{(t)} \right\|^2_2 \\
\hat{z}^{(t)} &= \arg\min_{z \in \mathbb{R}^N} \lambda D(x, z) + \frac{\beta}{2} \left\| z - \hat{v}^{(t)} \right\|^2_2 \\
u^{(t+1)} &= u^{(t)} + \left( \hat{v}^{(t)} - \hat{z}^{(t)} \right)
\end{align*}
\]
where \( \tilde{z}(t) = \tilde{z}(t-1) - u(t) \) and \( \tilde{v}(t) = \tilde{v}(t) + u(t) \). Nicely, the compression architecture \( \{S, R\} \) and the complex distortion metric \( D(\cdot, \cdot) \) were decoupled by the ADMM to distinct (and simpler) optimization tasks. Moreover, the problem (2.19) considers an optimization variable from a continuous domain (in contrast to the original discrete optimization problem).

The optimization formulation in (2.18) coincides with the Lagrangian rate-distortion optimization utilized for standard compression tasks employing the usual squared-error distortion metric (here the effective Lagrange multiplier is \( \lambda = \frac{\beta}{2} \)). Hence, we propose to replace the solution of (2.18) with an application of a standard compression method (and its corresponding decompression). Moreover, one can employ compression methods that do not exactly follow the Lagrangian optimization in (2.18). We refer to the standard compression and decompression as

\[
\begin{align*}
    b(t) &= \text{StandardCompress}(\tilde{z}(t), \theta) \\
    \hat{v}(t) &= \text{StandardDecompress}(b(t))
\end{align*}
\]

where \( \theta \) is a parameter generalizing the Lagrange multiplier role in regulating the rate-distortion trade-off (see Algorithm 2.1). The last generalizations establish the proposed procedure as a generic methodology for an optimized adjustment of standard compression methods to particular distortion metrics. The proposed generic method is summarized in Algorithm 2.1.

**Algorithm 2.1 The General Framework for Intricate Compression Problems**

1: Inputs: \( x, \theta, \tilde{\beta} \).
2: Initialize \( t = 0 \), \( \tilde{z}(0) = x \), \( u(1) = 0 \).
3: repeat
4:  \( t \leftarrow t + 1 \)
5:  \( \tilde{z}(t) = \tilde{z}(t-1) - u(t) \)
6:  \( b(t) = \text{StandardCompress}(\tilde{z}(t), \theta) \)
7:  \( \tilde{v}(t) = \text{StandardDecompress}(b(t)) \)
8:  \( \hat{v}(t) = \tilde{v}(t) + u(t) \)
9:  \( \tilde{z}(t) = \arg\min_{z \in \mathbb{R}^N} \lambda D(x, z) + \frac{\beta}{2} \|z - \tilde{v}(t)\|_2^2 \)
10: \( u(t+1) = u(t) + (\hat{v}(t) - \tilde{z}(t)) \)
11: until stopping criterion is satisfied
12: Output: The binary compressed data \( b(t) \).

The potential benefit from the proposed iterative optimization procedure (Algorithm 2.1) is clear: the intricate distortion metric was separated from a standard compression stage. The usefulness of the proposed approach depends on the specific problem addressed, i.e., the particular distortion metric that determines whether the optimization stage in (2.19) is computationally feasible such that a sufficiently accurate solution to (2.19) can be obtained. Another interesting problem-specific issue relates to the use of the ADMM approach for compression optimization problems that are discrete, non-convex,
and non-linear. Since convergence guarantees were provided only for the application of ADMM to convex problems [15], the benefits of the proposed approach should be examined for the specific problem structures addressed. Indeed, in Chapters 3-5 we show the usefulness of our approach to a variety of compression problems, and in Chapter 6 we demonstrate our framework for a restoration approach defined by complicated rate-distortion optimizations (where the bit-cost term is also intricate). For the diversity of problems examined, our approach provides excellent results. We also analyze the addressed problems using rate-distortion theory, justifying ideas implemented in our practical method.
Chapter 3

Optimized Pre-Compensating Compression

In imaging systems, following acquisition, an image/video is transmitted or stored and eventually presented to human observers using different and often imperfect display devices. While the resulting quality of the output image may severely be affected by the display, this degradation is usually ignored in the preceding compression. In this chapter we model the sub-optimality of the display device as a known degradation operator applied on the decompressed image/video. We assume the use of a standard compression path, and augment it with a suitable pre-processing procedure, providing a compressed signal intended to compensate the degradation without any post-filtering. Our approach originates from an intricate rate-distortion problem, optimizing the modifications to the input image/video for reaching best end-to-end performance. We address this seemingly computationally intractable problem using the alternating direction method of multipliers (ADMM) approach, leading to a procedure in which a standard compression technique is iteratively applied. We demonstrate the proposed method for adjusting HEVC image/video compression to compensate post-decompression visual effects due to a common type of displays. Particularly, we use our method to reduce motion-blur perceived while viewing video on LCD devices. The experiments establish our method as a leading approach for preprocessing high bit-rate compression to counterbalance a post-decompression degradation.

3.1 Introduction

Image and video signals have a significant, constantly growing, role in many contemporary applications. A fundamental need of image/video applications is to store and/or transmit a digital version of the signal, obeying a bit-budget constraint stemming from the available storage space or the communication channel bandwidth. This bit-budget limitation is managed by lossy compression that produces a compressed representation satisfying the bit-cost constraint at the expense of some distortion in the decompressed
signal. This systematic flow (see Fig. 3.1) usually ends with a human user watching the image/video on a display device. Accordingly, the quality of the viewed signal is determined by the compression, the imperfections of the display device, and the human visual system.

Lossy image and video compression methods trade-off the compressed-form bit-rate with distortion of the decompressed signal. Popular compression techniques (e.g., JPEG [36], JPEG2000 [37], HEVC [29]) substantially differ in their rate-distortion optimization strategies and the employed image/video models. However, these standard designs ignore other procedures possibly accompanying the compression, thus, may result in sub-optimal rate-distortion performance when considering the complete system.

In this work, we study an intriguing extension of the regular compression problem, where the decompressed signal is degraded by a known linear operator (see Fig. 3.2). Our goal is to compress by considering the squared error between the degraded decompression and the input image. The corresponding rate-distortion optimization has a challenging structure due to the degradation operator involved in the distortion term. We tackle the intricate optimization using the alternating direction method of multipliers (ADMM) approach [15], mapping the task to a sequence of easier problems including regular rate-distortion optimizations that are replaced with repeated applications of a standard compression technique. Remarkably, our iterative procedure generically adapts a regular compression method to consider the extended settings involving a post-decompression degradation.

The first part of our experiments considers the adjustment of HEVC image compression to a blur operator degrading the decompressed image. Our results demonstrate the effectiveness of the proposed approach, having superior rate-distortion performance compared to a regular HEVC compression. Another alternative to accommodate post-decompression degradation is by preceding the compression with a regular deblurring of the input, using the EPLL method [23]. Our method outperforms the EPLL-based approach at high bit-rate compression, reaching impressive average-PSNR (i.e., BD-PSNR [38]) gains of 2-3 dB.

As an important application of these ideas, we present a methodology for pre-compression treatment of motion-blur occurring while viewing videos on Liquid Crystal Displays (LCD). The prevalent technology of LCD devices relies on a hold-type mechanism, where each frame is constantly displayed until its replacement, resulting in

*Figure 3.1: The considered flow of an image/video that is first compressed and finally perceived by a human observer.*
delicate discontinuities of motions. The human eye tracks an object based on its smooth motion, trying to fix its location on the retina for a vivid perception. The smooth eye tracking of discontinuous motion displayed on LCD yields an unsteady positioning on the retina, causing a blurred perception of the moving object. This blur artifact is amplified for more rapid motions and/or when the video or the display frame-rates are inadequately low, implying too long constant-frame display duration. Importantly, motion blur due to the hold-type nature is still an issue of great interest in contemporary evaluations of LCD screens (for examples, see the technical reviews in [39] and the excellent experimental demonstrations therein) and considered as a crucial drawback of ultra high-definition displays [40].

Straightforward amendments for LCD motion-blur reduce the constant-frame display duration by black-frame insertion [41] that causes unwanted eye strains, or by interpolation-based frame-rate up conversion [42, 43, 44] that is computationally intensive and unsuited for complicated motion types. More sophisticated techniques [32, 33, 34] counteract the LCD motion blur by a pre-display frame filtering, designed based on blur models of the LCD hold-type behavior and the eye-tracking capability of the human visual system. These works achieved high PSNR gains using inverse filtering, [32], and the Lucy-Richardson deconvolution method [33], however, introduced subjectively annoying noise artifacts that were attenuated in [34] using spatio-temporal smoothness regularization.

While our application for LCD motion-blur reduction relates to the line of works [32, 33, 34], we are the first to address the problem via a pre-compression procedure suggesting computational and accuracy benefits. First, many video content types (e.g., entertainment) are compressed in offline settings rich in computation and time resources, contrasting the regular processing [32, 33, 34] intended for the display device. Accordingly, one can utilize our method in a video-on-demand system designed such that the display types are known and the suitable videos can be delivered to the users. Second, the blur-compensating filters make use of the current video-motion imperfectly estimated on the available data. While "on-device" methods should practically operate on decompressed frames leading to increased motion-estimation errors (especially at medium/low qualities), our approach uses the pre-compression frames for better motion estimation providing more accurate blur characterization and filtering. Nicely, our display-blur compensation is, in fact, constrained by the associated video coding procedure acting as a spatio-temporal complexity regularizer preferring smoother or other model-conforming signals costing less bits (see, e.g., in [45, 46]). Consequently, our motion-blur reduction technique provides impressive PSNR gains (with respect to the compression bit-rates) and a pleasing subjective quality.

This chapter is organized as follows. In Section 3.2 we present our method in its general form. In Section 3.3 the proposed approach is experimentally studied for adjusting HEVC image compression to balance a post-decompression blur. In Section 3.4 we employ our method for adapting HEVC video coding to reduce motion blur.
3.2 The Proposed Method

3.2.1 The Basic Rate-Distortion Optimization

We develop our method based on the system structure illustrated in Fig. 3.2 and explained next. First, an \( N \)-dimensional input signal, \( x \in \mathbb{R}^N \), goes through a lossy compression procedure resulting in a compressed binary description associated with an approximation of \( x \), denoted as \( v \in \mathbb{R}^N \), obtained after the decompression stage. However, the reconstruction \( v \) is further deteriorated, for instance, due to a sub-optimal display device. We consider here a linear deterioration operator, represented by the \( N \times N \) real-valued matrix \( H \). Then, the degraded decompressed signal is defined as

\[
\tilde{v} \triangleq Hv, \tag{3.1}
\]

the outcome of the entire process.

Our goal here is to optimize the compression procedure with respect to the squared error between the input signal \( x \) and the degraded decompression \( \tilde{v} \), that using (3.1) can be expressed as

\[
D_H (x, v) \triangleq \| x - Hv \|_2^2. \tag{3.2}
\]

Without loss of generality (as will be explained later), we develop our method with respect to a block-based compression design individually operating on blocks of \( N_b \) samples defined by a non-overlapping segmentation of the signal. We refer to members belonging to the grid of non-overlapping blocks via the set of indices \( I \). The block-level compression procedure is modeled as a vector quantizer having a codebook \( S_b \), being a finite set of block-reconstruction candidates and their respective variable-length binary codewords. Specifically, the block-reconstruction \( c \in S_b \) has a corresponding binary codeword of length \( R_b (c) \) defining the respective block bit-cost. Accordingly, the total bit-cost can be evaluated from the decompressed blocks \( \{ v_i \}_{i \in I} \) as the sum \( \sum_{i \in I} R_b (v_i) \).

We define the matrix \( P_i \) as a linear operator extracting the \( i^{th} \) block from the complete signal by the standard multiplication \( P_i v = v_i \). Then, the bit-cost of the entire signal

\[
\text{Technion - Computer Science Department - Ph.D. Thesis PHD-2018-13 - 2018}
\]

(a) Figure 3.2: Demonstration of the conceptual problem settings for compression that is oriented to post-decompression degradation.
can be expressed as

$$R(\mathbf{v}) = \sum_{i \in I} R_b(\mathbf{P}_i \mathbf{v}).$$ \hfill (3.3)

We use the quantities defined in (3.2)-(3.3) to formulate the rate-distortion optimization in the unconstrained Lagrangian form:

$$\hat{\mathbf{v}} = \arg\min_{\mathbf{v} \in S} \| \mathbf{x} - \mathbf{H} \mathbf{v} \|_2^2 + \lambda \sum_{i \in I} R_b(\mathbf{P}_i \mathbf{v})$$ \hfill (3.4)

where $\lambda \geq 0$ is the Lagrange multiplier associated with some total bit-cost constraint, and $\hat{\mathbf{v}}$ is the optimal decompressed signal among the candidates available in the effective full-signal codebook:

$$S = \left\{ \mathbf{c} \mid \mathbf{c} = \sum_{i \in I} \mathbf{P}_i^T \mathbf{c}_i, \quad \{\mathbf{c}_i\}_{i \in I} \in S_b \right\}$$ \hfill (3.5)

where the linear operator $\mathbf{P}_i^T$ places a block in the $i^{th}$ block location in a full-signal layout. One should note that, for an arbitrarily structured $\mathbf{H}$, the optimization (3.4) is difficult to solve since it does not allow the commonly used block-based treatment (for examples, see its various forms in the fundamental studies on operational rate-distortion optimization [24, 26, 27] and also in recent works [47, 48]).

### 3.2.2 Practical Iterative Procedure

The structural complication of the rate-distortion optimization (3.4) is facilitated using the ADMM strategy [15] as explained next. Initially, we define the auxiliary variable $\mathbf{z} \in \mathbb{R}^N$ letting us to reformulate the problem (3.4) into

$$\hat{\mathbf{v}}, \hat{\mathbf{z}} = \arg\min_{\mathbf{v} \in S, \mathbf{z} \in \mathbb{R}^N} \| \mathbf{x} - \mathbf{H} \mathbf{z} \|_2^2 + \lambda \sum_{i \in I} R_b(\mathbf{P}_i \mathbf{v})$$ \hfill (3.6)

s.t. \hfill (3.7)

$$\mathbf{z} = \mathbf{v}.$$  

Then, considering (3.6) via its augmented Lagrangian (in its scaled version [15, Ch. 2]) leads to an iterative procedure, where the $t^{th}$ iteration is

$$\left(\hat{\mathbf{v}}^{(t)}, \hat{\mathbf{z}}^{(t)}\right) = \arg\min_{\mathbf{v} \in S, \mathbf{z} \in \mathbb{R}^N} \| \mathbf{x} - \mathbf{H} \mathbf{z} \|_2^2 + \lambda \sum_{i \in I} R_b(\mathbf{P}_i \mathbf{v}) + \beta \| \mathbf{v} - \mathbf{z} - \mathbf{u}^{(t)} \|_2^2$$ \hfill (3.8)

$$\mathbf{u}^{(t+1)} = \mathbf{u}^{(t)} + \left(\hat{\mathbf{v}}^{(t)} - \hat{\mathbf{z}}^{(t)}\right)$$ \hfill (3.9)

where $\mathbf{u}^{(t)} \in \mathbb{R}^N$ is the scaled dual variable and $\beta$ is an auxiliary parameter originating at the Lagrangian.

Since each of the optimization variables in (3.8) participates only in two of the three terms in the cost function and, therefore, one iteration of alternating minimization
provides us the ADMM form that iterates over the following manageable optimizations:

\[
\hat{v}(t) = \arg\min_{v \in S} \frac{\beta}{2} \| \tilde{z}(t) - v \|_2^2 + \lambda \sum_{i \in I} R_\theta(P_i v) \quad (3.10)
\]

\[
\hat{z}(t) = \arg\min_{z \in \mathbb{R}^N} \| x - \mathbf{H}z \|_2^2 + \frac{\beta}{2} \| z - \tilde{v}(t) \|_2^2 \quad (3.11)
\]

\[
u^{(t+1)} = \nu^{(t)} + (\hat{v}^{(t)} - \hat{z}^{(t)}) \quad (3.12)
\]

where \( \tilde{z}(t) = \hat{z}^{(t-1)} - u^{(t)} \) and \( \tilde{v}(t) = \hat{v}^{(t)} + u^{(t)} \). The analytic solution of the second-stage problem in (3.11) is

\[
\hat{z}(t) = \left( \mathbf{H}^T \mathbf{H} + \frac{\beta}{2} \mathbf{I} \right)^{-1} \left( \mathbf{H}^T x + \frac{\beta}{2} \tilde{v}(t) \right), \quad (3.13)
\]

thus, exhibiting optimization (3.11) as a weighted averaging operation.

Importantly, the first stage (3.10) is a rate-distortion optimization compatible with a block-based treatment and considering the regular squared-error metric for the compression of \( \tilde{z}(t) \), obtained in the second stage of the former iteration. Moreover, this full-image rate-distortion optimization is done for a Lagrange multiplier of value \( \tilde{\lambda} = \frac{2\lambda}{\beta} \). We denote the compression-decompression procedure associated with (3.10) as

\[
\hat{v}(t) = \text{CompressDecompress}_{\lambda}(\tilde{z}(t)). \quad (3.14)
\]

We further suggest using a standard compression method as the compression-decompression operator (3.14). While many compression methods do not follow the exact rate-distortion optimization we got in our mathematical development (3.10), we still suggest using such techniques as replacements for (3.10). Additionally, since various compression methods do not rely on Lagrangian optimization, their operating parameters may differ (for example, quality parameters, compression ratios, or output bit-rates). Accordingly, we present the suggested algorithm with respect to a general compression procedure that its output bit-cost is directly or indirectly affected by a parameter denoted as \( \theta \). This generalization is used in Algorithm 3.1.

The replacement of the rate-distortion optimization formulation in (3.10) with a standard compression-decompression process (3.14) is motivated by a similar development step used in the Plug-and-Play Priors method [17] for image restoration, where an optimization stage corresponding to a Maximum A-Posteriori (MAP) Gaussian denoising problem is replaced with the application of an existing denoiser (such as BM3D [49]). In both cases (ours and in [17]), the application of an arbitrary compression/denoising method means that the convexity of the optimization problem cannot be guaranteed and, therefore, in some cases the optimization may not converge. Accordingly, the
Algorithm 3.1 Proposed Method: Compression Adjusted to Post-Decompression Degradation

1: Inputs: $x, \theta, \beta$.
2: Initialize $\hat{z}^{(0)} = x, u^{(1)} = 0$.
3: $t = 1$
4: repeat
5:   $\tilde{z}^{(t)} = \hat{z}^{(t-1)} - u^{(t)}$
6:   $\hat{v}^{(t)} = \text{CompressDecompress}_\theta(\tilde{z}^{(t)})$.
7:   $\tilde{v}^{(t)} = \hat{v}^{(t)} + u^{(t)}$
8:   $\hat{z}^{(t)} = \left(HTH + \frac{\beta}{2}I\right)^{-1}(HTx + \frac{\beta}{2}\tilde{v}^{(t)})$
9:   $u^{(t+1)} = u^{(t)} + (\hat{v}^{(t)} - \hat{z}^{(t)})$
10: $t \leftarrow t + 1$
11: until stopping criterion is satisfied
12: Output: Binary compressed data obtained in the last application of Stage 6.

Figure 3.3: Interpretation of the proposed compression method as a preprocessing stage followed by a single standard compression. The demonstration here assumes that our procedure runs $T$ iterations.

implementations we present in Sections 3.3-3.4 include a divergence detection mechanism as part of the stopping criterion, this feature is explained later in this chapter. Studying the convergence/divergence properties of our framework (in the spirit of the analysis in [21]) is left to a future work.

We can further interpret the proposed iterative compression approach as a preprocessing stage coupled with a single standard compression, being the one applied in the last iteration as the determining stage outputing the compressed binary data (see Fig. 3.3). Remarkably, our compression output is compatible with a standard decompression process. The overall quality improvement suggested by our method obviously entails an increased computational cost that, nevertheless, is distributed between the encoder and the decoder stages in the following attractive structure: the decoder complexity remains as in the standard form, while the encoder has the increased computational load of repeatedly applying standard compressions (3.14) and the $\ell_2$-constrained deconvolutions (3.13). This system layout is beneficial for applications where the compression can be carried out offline in environments rich in computational and time resources, whereas the decompression on the display devices should be of a low computational cost due to run-time and energy-consumption limitations.

25
3.3 Adjusting HEVC Image Compression to Blurry Decompression

In this section we demonstrate our approach for adapting HEVC’s still-image compression (in its version implemented in the BPG image format [50]) to a blur deteriorating the decompressed image. The experiment goal here is to study our method with respect to alternative processing strategies. Our ideal settings here, considering a known Gaussian blur kernel, serve as a preliminary stage to the intricate application presented in Section 3.4. The degradation operator $H$ is associated with a Gaussian blur kernel of standard deviation 0.6 and $15 \times 15$ pixels size. We consider a shift-invariant degradation, thus, efficiently degrade an image using a two-dimensional convolution with the blur kernel.

The Peak Signal-to-Noise Ratio (PSNR) is defined here based on the squared-error distortion (3.2) and can be written as

$$PSNR = 10 \log_{10} \left( \frac{P^2}{N \|x - \tilde{v}\|^2_2} \right)$$  \hspace{1cm} (3.15)

where $x$ and $\tilde{v}$ are the input and the degraded decompressed signals, respectively, and $N$ is the signal dimension. The maximal value attainable by the examined signals is denoted as $P$ that, e.g., equals to 255 for grayscale images with pixel values in the range $[0, 255]$. In the PSNR computation we ignore margins of 35 pixels along the borders of the image to exclude effects of specific boundary conditions used in the applied convolutions.

3.3.1 Competing Methods

We compare our approach to three competing strategies also considering HEVC image compression. In Figure 3.4 we compare the rate-distortion curves of the various methods corresponding to operating their HEVC component using quantization parameter (QP) values between 1 to 49 in jumps of 3. The examined compression procedures are:

Regular compression without any pre/post processing

This is the baseline approach where a regular compression-decompression application is followed by deterioration. Obviously, since this procedure ignores the degradation, it is the cheapest in computations and provides an inferior performance (see the solid-line black curves in Fig. 3.4).

Pre-compression pseudoinverse filtering of the input image

We consider here an ideal pseudoinverse filter, matched to the known degradation operator $H$, employed as a pre-compression filter. The numerical crudity of the pseudoinverse filter yields a very large dynamic range of pixel values, hence, requiring shifting
Figure 3.4: PSNR-bitrate curves comparing our approach to competing methods for three grayscale images (see also Table 3.1). The post-decompression deterioration is a $15 \times 15$ Gaussian blur kernel (standard deviation 0.6).
Table 3.1: Image Compression Considering Post-Decompression Deterioration of a Gaussian Blur. Average PSNR Gains (measured using BD-PSNR)

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Image</th>
<th>All Bit-Rates</th>
<th>High Bit-Rates</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Proposed over Regular</td>
<td>Proposed over EPLL</td>
</tr>
<tr>
<td>TESTIMAGES 300x300</td>
<td>Almonds</td>
<td>5.49</td>
<td>0.22</td>
</tr>
<tr>
<td></td>
<td>Flowers</td>
<td>5.57</td>
<td>0.41</td>
</tr>
<tr>
<td></td>
<td>Billiard balls</td>
<td>5.57</td>
<td>-0.08</td>
</tr>
<tr>
<td></td>
<td>Cards</td>
<td>6.08</td>
<td>0.90</td>
</tr>
<tr>
<td></td>
<td>Duck toys</td>
<td>3.62</td>
<td>-0.62</td>
</tr>
<tr>
<td></td>
<td>Garden table</td>
<td>3.78</td>
<td>0.29</td>
</tr>
<tr>
<td>UCID 384x512</td>
<td>Garden house</td>
<td>3.21</td>
<td>1.14</td>
</tr>
<tr>
<td></td>
<td>House and lawn</td>
<td>2.41</td>
<td>0.50</td>
</tr>
<tr>
<td></td>
<td>Tree</td>
<td>3.57</td>
<td>1.16</td>
</tr>
<tr>
<td></td>
<td>Garden</td>
<td>3.16</td>
<td>1.49</td>
</tr>
<tr>
<td></td>
<td>Teddy bear</td>
<td>3.65</td>
<td>-0.39</td>
</tr>
<tr>
<td></td>
<td>Duck</td>
<td>5.25</td>
<td>0.06</td>
</tr>
<tr>
<td>Berkeley 481x321</td>
<td>Starfish</td>
<td>4.72</td>
<td>-0.36</td>
</tr>
<tr>
<td></td>
<td>Bears</td>
<td>3.78</td>
<td>0.42</td>
</tr>
<tr>
<td></td>
<td>Boats</td>
<td>3.29</td>
<td>0.43</td>
</tr>
<tr>
<td></td>
<td>Butterfly</td>
<td>4.76</td>
<td>0.42</td>
</tr>
<tr>
<td></td>
<td>Flower and Bugs</td>
<td>4.18</td>
<td>0.02</td>
</tr>
<tr>
<td></td>
<td>Sea</td>
<td>4.58</td>
<td>0.21</td>
</tr>
</tbody>
</table>
and scaling before the compression and the inverse adaptations after decompression (before degradation). Since the pseudoinverse filtered image is far from obeying natural-image characteristics (e.g., smoothness), it is inefficiently compressed by a standard compression technique. This drawback results in performance inferior even to regular compression without any processing (see Fig. 3.4). Moreover, the unusual signals provided by the pseudoinverse filter can be compressed using HEVC to a limited range of bit-rates, for examples, observe the rightmost working-points of the magenta curves obtained using HEVC compression at the very high quality corresponding to $QP = 1$. This exemplary approach exhibits the challenges in pre-compression processing.

**Pre-compression filtering via the Expected Patch Log Likelihood (EPLL) method**

We define the main competing method to employ a pre-compression filtering in the form of the EPLL deblurring method relying on a Gaussian Mixture Model (GMM) prior learned for natural images (see [23]). Indeed, the processed image conforms with natural-image attributes, thus, efficiently compressed by HEVC leading to a good rate-distortion performance considering the degraded decompressed image (see the solid-line green curves in Fig. 3.4). In the EPLL experiments we used the implementation published by the authors of [23] with parameters we found to improve the rate-distortion performance considered here.
3.3.2 Our Method: Experiment Settings

We now turn to evaluate our method with respect to the above three reference techniques. In the implementation of Algorithm 3.1 we set $\beta$ to a value depending on the specific quantization parameter (QP) given to the HEVC compression:

$$
\beta = \begin{cases} 
0.03, & \text{for } 0 \leq QP \leq 20 \\
0.05, & \text{for } 20 < QP \leq 30 \\
0.10, & \text{for } 30 < QP \leq 40 \\
0.35, & \text{for } 40 < QP \leq 45 \\
0.45, & \text{for } 45 < QP \leq 51 
\end{cases}
$$

(3.16)

Recall that HEVC QP values are integers between 0 to 51, where a lower value yields a higher quality. The stopping criterion was defined to a maximal number of 40 iterations or to end earlier when $\hat{v}^{(t)}$ and $\hat{z}^{(t)}$ are detected to converge or diverge. The convergence/divergence detection relies on the total absolute difference between $\hat{v}^{(t)}$ and $\hat{z}^{(t)}$ in each iteration, namely, $w^{(t)} = \|\hat{v}^{(t)} - \hat{z}^{(t)}\|_1$. Accordingly, we determine convergence when $|w^{(t)} - w^{(t-1)}| < 0.2$ for three consecutive iterations. Divergence is identified when $w^{(t)} - w^{(t-1)} > 50$ and, in that case, the algorithm output is taken from the preceding iteration. Note that the threshold values given here for convergence/divergence detection depend on the signal dimension and the typical value-range (the thresholds specified here are for signal with values in the range $[0,1]$).

The rate-distortion curves of our method, presented in Fig. 3.4 as the blue solid-lines, outperform the other pre-compression techniques at the high bit-rate range. The PSNR gains at high bit-rates are significant (note the wide PSNR range of the graphs that may visually mislead), reaching improvements of several dBS. These impressive PSNR gains at high bit-rates were further established by examining 18 images (see Fig. 3.5 and Table 3.1) collected from three different datasets (TESTIMAGES [51], UCID [52], and Berkeley [53]). The comparison in Table 3.1 considers the average PSNR difference between performance curves (e.g., see Fig. 3.4) of the proposed, the EPLL-based, and the regular methods. The average PSNR differences between curves were calculated using the BD-PSNR metric [38, 54], for the entire bit-rate range (i.e., the complete curves generated for QP values between 1 to 49 in jumps of 3) and for curve segments corresponding to high bit-rates (defined by QP values 1, 7, 13 and 19).

In Figure 3.6 we present visual results for the 'Starfish' image. First, we examine the regular compression procedure where the input image (Fig. 3.6a) is compressed using HEVC at a bit-rate of 5.061 bits per pixel (bpp), leading to the decompressed image in Fig. 3.6b (note that this is the pre-degradation image). Then, obviously, the post-degradation decompressed image (Fig. 3.6c) suffers from tremendous blur affecting also the PSNR (measured with respect to the precompression image). In contrast, our approach processes the input image such that the compression in the last iteration gets a sharpened version (see Fig. 3.6d) adjusted to the specifically known blur operator,
Figure 3.6: Demonstrating the intermediate and resultant images of the regular and the proposed compression methods for a post-decompression deterioration of a Gaussian blur kernel. The image is 'Starfish' (a segment of 256 × 256 pixels is shown).
then, the compressed image at bit-rate 4.296 bpp leads to a degraded decompression with moderate blur effects (Fig. 3.6f) and PSNR improvement of 15.25 dB with respect to the regular compression at a higher bit-rate.

The experiments show that the current implementation of our approach is significantly better than the considered alternatives for compression at the high bit-rate range. We consider this behavior to emerge from the following two facts. First, the details of the pre-compression processing are preserved better when high bit-rate compression is applied. Second, at high bit-rate compression the employed quantization is finer, thus, the discrete optimization problem imitates more closely an optimization over a continuous domain - which is more suitable for the ADMM optimization technique.

3.4 Application to LCD Motion-Blur Reduction

3.4.1 The LCD Motion Blur and Its Modeling

A prominent type of post-decompression degradations is defined as the inevitable artifacts arising due to various display device technologies. For instance, the formerly prevalent Cathode Ray Tube (CRT) displays employ an impulse-type mechanism where video frames are instantaneously presented, producing a good perceptual motion-continuity and unpleasing flickering artifacts. Here we focus on the current Liquid Crystal Display (LCD) technology, the ultimate successor of CRT, being a hold-type display where each frame is constantly presented for a duration of \((frame\ rate)^{-1}\) seconds, referred to as the hold time (e.g., for a rate of 60 frames per second the hold time is 16.6 milliseconds). While this hold-type architecture is flickering free, it suffers from a non-smooth presentation of motions that cause blur in the image perceived by the viewer. Specifically, the human eye pursues constant motion of an object to fix its image location on the retina for a detailed perception. While motion presented on an LCD device has delicate discontinuities, the eye still tracks it as if it was continuous and, thus, suffers from corresponding spatial displacements on the retina that blur the perceived image.

Additional LCD blur stems from the response time, which is the duration taking a pixel to change its intensity, that despite its reduction along the years still introducing some amount of blur (see, e.g., [39]). As in [32, 33, 34], we consider motion blur arising only from the hold-type method of the LCD. The reader is referred to [32, 33, 34, 55, 40] for additional discussions on the above described CRT and LCD motion artifacts. In this section we will rely on existing models and problem settings addressing the LCD motion blur, and utilize our method from Section 3.2 for adjusting HEVC video coding to pre-compensate the perceived motion blur.

The two prominent signal-processing models of LCD motion blur were developed in the frequency [32] and signal [55] domains, considering the display impulse response and the human visual system (HVS) mechanisms of motion tracking and spatio-temporal
low-pass filtering. A later model [34] interpreted the former analyses to the case of discrete video signals, and approximated the temporal blur operator as an intra-frame spatial degradation determined by the current motion. This spatio-temporal equivalence of motion-blur degradation due to the hold-type nature of the LCD was used in various forms in [56, 57, 58, 59]. We here follow the model for LCD motion-blur given in [34] to be aligned with the problem settings defined therein.

The \( k \)th frame of the displayed video is a \( W \times H \) two-dimensional discrete signal, comprised of \( N_f = W \cdot H \) pixels that their column-stack form is denoted here as \( v_k \in \mathbb{R}^{N_f} \). The perceived image corresponding to the \( k \)th frame is

\[
\tilde{v}_k = H_k v_k
\]

(3.17)

where \( H_k \) is a \( N_f \times N_f \) matrix representing the motion-blur as a spatial operator. The \( r \)th row of \( H_k \) specifies the blur operation producing the \( r \)th pixel of the degraded frame. The local blur operation is determined by the associated motion vector that may vary for different pixels. For example, assume the \( r \)th pixel corresponds to the motion vector \((0, -3)\) describing a vertical motion upwards in 3 pixels with respect to the previous frame, accordingly, (assuming \( r \) corresponds to a coordinate sufficiently distant from the frame boundaries in its 2D arrangement) the \( r \)th row of \( H_k \) should be formed from the following entries

\[
H_k [r, c] = \begin{cases} 
\frac{1}{3}, & \text{for } c = r, r - 1, r - 2 \\
0, & \text{otherwise}
\end{cases}
\]

(3.18)

A detailed numerical method for defining the blur kernel given a motion vector was described in [34]. An important particular case occurs when the frame motion is global, leading to a block-circulant matrix \( H_k \).

### 3.4.2 The Proposed Method for Motion-Blur Reduction

We now turn to translate our general method given in Section 3.2 to the specific degradation of LCD motion blur. The considered video signal is a sequence of \( T \) frames, each of \( W \times H \) pixels. The column-stack form of the video is denoted as \( x \in \mathbb{R}^N \) where \( N = T \cdot W \cdot H \) is the total amount of pixels. The signal \( x \) is comprised from a concatenation of the \( T \) frames, i.e.,

\[
x = \begin{bmatrix} x_1 \\ \vdots \\ x_T \end{bmatrix}
\]

(3.19)

where \( x_i \in \mathbb{R}^{N_f} \) is the column-stack form of the \( i \)th frame, and \( N_f = W \cdot H \) is the number of pixels in a frame.

The degradation considered here is modeled as independent spatial operations on
each of the frames. Accordingly, the full signal degradation operator is the following block-diagonal matrix:

\[
H = \begin{bmatrix}
H_1 & 0 & \cdots & 0 \\
0 & H_2 & 0 & \vdots \\
\vdots & 0 & \ddots & 0 \\
0 & \cdots & 0 & H_T
\end{bmatrix}
\] (3.20)

where the \( N_f \times N_f \) matrix \( H_i \) is the spatial blur operator of the \( i^{th} \) frame.

Then, the block-diagonal structure (3.20) lets us to decompose the optimization (3.11), which is an intermediate stage in the ADMM iteration, into the following frame-level optimizations:

\[
\hat{z}_i^{(t)} = \arg\min_{z_i \in \mathbb{R}^{N_f}} \| x_i - H_i z_i \|_2^2 + \frac{\beta}{2} \| z_i - \tilde{v}_i^{(t)} \|_2^2, i = 1, ..., T
\] (3.21)

where \( \hat{z}_i^{(t)} \) and \( \tilde{v}_i^{(t)} \) are the column-vector forms of the \( i^{th} \) frames of the video signals \( \hat{z}^{(t)} \) and \( \tilde{v}^{(t)} \), respectively. The analytic solution of the \( i^{th} \) frame optimization from (3.21) is

\[
\hat{z}_i^{(t)} = \left( H_i^T H_i + \frac{\beta}{2} I \right)^{-1} \left( H_i^T x_i + \frac{\beta}{2} \tilde{v}_i^{(t)} \right).
\] (3.22)

This computationally important update of Algorithm 3.1 is employed in Algorithm 3.2 describing the video compression method compensating a post-decompression LCD motion blur.

**Algorithm 3.2** Proposed Method: Video Coding Adjusted to Compensate LCD Motion Blur

1: Inputs: \( x, \theta, \beta \).
2: Set frame blur operators \( \{ H_i \}_{i=1}^T \) based on motion estimation.
3: Initialize \( \hat{z}^{(0)} = x, u^{(1)} = 0, t = 1 \).
4: repeat
5: \( \tilde{z}^{(t)} = \tilde{z}^{(t-1)} - u^{(t)} \)
6: \( \tilde{v}^{(t)} = \text{CompressDecompress}_\theta (\tilde{z}^{(t)}) \).
7: \( \hat{v}^{(t)} = \tilde{v}^{(t)} + u^{(t)} \)
8: Form \( \hat{z}^{(t)} \) via frame-level solutions for \( i = 1, ..., T \):
9: \( \hat{z}_i^{(t)} = \left( H_i^T H_i + \frac{\beta}{2} I \right)^{-1} \left( H_i^T x_i + \frac{\beta}{2} \tilde{v}_i^{(t)} \right) \)
10: \( u^{(t+1)} = u^{(t)} + (\hat{v}^{(t)} - \hat{z}^{(t)}) \)
11: \( t \leftarrow t + 1 \)
12: until stopping criterion is satisfied
13: Output: Binary compressed data obtained in the last application of Stage 6.
3.4.3 Experimental Results

We evaluated our method by adjusting the HEVC video coding standard to compensate the perceptual motion-blur caused by LCD devices. As in previous works for LCD motion-blur reduction [33, 34] we considered the ‘Shields’ and ‘Stockholm’ sequences (60 frames per second) [60], having global horizontal camera motions of -3 pixels/frame and 2 pixels/frame, respectively. The considered video segments were defined as 120 frames of 480×480 pixels taken from the 720p sequences mentioned above.

In the experiments of this section we use the HEVC implementation given in the reference software HM 15.0 [61] set to the ‘random access’ profile, where powerful motion-compensation procedures together with P and B frame types are employed. The presented comparison (Figs. 3.7-3.8) consider HEVC compression operated for QP values between 1 and 19 in jumps of 3. The performance evaluations in Figs. 3.7a,3.8a rely on the average PSNR of the frames. The comparisons in Figs. 3.7b,3.8b consider the value of the SSIM metric [62] averaged over all the frames considered. The basic reference performance is the regular compression where no pre or post processing is done and, consequently, the respective performance is inferior to the other processing-aided methods (see black curves in Figs. 3.7-3.8).

The main competing approach is to precede the compression with a video deblurring method addressing the motion blur using spatio-temporal total-variation (TV) regularization [63] (note that the deblurring technique in [63] extends and improves upon the LCD motion-blur reduction method in [34]). Importantly, the methods in [34, 63] consider the video motion deblurring without any aspect of compression, thus, we employ them in the problem settings considered here as a pre-compression stage. Accordingly, we optimized the parameters of [63] to provide a high average frame-PSNR in our settings (see red solid-line performance curves in Figs. 3.7-3.8). In addition, high PSNR is not necessarily coupled with high visual quality, as the perceived video may look noisy and/or flickery (see, e.g., [34]). Therefore, we also define an additional parameter setting of [63] to provide more visually pleasing results at the expense of the PSNR (the performance curves of this smoothness-oriented setting appear as red dashed-line curves in Figs. 3.7-3.8. Moreover, we noticed that the total-variation deblurring method produces artifacts along the vertical borders of the frames (the horizontal borders are artifact free because the global motion is horizontal). Accordingly, we gave the total-variation deblurring method a larger portion of the frame with margins of 100 pixels in each side, then, these margins are removed before given to the compression. This procedure was carried out only for the total-variation deblurring approach.

We evaluated our method in two modes: the first aims to a high PSNR by the setting $\beta = 10\tilde{\beta}$ where $\tilde{\beta}$ is defined by the QP-dependent rules in (3.16). The second version, referred to as smoothness-oriented, is determined by setting $\beta = 50\tilde{\beta}$ that leads to an increased spatio-temporal smoothness (a visually pleasing property) at the expense of the PSNR. Both of these settings employed a maximal number of 10
iterations, or stopped earlier if convergence or divergence are detected. The rules defining convergence/divergence are as in the image experiment presented in Section 3.3.2, but with the different threshold values of $0.5 \cdot T$ and $50/T$ for convergence and divergence, respectively (also note the dependency on the number of frames $T$). The performance curves of the PSNR-oriented and the smoothness-oriented modes of our method appear in Figs. 3.7-3.8 as blue solid-lines and blue dashed-lines, respectively.

Figures 3.7-3.8 show that our method greatly outperforms the regular compression procedure: a BD-PSNR gain of 13.90 dB was achieved for the 'Shields' sequence (Fig. 3.7a), and a gain of 13.28 dB was obtained for the 'Stockholm' sequence (Fig. 3.8a). Figures 3.7b,3.8b present also significant gains in SSIM terms.

Let us examine the performance of our approach with respect to the total-variation deblurring technique, both in their PSNR-oriented settings (the blue and red solid-line curves in Figs. 3.7-3.8). Considering the 'Shields' (Fig. 3.7a) and 'Stockholm' (Fig. 3.8a) sequences, our method achieved respective BD-PSNR gains of 1.06 dB and 2.16 dB over the total-variation deblurring technique. Figures 3.7b,3.8b exhibit that our PSNR-oriented method is better than the TV deblurring approach also with respect to the SSIM quality metric.

The third comparison considers our method with respect to the total-variation deblurring approach, both in their smoothness-oriented settings (the blue and red dashed-line curves in Figs. 3.7-3.8). Considering the 'Shields' (Fig. 3.7a) and 'Stockholm' (Fig. 3.8a) sequences, our method differs from the TV-deblurring technique in BD-PSNR values of $-0.62$ dB and $0.98$ dB. One should recall that this is the smoothness-oriented settings, thus, visual quality is preferred over optimizing the PSNR. Indeed, examining the SSIM-bitrate curves of the 'Shields' (Fig. 3.7b) and 'Stockholm' (Fig. 3.8b) sequences, exhibit that our smoothness-oriented method obtains the respective average SSIM gains of $3.4 \times 10^{-3}$ and $5.8 \times 10^{-3}$ over the smoothness-oriented TV deblurring technique. These results point on the good visual quality offered by the smoothness-oriented settings of our method.

In Figure 3.9 we provide a visual demonstration of the results obtained for a frame segment from the sequence 'Shields' (Fig. 3.9a). As the various methods do not produce equal bit-rates, the comparison is for relatively close bit-rates – specifically, the results presented for our method obtained using lower bit-rates than the other techniques. Figures 3.9b-3.9f show the frames given to the display using the various methods and their settings. Our method (Fig. 3.9e-3.9f) as well as the other deblurring-based approach (Fig. 3.9c-3.9d) provide sharpened images to display. Figures 3.9g-3.9k exhibit the simulated perceived image (i.e., the displayed frame after the blur degradation modeled above). Evidently, the perceived images corresponding to the pre-compensating techniques highly resemble the original frame. Moreover, Fig. 3.10 shows for each of the methods the difference between the simulated-perceived and the original images, suggesting that our PSNR-oriented method avoids noticeable image-detail loss, at the expense of some textural-noise that can be attenuated using the smoothness-oriented
Figure 3.7: LCD motion blur experiment for the 'Shields' sequence. The performance of our method in its PSNR-oriented (solid blue line) and smoothness-oriented (dashed blue line) modes is compared to preprocessing using total-variation deblurring [63] in its PSNR-oriented (solid red line) and smoothness-oriented (dashed red line) parameter settings, and to a regular compression-decompression procedure without any additional processing (solid black line). The average frame-PSNR and average frame-SSIM are evaluated in (a) and (b), respectively.

Figure 3.8: LCD motion blur experiment for the 'Stockholm' sequence. The performance of our method in its PSNR-oriented (solid blue line) and smoothness-oriented (dashed blue line) modes is compared to preprocessing using total-variation deblurring [63] in its PSNR-oriented (solid red line) and smoothness-oriented (dashed red line) parameter settings, and to a regular compression-decompression procedure without any additional processing (solid black line). The average frame-PSNR and average frame-SSIM are evaluated in (a) and (b), respectively.
settings. Importantly, the PSNR-oriented settings of our method achieve the highest PSNR and SSIM values for the presented frame, while using a lower bit-rate than the regular and the total-variation approaches.
Figure 3.9: LCD motion blur experiment for the sequence 'Shields' (grayscale, 120 frames at 60fps, 480x480 pixels) where a global horizontal motion of -3 pixels/frame causes the blur. An exemplary frame segment is presented. (a) the original frame segment. (b)-(f) are the displayed frame segments using each of the five examined methods. (g)-(k) are the simulated perceived frame segments corresponding to each of the displayed frames. The presented PSNR and SSIM evaluations are for the complete perceived frame, and the bit-rates are those measured for the compression of the complete sequence of 120 frames.
Figure 3.10: LCD motion blur experiment for the sequence 'Shields'. Here we present the difference images between the perceived and the original frame segments, for each of the methods presented in Fig. 3.9: (a) regular, (b) TV smoothness-oriented, (c) TV PSNR-oriented, (d) proposed smoothness-oriented, and (e) proposed PSNR-oriented. The difference images are presented in grayscale by scaling the value range of $[-50, 50]$ for the regular approach, and the value range of $[-15, 15]$ for the other methods.
Chapter 4

System-Aware Compression:
Optimizing Systems from the
Compression Standpoint

Here we extend the discussion of the former chapter and consider a system employing lossy compression as an intermediate stage between a pre and post degradation stages. The discussed system structure corresponds to various acquisition-rendering architectures. Based on our general compression framework we develop a methodology for optimizing the system rate-distortion performance from the compression standpoint. We present a generic operational method and explain its main ideas using rate-distortion theory of Gaussian signals. Experimental demonstrations for coding of 1D signals and video sequences are provided, showing the effectiveness of our approach for adjusting standard compression designs to specific acquisition-rendering system models.

4.1 Introduction

Lossy compression has a central role in information systems where the data may be inaccurately represented in order to meet storage-space or transmission-bandwidth constraints. While the compression is a crucial system-component, it is only an intermediate stage among data processing procedures that determine the eventual output of the system. For example, consider a common audio/visual system structure where the source signal is acquired and compressed for its storage/transmission, then the decompression is followed by a rendering stage producing the ultimate system output. Evidently, in this example, the quality of the system output is determined by the acquisition-rendering chain, and not solely on the lossy compression stage. Nevertheless, the compression is usually designed independently of the system structure, thus inducing a sub-optimal rate-distortion performance for the complete system.

Here we propose a compression approach defined by an operational rate-distortion optimization considering a known system structure. Specifically, we study a general flow
Figure 4.1: The general system structure considered in this chapter.

(Fig. 4.1) where the compression is preceded by a linear operator distorting the input along with an additive white noise, and the decompression is followed by another linear operation. We formulate a rate-distortion optimization based on a quadratic distortion metric involving the system’s linear operators. For general linear operators, this intricate rate-distortion optimization is too hard to be directly solved for high dimensional signals. Consequently, we address this challenge using the alternating direction method of multipliers (ADMM) technique \[15\], suggesting an iterative procedure that relies on the simpler tasks of standard compression (which is system independent!) and \(\ell_2\)-constrained deconvolution for the linear operators of the system.

Optimizing the system output quality from the compression standpoint is an attractive answer to the inherent tradeoff among distortion, bit-cost, and computational complexity. First, bits are wisely spent for representing signal components that will be important at the output of the overall system (for example, one should obviously not code signal components belonging to the null space of the post-decompression operator). Second, the added computational load can be well accommodated in the compression environment that is often rich in computational and time resources, in contrast to the decompression stage.

Importantly, the proposed compression framework is a paradigm for addressing intricate rate-distortion optimization forms via iterative solution of easier problems emerging from the ADMM method (or other variable-splitting optimization techniques \[46\]). Indeed, the problem addressed here is an extension of the compression method presented in Chapter 3 for pre-compensating a degradation occurring after decompression. In this chapter the task is harder than in Chapter 3, as the source signal is available only in its degraded version. The recent wide use of ADMM for complicated signal restoration problems (e.g., \[16, 17, 64, 19, 20\]) suggests that our ADMM-based approach for complicated compression problems entails great potential.

Essentially, we study here a remote source coding problem (e.g., see its origins in \[13, 14\] and their successors). Our contribution is mainly the deterministic setting
versus the statistical perspective used in previous works. Specifically, we consider an operational rate-distortion problem for the compression of a given signal, based on a different distortion metric imposed by the lack of an explicit statistical model of the unknown source. Our settings lead to the remarkable result that, using the ADMM technique, one can employ a standard compression method to address remote source coding problems in much more complicated instances than were previously feasible (e.g., [35]).

Using rate-distortion theory, we further study the examined problem in statistical settings (considering the proposed distortion metric) for the case of a cyclo-stationary Gaussian source signal and linear shift-invariant system operators. Our results show that the initial rate-distortion optimization reduces to a reverse water-filling procedure adjusted to the system operators and considering the pseudoinverse-filtered version of the input signal. We use these theoretic results to explain concepts appearing (differently) in the proposed practical method intended for non-Gaussian signals and general linear system operators.

Jointly using sampling and source coding procedures is fundamental to digitization-based systems (see, e.g., [65]). Accordingly, we demonstrate our general framework for adapting standard compression methods to the specific settings of a complete acquisition-rendering system. We present experiments considering coding of one-dimensional signals using an adaptive tree-based technique, and to video compression using the state-of-the-art HEVC standard [29]. Comparisons of our strategy to a regular compression flow exhibited that our method achieves significant gains at medium/high bit-rates.

### 4.2 The Proposed Method

Let us describe the considered system structure (Fig. 4.1). A source signal, an $N$-length column vector $x \in \mathbb{R}^N$, undergoes a linear processing represented by the $M \times N$ matrix $A$ and, then, deteriorated by an additive white Gaussian noise vector $n \sim \mathcal{N}(0, \sigma_n^2 I)$, resulting in the signal

$$w = Ax + n$$  \hspace{1cm} (4.1)

where $w$ and $n$ are $M$-length column vectors. We represent the lossy compression procedure via the mapping $C : \mathbb{R}^M \rightarrow B$ from the $M$-dimensional signal domain to a discrete set $B$ of binary compressed representations (that may have different lengths). The signal $w$ is the input to the compression component of the system, producing the compressed binary data $b = C(w)$ that can be stored or transmitted in an error-free manner. Then, on a device and settings depending on the specific application, the compressed data $b \in B$ is decompressed to provide the signal $v = F(b)$ where $F : B \rightarrow S$ represents the decompression mapping between the binary compressed representations in $B$ to the corresponding decompressed signals in the discrete set $S \subset \mathbb{R}^M$. The
decompressed signal \( v \) is further processed by the linear operator denoted as the \( N \times M \) matrix \( B \), resulting in the system output signal
\[
y = Bv,
\] (4.2)
which is an \( N \)-length real-valued column vector.

As an example, consider an acquisition-compression-rendering system where the signal \( w \) is a sampled version of the source signal \( x \), and the system output \( y \) is the rendered version of the decompressed signal \( v \).

We assume here that the operators \( A \) and \( B \), as well as the noise variance \( \sigma^2_n \), are known and fixed (i.e., cannot be optimized). Consequently, we formulate a new compression procedure in order to optimize the end-to-end rate-distortion performance of the entire system. Specifically, we want the system output \( y \) to be the best approximation of the source signal \( x \) under the bit-budget constraint. However, at the compression stage we do not accurately know \( x \), but rather its degraded form \( w \) formulated in (4.1). This motivates us to suggest the following distortion metric with respect to the system output \( y \)
\[
D_s(w, y) = \frac{1}{M} \| w - Ay \|_2^2.
\] (4.3)
This metric conforms with the fact that if \( y \) is close to \( x \), then, by (4.1), \( w \) will be close to \( Ay \) up to the noise \( n \). Indeed, for the ideal case of \( y = x \) the metric (4.3) becomes
\[
D_s(w, x) = \frac{1}{M} \| n \|_2^2 \approx \sigma^2_n
\] (4.4)
where the last approximate equality is under the assumption of a sufficiently large \( M \) (the length of \( n \)). Since \( y = Bv \), we can rewrite the distortion \( D_s(w, y) \) in (4.3) as a function of the decompressed signal \( v \), namely,
\[
D_c(w, v) = \frac{1}{M} \| w - ABv \|_2^2.
\] (4.5)
Since the operator \( B \) produces the output signal \( y \), an ideal result will be \( y = P_B x \), where \( P_B \) is the matrix projecting onto \( B \)'s range. The corresponding ideal distortion is
\[
d_0 \triangleq D_s(w, P_B x) = \frac{1}{M} \| A(I - P_B) x + n \|_2^2.
\] (4.6)

We use the distortion metric (4.5) to constrain the bit-cost minimization in the following rate-distortion optimization
\[
\hat{v} = \arg\min_{v \in S} R(v) \text{ subject to } d_0 \leq \frac{1}{M} \| w - ABv \|_2^2 \leq d_0 + d,
\] (4.7)
where $R(v)$ evaluates the length of the binary compressed description of the decompressed signal $v$, and $d \geq 0$ determines the allowed distortion. By (4.6), the value $d_0$ depends on the operator $A$, the null space of $B$, the source signal $x$, and the noise realization $n$. Since $x$ and $n$ are unknown, $d_0$ cannot be accurately calculated in the operational case (in Section 4.3 we formulate the expected value of $d_0$ for the case of a cyclo-stationary Gaussian source signal). We address the optimization (4.7) using its unconstrained Lagrangian form

$$\hat{v} = \arg\min_{v \in S} R(v) + \lambda \frac{1}{M} \| w - ABv \|^2 \quad (4.8)$$

where $\lambda \geq 0$ is a Lagrange multiplier corresponding to some distortion constraint $d_\lambda \geq d_0$ (such optimization strategy with respect to some Lagrange multiplier is common, e.g., in video coding [29]). In the case of high-dimensional signals, the discrete set $S$ is extremely large and, therefore, it is impractical to directly solve the Lagrangian form in (4.8) for generally structured matrices $A$ and $B$. This difficulty vanishes, for example, when $A = B = I$, reducing the Lagrangian optimization in (4.8) to the standard (system independent) compression form (see, e.g., [24, 26]). Indeed, such standard Lagrangian rate-distortion optimizations are practically solved using block-based designs that translate the task to a sequence of block-level optimizations of feasible dimensions.

Here we consider general $A$ and $B$ matrices, and address the computational difficulty in solving (4.8) using the alternating direction method of multipliers (ADMM) technique [15]. For start, we apply variable splitting to rewrite (4.8) as

$$\hat{v} = \arg\min_{v \in S, z \in \mathbb{R}^M} R(v) + \lambda \frac{1}{M} \| w - ABz \|^2$$

subject to $v = z \quad (4.9)$

where $z \in \mathbb{R}^M$ is an auxiliary variable that is not (directly) restricted to the discrete set $S$. The augmented Lagrangian (in its scaled form) and the method of multipliers (see [15, Ch. 2]) turn (4.9) into the following iterative procedure

$$\begin{align*}
(\hat{v}^{(t)}, \hat{z}^{(t)}) &= \arg\min_{v \in S, z \in \mathbb{R}^M} R(v) + \lambda \frac{1}{M} \| w - ABz \|^2 + \beta \left( \frac{1}{2} \left\| v - z + u^{(t)} \right\|^2 \right) \\
u^{(t+1)} &= u^{(t)} + \left( \hat{v}^{(t)} - \hat{z}^{(t)} \right) \quad (4.11)
\end{align*}$$

where $t$ is the iteration number, $u^{(t)} \in \mathbb{R}^M$ is the scaled dual variable, and $\beta$ is an auxiliary parameter originating at the augmented Lagrangian. Further simplifying (4.10)
using one iteration of alternating minimization gives the ADMM form of the problem

\[
\hat{v}^{(t)} = \arg\min_{v \in S} R(v) + \frac{\beta}{2} \|v - \tilde{z}^{(t)}\|_2^2 \tag{4.12}
\]

\[
\hat{z}^{(t)} = \arg\min_{z \in \mathbb{R}^M} \lambda \frac{1}{M} \|w - ABz\|_2^2 + \frac{\beta}{2} \|z - \hat{v}^{(t)}\|_2^2 \tag{4.13}
\]

\[
u^{(t+1)} = u^{(t)} + (\hat{v}^{(t)} - \hat{z}^{(t)}) \tag{4.14}
\]

where \( \tilde{z}^{(t)} = \hat{z}^{(t-1)} - u^{(t)} \) and \( \hat{v}^{(t)} = \hat{v}^{(t)} + u^{(t)} \). Importantly, the compression design, expressed by \( \{S, R\} \), and the system-specific operators \( \{A, B\} \) were decoupled by the ADMM to reside in distinct optimization problems that are easier to solve.

We identify the first stage (4.12) as the Lagrangian optimization employed for standard compression (and decompression) tasks considering the regular mean squared error metric. Specifically, the Lagrange multiplier for this standard optimization is \( \tilde{\lambda} = \frac{\beta M}{2} \). Furthermore, we suggest to replace the solution of (4.12) with the application of a standard compression (and decompression) technique, even one that does not follow the Lagrangian optimization formulated in (4.12). We denote the standard compression and decompression via

\[
b^{(t)} = \text{StandardCompress} \left( \hat{z}^{(t)}, \theta \right) \tag{4.15}
\]

\[
\hat{v}^{(t)} = \text{StandardDecompress} \left( b^{(t)} \right) \tag{4.16}
\]

where \( \theta \) is a general parameter that extends the Lagrange multiplier role in determining the rate-distortion tradeoff (see Algorithm 4.1). This important suggestion defines our method as a generic approach that can optimize any compression technique with respect to the specific system it resides in.

The second optimization stage (4.13) is an \( \ell_2 \)-constrained deconvolution problem, that can be easily solved in various ways. The analytic solution of (4.13) is

\[
\hat{z}^{(t)} = \left( B^*A^*AB + \frac{\beta M}{2\lambda} I \right)^{-1} \left( B^*A^*w + \frac{\beta M}{2\lambda} \hat{v}^{(t)} \right), \tag{4.17}
\]

showing it as a weighted averaging of \( w \) and \( \hat{v}^{(t)} \). In the generic description given in Algorithm 4.1 we replace the quantity \( \frac{\beta M}{2\lambda} \) with the parameter \( \tilde{\beta} \).

The proposed method is summarized in Algorithm 4.1. Our goal is to provide a binary compressed representation of the optimized solution. Hence, the procedure output is the compressed data, \( b^{(t)} \), obtained in the compression stage of the last iteration.
Algorithm 4.1 Generic System-Aware Compression

1: Inputs: $w, \theta, \tilde{\beta}$.
2: Initialize $t = 0$, $\hat{z}^{(0)} = w$, $u^{(1)} = 0$.
3: repeat
4: $t \leftarrow t + 1$
5: $\tilde{z}^{(t)} = \tilde{z}^{(t-1)} - u^{(t)}$
6: $b^{(t)} = \text{StandardCompress} (\tilde{z}^{(t)}, \theta)$
7: $\hat{v}^{(t)} = \text{StandardDecompress} (b^{(t)})$
8: $\hat{v}^{(t)} = \hat{v}^{(t)} + u^{(t)}$
9: $\hat{z}^{(t)} = \left( B^* A^* A B + \tilde{\beta} I \right)^{-1} \left( B^* A^* w + \tilde{\beta} \hat{v}^{(t)} \right)$
10: $u^{(t+1)} = u^{(t)} + (\hat{v}^{(t)} - \hat{z}^{(t)})$
11: until stopping criterion is satisfied
12: Output: $b^{(t)}$, which is the binary compressed data obtained in the last iteration.

4.3 Theoretic Analysis for the Gaussian Case

In this section we use the rate-distortion theory framework to further explore the fundamental problem of system-optimized compression. We consider the case of a cyclo-stationary Gaussian signal and system involving linear shift-invariant operators and no noise, yet, the obtained results exhibit the prominent ideas of the general problem and the operational method presented in Section 4.2.

4.3.1 Problem Formulation and Solution

The source signal is modeled here as $x \sim \mathcal{N}(0, R_x)$, i.e., a zero-mean Gaussian random vector with a circulant autocorrelation matrix $R_x$. The eigenvalues of $R_x$ are denoted as $\{ \lambda_k(x) \}_{k=0}^{N-1}$. The first processing part of the system produces the signal $w = Ax$, where here $A$ is a real-valued $N \times N$ circulant matrix. Evidently, the signal $w$ is a zero-mean Gaussian random vector with autocorrelation matrix $R_w = AR_xA^*$.

Here $A$ and $B$ are circulant $N \times N$ matrices, thus, diagonalized by the $N \times N$ Discrete Fourier Transform (DFT) matrix $F$ as $FAF^* = \Lambda_A$ and $FBF^* = \Lambda_B$, where $\Lambda_A$ and $\Lambda_B$ are diagonal matrices formed by the elements $\{ a_F^k \}_{k=0}^{N-1}$ and $\{ b_F^k \}_{k=0}^{N-1}$, respectively. Accordingly, the pseudoinverse matrix of $A$ is defined as $A^+ = F^* \Lambda_A^+ F$, where $\Lambda_A^+$ is the pseudoinverse of $\Lambda_A$, an $N \times N$ diagonal matrix with the $k$th diagonal component $a_F^k$ for $k$ where $a_F^k \neq 0$, and $a_F^k = 0$ for $k$ where $a_F^k = 0$. The matrix $B$ has corresponding definitions to those given above for $A$.

Recall that the system output is $y = Bv$, where here $v$ is the random vector representing the decompressed signal. Accordingly, in the theoretic framework here, the rate is associated with the mutual information $I(w, y)$. In (4.7) we formulated the operational rate-distortion optimization describing our practical task, cast here in the
theoretic framework to
\[
\min_{p_{v|w}} I(w; Bv)
\]
subject to
\[
NE\{d_0\} \leq E\{\|w - ABv\|^2\} \leq N\{E\{d_0\} + d\}
\]
where \(p_{v|w}\) is the conditional probability-density-function of \(v\) given \(w\), and \(d \geq 0\) determines the allowed expected distortion. The value \(E\{d_0\}\), stemming from (4.6) and the noiseless settings here, is the minimal expected distortion evaluated (see Section 4.5.1) as
\[
E\{d_0\} = \frac{1}{N} \sum_{k: a_k^F \neq 0, b_k^F = 0} |a_k^F|^2 \lambda_k^{(w)}.
\]

We state that the basic rate-distortion optimization problem in (4.18) is equivalent to (see proof in Section 4.5.2)
\[
\min_{p_{v|\tilde{w}}} I(\tilde{w}; P_B P_A v)
\]
subject to
\[
E\{\|AB(\tilde{w} - v)\|^2\} \leq Nd
\]
where \(\tilde{w} = B^+ A^+ w\) is the pseudoinverse filtered compression-input \(w\). Here \(P_A\) and \(P_B\) are projection matrices corresponding to the range of \(A\) and \(B\), respectively. We define the set \(K_{AB} \triangleq \{k: a_k^F \neq 0 \text{ and } b_k^F \neq 0\}\) that contains the DFT-domain component indices belonging to the range of the joint operator \(AB\).

Since \(\tilde{w}\) is a cyclo-stationary Gaussian random vector, we transform the optimization (4.20) into a Fourier-domain distortion-allocation problem considering independent Gaussian variables (see proof in Section 4.5.3), namely,
\[
\min_{\{d_k\}_{k \in K_{AB}}} \sum_{k \in K_{AB}} \frac{1}{2} \log \left(\frac{\lambda_k^{(\tilde{w})}}{d_k}\right)
\]
subject to
\[
\sum_{k \in K_{AB}} |a_k^F b_k^F|^2 d_k \leq Nd
\]
\[
0 \leq d_k \leq \lambda_k^{(\tilde{w})}, \quad k \in K_{AB}.
\]

where for \(k \in K_{AB}\) the value \(\lambda_k^{(\tilde{w})} = \lambda_k^{(w)}/|a_k^F b_k^F|^2\) is the variance of the \(k^\text{th}\) DFT-coefficient of \(\tilde{w}\). The solution of problem (4.21), obtained using Lagrangian optimization and the KKT conditions, is given by the following distortion allocation: for \(k \in K_{AB}\)
\[
d_k = \begin{cases} 
\theta/|a_k^F b_k^F|^2, & \text{for } 0 \leq \theta < |a_k^F b_k^F|^2 \lambda_k^{(\tilde{w})} \\
\lambda_k^{(\tilde{w})}, & \text{for } \theta \geq |a_k^F b_k^F|^2 \lambda_k^{(\tilde{w})},
\end{cases}
\]
where \(\theta\) is set such that \(\sum_{k \in K_{AB}} |a_k^F b_k^F|^2 d_k = Nd\) is satisfied. Importantly, for \(k \notin K_{AB}\)
the rate is $R_k = 0$ and the distortion is set to $d_k = 0$ (see Section 4.5.3). The optimal rates corresponding to (4.22) assign for $k \in K_{AB}$ that also obey $0 \leq \theta < \left| a_k^F b_k^F \right|^2 \lambda_k^{(w)}$

$$R_k = \frac{1}{2} \log \left( \left| a_k^F b_k^F \right|^2 \frac{\lambda_k^{(w)}}{\theta} \right)$$

(4.23)

and otherwise $R_k = 0$. Eq. (4.23) shows that the optimal rate assignments include compensation for the modulation applied beforehand in the pseudoinverse filtering of the input $w$. Moreover, DFT components belonging to the null spaces of $A$ and $B$ are not coded (i.e., get zero rates).

4.3.2 Problem Solution in Theory and Practice

While the theoretic framework above considers a cyclo-stationary Gaussian signal and a noiseless system composed of linear shift-invariant operators, the solution presented exhibits important ideas that also appear in the practical method of Section 4.2. Recall that our method is designed for the operational settings of the problem treating non-Gaussian signals and general linear operators, hence, the resemblances between the above theory and practice are at the conceptual level and may materialize differently.

We addressed the theoretic problem (4.18) using a simple inverse filtering of the input data $w$, transforming the problem into (4.20) and (4.21) that were solved using an extended version of the standard reverse water-filling procedure. Analogously, our practical method (Algorithm 4.1) repeatedly compresses a signal formed by an $\ell_2$-constrained deconvolution filtering, that can be rewritten also as a pseudoinverse filtering of the input followed by a weighted averaging with $\hat{v}^{(t)}$, i.e.,

$$\hat{z}^{(t)} = \left( B^* A^* A B + \tilde{\beta} I \right)^{-1} \left( B^* A^* A B \tilde{w} + \tilde{\beta} \hat{v}^{(t)} \right).$$

(4.24)

This shows that, in practice as well as in theory, the input $w$ should go through a pseudoinverse filtering (softened via (4.24)) as a preceding stage to compression.

The second prominent principle of the theoretic solution, exhibited in (4.23), is to compensate for the modulation applied by the pseudoinverse filter corresponding to the effective system operator $A B$. Similarly in Algorithm 4.1, the constrained deconvolution stage (4.24) implements this idea by better preserving $\tilde{w}$ components corresponding to higher energy parts of $A B$. This is clearly observed in the particular case of circulant $A$ and $B$, where the filtering (4.24) reduces to the DFT-domain component-level operation of

$$z_k^{F(t)} = \frac{\left| a_k^F b_k^F \right|^2 \tilde{w}_k^F + \tilde{\beta} \hat{v}_k^{F(t)}}{\left| a_k^F b_k^F \right|^2 + \tilde{\beta}}$$

(4.25)

where $\tilde{w}_k^F$ and $\hat{v}_k^{F(t)}$ are the $k^{th}$ DFT-coefficients of $\tilde{w}$ and $\hat{v}^{(t)}$, respectively.
4.4 Experimental Results

In this section we employ Algorithm 4.1 to adjust standard compression designs to acquisition-rendering systems where the compression is an intermediate stage. Specifically, we model the source as a high-resolution discrete signal, acquired via linear shift-invariant low-pass filtering and uniform sub-sampling. Then, this acquired signal is available for compression, and after decompression it is linearly rendered back to the source resolution by replicating each decompressed sample (in a uniform pattern matching the source sub-sampling).

Coding of One Dimensional Signals

The compression approach here relies on adaptive tree-based segmentation of the signal (for another instance of this prevalent idea see [28]). Specifically, here we compress a one-dimensional signal using an operational rate-distortion optimization determining a binary-tree corresponding to a non-uniform segmentation of the signal, where each segment is represented by a constant value defined by the quantized average-value of the interval (see more details in Appendix A). We consider the system flow of acquisition-compression-decompression-rendering for an amplitude-modulated chirp source signal, as will be explained next.

The considered source signal $x$ is an amplitude-modulated chirp (see the blue curves in Figs. 4.2a-4.2b), defined by sampling its mathematical formulation using 1024 samples uniformly spread in the "time" interval $[0, 1)$. Note that the chirp signal values are in the range $[0, 1]$, a property used in the PSNR computation shown in the result evaluation in Fig. 4.2c.

The acquisition is modeled here by a low-pass filtering applied using a convolution with a Gaussian kernel (standard deviation 15) and support of 15 samples, followed by sub-sampling in a factor of 4, and an additive white Gaussian noise with standard deviation 0.001. This procedure results with the 256-samples signal $w$ that is given to the compression. After decompression the rendering operator is applied by replicating each sample of $v$ four times such that the piecewise-constant signal $y$ is formed, having a size of 1024 samples.

We tested two compression strategies to employ in the acquisition-rendering system. The first is a regular compression using the tree-based procedure, described in the former subsection, applied based on some Lagrange multiplier $\nu$. The experiment for this regular flow was repeated for various values of the Lagrange multiplier $\nu$ to produce the PSNR-bitrate curve in Fig. 4.2c. The second approach relied on the implementation of the proposed method (Algorithm 4.1) as the compression stage that leverages the standard tree-based coding described in the former subsection. This approach was evaluated for a range of $\nu$ values to obtain the corresponding PSNR-bitrate curve in Fig. 4.2c. The implementation of our iterative method (Algorithm 4.1) run a maximum of 40 iterations or until convergence is detected.
Figures 4.2a and 4.2b exhibit the system output (i.e., the rendered signal) resulting from the use of a regular compression at 4.71 bpp and from employing our method at 3.69 bpp, respectively. Evidently, our method outperforms the regular approach in terms of PSNR and also in reproducing the chirp signal peaks. Comparing our method to the regular approach at various bit-rates (Fig. 4.2c) shows significant PSNR gains at medium/high bit-rates.

Figure 4.2: Experiment for one-dimensional signal and adaptive tree-based coding in an acquisition-rendering system. The amplitude-modulated chirp source signal is the blue curve in (a) and (b). The red curves in (a) and (b) are the system output (rendered) signals resulting from regular compression and the proposed method, respectively. (c) Comparison of PSNR-bitrate curves.

**Video Coding**

We evaluated our method also for adjusting the HEVC coding standard [29] to a simplified acquisition-rendering system, considering the spatial dimensions of the frames. The source signal $x$ is a sequence of 10 frames, each of $480 \times 480$ pixels. The acquisition is
modeled here by a low-pass filtering carried out by a convolution with a two-dimensional Gaussian kernel (standard deviation 1) and support of $5 \times 5$ pixels, followed by horizontal and vertical sub-sampling in a factor of 2, and an additive white Gaussian noise with standard deviation 0.001. This procedure results with the 10-frame sequence $w$ with a frame size of $240 \times 240$ pixels. The rendering applied after decompression is simply done by replicating each pixel of $v$ in a $2 \times 2$ pixels square such that the rendered signal $y$ has frames of $480 \times 480$ pixels having spatial piecewise-constant form.

We evaluated two compression approaches to apply in the acquisition-rendering system. One, is to employ a regular compression using the HEVC video coding standard (using its reference software, version HM 15.0, available at http://hevc.hhi.fraunhofer.de/). The experiment for this regular approach was repeated for various values of the quality parameter of the HEVC to produce the PSNR-bitrate curve in Fig. 4.3. The second strategy implemented the proposed method (Algorithm 4.1) as the compression procedure that utilizes the HEVC standard. This approach was evaluated for a range of HEVC quality parameters to provide the corresponding PSNR-bitrate curve in Fig. 4.3. The implementation of our iterative method (Algorithm 4.1) run a maximum of 10 iterations or until convergence is detected.

In Fig. 4.3 we presented the PSNR-bitrate curves for the compression of segments of two video signals: 'Stockholm' and 'Shields'. Here, in Figures 4.4 and 4.5, we show the third frame from the sequence, in its original form and in its rendered form using the regular approach and via the proposed method. Clearly, our method provides a more vivid image (also having higher PSNR) at a lower bit-rate. The PSNR-bitrate curves in Fig. 4.3 and the visual results show the capability of our approach for generically adapting a complicated compression method to a given system structure.
Figure 4.4: Coding a group of 10 frame from the 'Stockholm' sequence (spatial portion of 480x480 pixels). (a) The third source frame. (b) the rendered frame using regular compression (28.29 dB at 2.37 bpp). (c) the rendered frame using the proposed compression (29.45 dB at 1.34 bpp).
Figure 4.5: Coding a group of 10 frame from the 'Shields' sequence (spatial portion of 480x480 pixels). (a) The third source frame. (b) the rendered frame using regular compression (27.93 dB at 2.41 bpp). (c) the rendered frame using the proposed compression (29.31 dB at 1.31 bpp).
4.5 Proofs for the Rate-Distortion Theoretic Analysis

4.5.1 Minimal Expected Distortion

The minimal distortion was defined in (4.6) for the operational settings. Here we will formulate its expected value, needed for the statistical settings of Section 4.3. Expecting (4.6), considering

\[ M = N \]

and no noise, gives

\[
E \{ d_0 \} = \frac{1}{N} E \left\{ \| A (I - P_B) x \|_2^2 \right\} \\
= \frac{1}{N} E \left\{ \| F^* A F F^* (I - \Lambda_{P_B}) F x \|_2^2 \right\} \\
= \frac{1}{N} E \left\{ \| \Lambda_A (I - \Lambda_{P_B}) x^F \|_2^2 \right\} \\
= \frac{1}{N} E \left\{ \sum_{k : a_k^F \neq 0, b_k^F = 0} |a_k^F|^2 \lambda_k^{(x)} \right\}
\]

where \( x^F \triangleq Fx \). Furthermore, we used the circulant structure of \( P_B \) (as it is the projection matrix corresponding to the range of the circulant matrix \( B \)) providing the DFT-based diagonalization via \( FP_B F^* = \Lambda_{P_B} \). Moreover, we utilized the formation of the diagonal matrix \( \Lambda_{P_B} \) where the \((k,k)\) entry is 1 for \( k \) corresponding to \( b_k^F \neq 0 \), and otherwise the \((k,k)\) entry is 0.

\[
(4.26)
\]

4.5.2 Equivalence of Optimization Problems (4.18) and (4.20)

Let us begin by proving the equivalence between the distortion constraints of optimizations (4.18) and (4.20). We define the matrix \( H \triangleq AB \) joining the two linear operators of the system. In Section 4.3 we considered \( A \) and \( B \) to be \( N \times N \) circulant matrices and, thus, \( H \) is also a \( N \times N \) circulant matrix. Consequently, \( H \) is diagonalized by the \( N \times N \) DFT matrix \( F \) via \( FHF^* = \Lambda_H \), where \( \Lambda_H \) is a diagonal matrix formed by the elements \( h_k^F = a_k^F b_k^F \) for \( k = 0, ..., N - 1 \). The pseudoinverse matrix of \( H \) is defined as \( H^+ = F^* \Lambda_H^+ F \), where \( \Lambda_H^+ \) is the pseudoinverse of \( \Lambda_H \), an \( N \times N \) diagonal matrix with the \( k^{th} \) diagonal component

\[
h_k^{F,+} = \begin{cases} 
\frac{1}{a_k^F b_k^F} & \text{, for } k \in \mathcal{K}_{AB} \\
0 & \text{, for } k \notin \mathcal{K}_{AB} 
\end{cases}
\]

where \( \mathcal{K}_{AB} \triangleq \{ k : a_k^F \neq 0 \text{ and } b_k^F \neq 0 \} \) includes the DFT-domain component indices defining the range of the joint operator \( AB \).

Now, let us develop the distortion expression appearing in the constraint of opti-
\[ \| w - ABv \|_2^2 = \| w - Hv \|_2^2 \]
\[ = \|(I - HH^+)w + HH^+w - Hv\|_2^2 \]
\[ = \|(I - HH^+)w + H(H^+w - v)\|_2^2 \]
\[ = \|(I - HH^+)w\|_2^2 + \|H(H^+w - v)\|_2^2 \]
\[ + (H^+w - v)^*H^+(I - HH^+)w \]
\[ + w^*(I - HH^+)^*H(H^+w - v) \]
\[ = \|(I - HH^+)w\|_2^2 + \|H(H^+w - v)\|_2^2 \]

(4.28)

where the last equality readily stems from the relation
\[ H^*(I - HH^+) = 0. \]  

(4.29)

We use the DFT-based diagonalization of \( H \) to continue developing the expression of the first term in (4.28):
\[ \|(I - HH^+)w\|_2^2 = \|(I - F^*A_HFF^*A_H^+F)w\|_2^2 \]
\[ = \|F^*(I - A_HA_H^+)Fw\|_2^2 \]
\[ = \|(I - A_HA_H^+)w^F\|_2^2 \]
\[ = \sum_{k:b_k^F=0} \|w_k^F\|^2 = \]
\[ \sum_{k:a_k^F\neq0,b_k^F=0} |a_k^F|^2 \]

(4.30)

where we used the DFT-domain expression of the \( k^{th} \) component of \( w^F \) as \( w_k^F = a_k^Fx_k \), reducing to \( w_k^F = 0 \) when \( a_k^F = 0 \). Taking the expectation of (4.30), noting that the source signal \( x \) and the noise \( n \) are independent, yields
\[ E\left\{ \|(I - HH^+)w\|_2^2 \right\} = \sum_{k:a_k^F\neq0,b_k^F=0} |a_k^F|^2 \lambda_k^{(x)}. \]  

(4.31)

Then, using (4.28) and (4.31) we get that
\[ E\left\{ \|w - ABv\|_2^2 \right\} = E\left\{ \|AB(\tilde{w} - v)\|_2^2 \right\} \]
\[ + \sum_{k:a_k^F\neq0,b_k^F=0} |a_k^F|^2 \lambda_k^{(x)}, \]  

(4.32)
where \( \tilde{w} = B^+ A^+ w \). Eq. (4.26) implies

\[
NE \{d_0\} = \sum_{k: ak_k \neq 0, b_k = 0} \left| a_k F \right|^2 \lambda_k^{(x)}
\]

(4.33)

that jointly with (4.32) yields the transformation of the expected distortion constraint in the optimization (4.18), namely,

\[
NE \{d_0\} \leq E \left\{ \| w - ABv \|_2^2 \right\} \leq N \left( E \{d_0\} + d \right)
\]

(4.34)

into

\[
0 \leq E \left\{ \| AB (\tilde{w} - v) \|_2^2 \right\} \leq Nd.
\]

(4.35)

Since (4.35) is the distortion constraint in optimization (4.20), we proved that the distortion constraints of optimizations (4.18) and (4.20) are interchangeable.

We continue by showing the equivalence of the mutual information terms appearing in the optimization costs of (4.18) and (4.20). We start with \( I(w; Bv) \) used in (4.18). By the data processing inequality, the compression-input \( w \) and its filtered version \( A^+ w \) obey

\[
I(w; Bv) \geq I(A^+ w; Bv)
\]

(4.36)

In the theoretic settings we consider the noiseless case where \( w = Ax \), hence, \( A^+ w = P_A x \). Recall that \( P_A \) is the projection matrix of the range of \( A \). Since \( A \) is circulant, \( P_A \) is also circulant and diagonalized by the DFT matrix via \( FP_A F^* = \Lambda_P A \) where \( \Lambda_P A = \Lambda A A_P^+ \). Accordingly, we can write

\[
A A^+ w = A P_A x = Ax = w
\]

(4.37)

exhibiting \( w \) as a processing of \( A^+ w \), thus, by the data processing inequality we get

\[
I(w; Bv) \leq I(A^+ w; Bv)
\]

(4.38)

that together with (4.36) implies

\[
I(w; Bv) = I(A^+ w; Bv)
\]

(4.39)

The above showed that since \( w \) is in the range of \( A \), its processing via \( A^+ \) is invertible and, thus, the mutual information is preserved under such processing of \( w \).

Similar to the above arguments, we can show that since the system output \( Bv \) is in the range of \( B \), its processing via \( B^+ \) is invertible and, therefore, the mutual information
is preserved under such filtering of $Bv$:

$$ I(\mathbf{A}^+w; Bv) = I(\mathbf{A}^+w; B^+Bv) = I(\mathbf{A}^+w; P_Bv). \quad (4.40) $$

Consider the decomposition of $\mathbf{A}^+w$ based on its projections on $B$’s range and nullspace, namely,

$$ \mathbf{A}^+w = P_B\mathbf{A}^+w + (I - P_B)\mathbf{A}^+w. \quad (4.41) $$

Observe that

$$ (I - P_B)\mathbf{A}^+w = F^*(I - \Lambda P_B)FF^*\Lambda_B^+FW = F^*(I - \Lambda P_B)\Lambda_B^+w^F \quad (4.42) $$

and the second random variable considered by the mutual information in (4.40) is

$$ P_Bv = F^*\Lambda_BFW = F^*\Lambda_B^+w^F \quad (4.43) $$

where $w^F$ is the DFT-domain representation of $w$. Note that the diagonal projection matrices $\Lambda_P \triangleq \Lambda_B\Lambda_B^+$ and $I - \Lambda_P$ appearing in (4.43) and (4.42), respectively, imply that the DFT-domain representations of $(I - P_B)\mathbf{A}^+w$ and $P_Bv$ have complementary non-overlapping supports (defined by the component indices where the value may be nonzero). Let us develop the mutual information in (4.40) to leverage the mentioned structures and the decomposition (4.41):

$$ I(\mathbf{A}^+w; P_Bv) = $$

$$ = I(F\mathbf{A}^+w; FP_Bv) = I(\Lambda_B^+w^F; \Lambda_B^+v^F) \quad (4.44) $$

$$ \geq \sum_{k:b_k^F \neq 0} I(a_k^+w_k^F; v_k^F) + \sum_{k:b_k^F = 0} I(a_k^+w_k^F; 0) \quad (4.45) $$

$$ = \sum_{k:b_k^F \neq 0} I(a_k^+w_k^F; v_k^F) \quad (4.46) $$

$$ \geq \sum_{k:b_k^F \neq 0} I(a_k^+w_k^F; v_k^F) \quad (4.47) $$

where (4.44) is due to the preservation of mutual information under a unitary transformation. The lower bound in (4.46) emerges from the independence of the DFT-domain variables $w_k^F$, $k = 0, ..., N - 1$, since they originate in the cyclo-stationary Gaussian signal $\mathbf{x}$ processed by the circulant matrix $\mathbf{A}$. In Section 4.5.3 we exhibit a backward channel construction based on independent components $v_k^F$ in the DFT domain, therefore, satisfying the inequality (4.46) with equality. Using (4.47) and pending on the
construction shown in Section 4.5, we state that

\[
I (A^+w; P_Bv) = \sum_{k: h_k^T \neq 0} I \left( a_k^F + w_k^F; v_k^F \right)
= I (P_BA^+w; P_Bv) \quad \text{(4.48)}
\]

Note that \(P_BA^+w\) is in the range of \(A\), hence, using similar arguments to those given in the last paragraph, one can show that

\[
I (P_BA^+w; P_Bv) = I (P_BA^+w; PA_PBv) \quad \text{(4.49)}
\]

Furthermore, since \(P_BA^+w\) belongs also to the range of \(B\), its processing via \(B^+\) is invertible, thus,

\[
I (P_BA^+w; PA_PBv) = I (B^+A^+w; PA_PBv) \quad \text{(4.50)}
\]

Due to the circulant structure of \(P_A\) and \(P_B\), they commute and, therefore,

\[
I (P_BA^+w; PA_PBv) = I (B^+A^+w; BP_Av) \quad \text{(4.51)}
\]

that is the mutual information in the cost of problem (4.20). To conclude, we showed the equivalence of the costs (based on the backward-channel construction presented next in Section 4.5.3) and the constraints of problems (4.18) and (4.20), therefore, these optimization problems are interchangeable.

### 4.5.3 Equivalence of Optimization Problems (4.20) and (4.21)

Since \(A\) and \(B\) are circulant matrices, the expected distortion \(E \left\{ \| AB (\tilde{w} - v) \|_2^2 \right\} \), appearing in the constraint of (4.20), has the following additively-separable form in the DFT domain

\[
E \left\{ \| AB (\tilde{w} - v) \|_2^2 \right\} = E \left\{ \| A_A A_B (\tilde{w}^F - v^F) \|_2^2 \right\}
= \sum_{k=0}^{N-1} |a_k^F b_k^F|^2 E \left\{ |\tilde{w}_k^F - v_k^F|^2 \right\}. \quad \text{(4.52)}
\]

This expected distortion formulation motivates us to address the entire rate-distortion optimization in the DFT domain and with respect to a pseudoinverse filtered version of the input \(w\). For this purpose we will treat next the optimization cost of the problem given in (4.20).

Since \(\tilde{w} \triangleq B^+A^+w\) is a cyclo-stationary Gaussian signal, we apply the following
familiar lower bounds:

\[ I(\tilde{w}; P_B P_A v) = I(F\tilde{w}; FP_B P_A v) \] (4.53)
\[ = I(\tilde{w}^F; AP_B AP_A v^F) \] (4.54)
\[ \geq \sum_{k \in K_{AB}} I(\tilde{w}_k^F; v_k^F) \] (4.55)

where (4.53) is due to the invariance of mutual information under a unitary transformation, (4.54) exhibits the diagonal forms of the projection matrices, and the bound (4.55) is due to the independence of the DFT coefficients \( \{\tilde{w}_k^F\}_{k=0}^{N-1} \) stemming from cyclo-stationary Gaussian characteristics of \( \tilde{w} \). Moreover, note that (4.55) refers only to mutual information of DFT components belonging to the range of \( AB \), this is due to (4.54) where \( v^F \) components corresponding to the nullspace of \( AB \) are zeroed (hence, they yield zero mutual information and zero rate). The next lower bound emerges from the definition of the rate-distortion function for each of the components, i.e.,

\[ \sum_{k \in K_{AB}} I(\tilde{w}_k^F; v_k^F) \geq \sum_{k \in K_{AB}} R_k(d_k) \] (4.56)
\[ = \sum_{k \in K_{AB}} \left[ \frac{1}{2} \log \left( \frac{\lambda_k(\tilde{w})}{d_k} \right) \right] \] (4.57)

where the last equality relies on the rate-distortion function formulation for a scalar Gaussian source. The \( k^{th} \) variable here is \( \tilde{w}_k^F \), having the a variance denoted as \( \lambda_k(\tilde{w}) \). Here, the \( k^{th} \)-component rate corresponds to an expected squared-error distortion denoted as \( d_k \triangleq E\left\{||\tilde{w}_k^F - v_k^F||^2\right\} \).

The mutual-information lower bound in (4.57) is further minimized under the total distortion constraint that expresses the weights introduced in (4.52) for each of the DFT components, i.e., the distortion-allocation optimization is

\[ \min_{\{d_k\}_{k \in K_{AB}}} \sum_{k \in K_{AB}} \frac{1}{2} \log \left( \frac{\lambda_k(\tilde{w})}{d_k} \right) \] s.t. \[ \sum_{k \in K_{AB}} |a_k^F b_k^F|^2 d_k \leq Nd \] (4.58) \[ 0 \leq d_k \leq \lambda_k(\tilde{w}), \quad k \in K_{AB}. \]

where the operator \([\cdot]_+\) in (4.57) was replaced with componentwise constraints. The optimization (4.58) is solved using Lagrangian optimization and the KKT conditions. We denote here the optimal distortions as \( \{\hat{d}_k\}_{k=0}^{N-1} \), where for \( k \notin K_{AB} \) we set \( \hat{d}_k = 0 \), and use them next for showing the achievability of the mutual-information lower bound.

We define \( A_d \) as the \( N \times N \) diagonal matrix with \( \hat{d}_k \) as the \( k^{th} \) diagonal value. Also recall that \( H \triangleq AB \) and the related definitions given above. Consider the following
construction for a backward channel producing \( \tilde{w} \) from \( v \). Let

\[
\begin{align*}
v & \sim \mathcal{N} \left( 0, H^+ R_w H^{++} - H^+ F^* \Lambda_d^t F H^+ \right) \quad (4.59) \\
z & \sim \mathcal{N} \left( 0, H^+ F^* \Lambda_d^t F H^+ \right) \quad (4.60)
\end{align*}
\]

be two independent random vectors, constructing \( \tilde{w} \) via

\[
\tilde{w} = v + z, \quad (4.61)
\]

hence, \( \tilde{w} \sim \mathcal{N} \left( 0, H^+ R_w H^{++} \right) \), agreeing with \( \tilde{w} = H^+ w \) where \( w \sim \mathcal{N} \left( 0, R_w \right) \). Additionally, the construction (4.59)-(4.61) leads to

\[
E \left\{ \|H (\tilde{w} - v)\|_2^2 \right\} = E \left\{ \|Hz\|_2^2 \right\} = E \{ z^* H^* Hz \}
= E \{ Trace \{ z^* H^* Hz \} \}
= E \{ Trace \{ Hz \tilde{H}^* \} \}
= Trace \{ H R_w H^* \}
= Trace \{ H H^+ F^* \Lambda_d^t F H^+ H^* \}
= Trace \{ F^* \Lambda_H \tilde{H}^* \Lambda_d^t \tilde{H}^* \}
= Trace \{ F^* \Lambda_d^t \}
= Trace \{ \Lambda_d \}
= Nd \quad (4.62)
\]

that reaches the maximal allowed distortion in (4.58). We used here the relation \( \Lambda_H \tilde{H}^* \Lambda_d^t \tilde{H}^* \Lambda_H^* = \Lambda_d \), emerging from the fact that \( d_k = 0 \) for components in the nullspace of \( H \). This construction fulfills (4.46), (4.55) and (4.56) with equality, thus, proves the equivalence of the optimization problems (4.18), (4.20), and (4.21).
Chapter 5

Compression for Multiple Reconstructions

In Chapter 3 we proposed an optimization methodology to adjust standard image/video compression to a known type of display presenting the decompressed signal to the viewer. Our framework essentially pre-compensates the display degradation from the compression standpoint in a rate-distortion optimized manner. Here we extend the problem settings from Chapter 3 to optimize the compression with respect to a set of display devices, described by several linear rendering models and their probabilities to be in use by consumers. One can interpret the display models and their usage probabilities as a characterization of a multimedia distribution network.

5.1 Introduction

Multimedia content is often distributed using broadcast and "on-demand" services reaching consumers with various display devices. Therefore, rendering the image/video can widely differ due to various technical aspects such as the specific display technology, different screen resolutions, etc. Such multimedia distribution systems fundamentally rely on lossy compression in order to meet storage and transmission-bandwidth limitations. However, while the displayed signals are the important outcomes of the flow, the compression is usually optimized only with respect to the decompressed signal, ignoring the subsequent processing and degradations occurring at the different displays. In this chapter we study the problem of optimizing signal compression to a known set of display settings having different usage probabilities.

We formulate a rate-distortion optimization to trade-off the compression bit-cost and the expected mean-squared error of the displayed signal. Similar to our methods presented in Chapters 3-4, we address the computationally hard optimization using the alternating direction method of multipliers (ADMM) [15] translating the task to sequentially applying standard compressions (that are network independent!) and $\ell_2$-constrained deconvolutions expressing the network structure. This procedure can
be generically adapted to various network layouts and to any standard compression technique, providing network-optimized binary data that is compatible with desired standard decompression processes.

The important problem of various display devices is treated also from the perspective of scalable image/video coding methods (see, e.g., the extension of the HEVC standard in [66]), where the signal is coded in layers of increasing quality/resolution to be peeled by the network or the user device. In contrast, we take here another viewpoint on the problem, optimizing a single (non-layered) compression of a signal to a given collection of rendering models.

We demonstrate our general approach for image compression using the state-of-the-art HEVC standard coupled with various simplified display models in the form of linear blur operators following the decompression. While another recent method [67] optimizes image rendering with respect to a perceptual quality metric, we present here (and in Chapter 3) a method to globally optimize the flow of compression, decompression and rendering. Since our optimization goal and the distortion type differ from those in [67], the two methods cannot be quantitatively compared. In our experiments we compared our approach to regular HEVC compression, and to preceding the compression with the Expected Patch Log Likelihood (EPLL) deblurring method [23] adapted to the same fidelity term as we use in our method. The rate-distortion performance of the various methods clearly exhibit our method as the leading approach at high bit-rate compression.

5.2 The Proposed Method

5.2.1 Problem Formulation

We consider the network structure described in Fig. 5.1, starting with an input signal in the form of an N-length column-vector $x \in \mathbb{R}^N$ that is compressed and
distributed over the network to users having various reconstruction systems. We describe the lossy compression procedure using the function $C : \mathbb{R}^N \rightarrow \mathcal{B}$, mapping the $N$-dimensional input-signal domain to the discrete set $\mathcal{B}$ of compressed representations in the form of variable-length binary descriptions. The compression of $x$ is denoted by $b = C(x)$, where $b \in \mathcal{B}$ is the compressed data to transmit over the network to an arbitrary number of users. The users have reconstruction systems that first decompress the data via $v = F(b)$, where $F : \mathcal{B} \rightarrow \mathcal{S}$ maps the binary compressed representations in $\mathcal{B}$ to the respective decompressed signals in the discrete set $\mathcal{S} \subset \mathbb{R}^N$. The decompressed signal $v$ (which is an $N$-length column vector) further goes through a linear operation, associated with a processing/degradation stage, that produces the reconstruction available to the user. In the case of visual signals, the post-decompression component may be a display rendering the viewed image. We assume a user may have one of $K$ reconstruction systems (where $K$ is a positive finite integer), differing in the linear operator applied after decompression. The post-decompression linear operator of the $k^{th}$ system type ($k = 1, ..., K$) is denoted by the $N \times N$ matrix $H_k$, producing a corresponding reconstructed (output) signal

$$y_k = H_k v.$$ (5.1)

We assume the portions of using each of the $K$ reconstruction systems are known and denoted by $p_1, ..., p_K \geq 0$, where $\sum_{k=1}^{K} p_k = 1$. Accordingly, a network user can be modeled to have a reconstruction system of a type corresponding to a discrete random variable over the values $\{1, ..., K\}$ with the respective probabilities $p_1, ..., p_K$. Then, by (5.1) the reconstructed signal is a random vector $y$ having the value $y_k$ with probability $p_k$ for $k = 1, ..., K$. For a given (deterministic) input signal $x$ and its decompressed version $v$, and by the network structure, we quantify the expected mean-squared-error (MSE) of the reconstruction as

$$D(x, v) \triangleq \frac{1}{N} \sum_{k=1}^{K} p_k \|x - H_k v\|_2^2.$$ (5.2)

Our goal here is to optimize the rate-distortion performance of the network for a given input signal $x$. Accordingly, we formulate the task as the minimization of the compression bit-cost under constrained expected distortion (5.2), namely,

$$\hat{v} = \arg\min_{v \in \mathcal{S}} \ R(v) \quad \text{s.t.} \quad \frac{1}{N} \sum_{k=1}^{K} p_k \|x - H_k v\|_2^2 \leq d$$ (5.3)

where $R(v)$ evaluates the length of the binary compressed description $b \in \mathcal{B}$ matched to the decompressed signal $v$, and $d \geq 0$ is the allowed distortion.

Similar to contemporary compression tasks (see, e.g., [26, 29]), we turn (5.3) into its
unconstrained Lagrangian form

\[ \hat{v} = \arg\min_{v \in \mathcal{S}} R(v) + \lambda \frac{1}{N} \sum_{k=1}^{K} p_k \| x - H_k v \|_2^2 \]  

(5.4)

where \( \lambda \geq 0 \) is a Lagrange multiplier matching to a distortion constraint \( d_\lambda \geq 0 \) (such coding without a specified distortion constraint is prevalent, e.g., in video coding [29]).

Since we consider the compression of high-dimensional signals (i.e., \( N \) is large) the discrete set \( \mathcal{S} \) is prohibitively large, meaning that a direct solution of the Lagrangian form in (5.4) is impractical for arbitrarily structured matrices \( \{H_k\}_{k=1}^{K} \). When \( H_k = I \) for \( k = 1, \ldots, K \), the optimization in (5.4) reduces to the standard compression form [26, 24], ignoring the network-oriented problem, and practically solvable using block-based architectures that decompose the problem to a sequence of block-level optimizations of sufficiently low dimensions.

### 5.2.2 Practical Iterative Procedure

We employ the alternating direction method of multipliers (ADMM) technique [15] to resolve the computationally challenging problem (5.4) when the post-decompression operators \( \{H_k\}_{k=1}^{K} \) are arbitrarily structured. We begin by splitting the optimization variable such that (5.4) becomes

\[ \hat{v} = \arg\min_{v \in \mathcal{S}, z \in \mathbb{R}^N} R(v) + \lambda \frac{1}{N} \sum_{k=1}^{K} p_k \| x - H_k z \|_2^2 \quad \text{s.t.} \quad v = z \]  

(5.5)

where \( z \in \mathbb{R}^N \) is an auxiliary variable that is not limited to the discrete set \( \mathcal{S} \). Applying the scaled form of the augmented Lagrangian and the method of multipliers [15, Ch. 2] on (5.5) yields an iterative process formulated as

\[ \begin{align*}
(\hat{v}^{(t)}, \hat{z}^{(t)}) &= \arg\min_{v \in \mathcal{S}, z \in \mathbb{R}^N} R(v) + \lambda \frac{1}{N} \sum_{k=1}^{K} p_k \| x - H_k z \|_2^2 + \frac{\beta}{2} \| v - z + u^{(t)} \|_2^2 \\
u^{(t+1)} &= u^{(t)} + \left( \hat{v}^{(t)} - \hat{z}^{(t)} \right),
\end{align*} \]  

(5.6, 5.7)

where \( t \) denotes the iteration index, \( u^{(t)} \in \mathbb{R}^N \) is the scaled dual variable, and \( \beta \) is an auxiliary parameter introduced by the augmented Lagrangian. We get the ADMM form of the problem by applying one iteration of alternating minimization on (5.6), leading
to a sequence of easier optimizations:

\[
\hat{v}(t) = \arg\min_{v \in S} R(v) + \frac{\beta}{2} \|v - \tilde{z}(t)\|_2^2 
\]

\[
\hat{z}(t) = \arg\min_{z \in \mathbb{R}^N} \frac{\lambda}{N} \sum_{k=1}^{K} p_k \|x - H_k z\|_2^2 + \frac{\beta}{2} \|z - \tilde{v}(t)\|_2^2 
\]

\[
u(t+1) = \nu(t) + (\hat{v}(t) - \hat{z}(t)).
\]

where \(\tilde{z}(t) = \tilde{z}(t-1) - \nu(t)\) and \(\tilde{v}(t) = \tilde{v}(t) + \nu(t)\). Nicely, the compression architecture \(\{S, R\}\) and the network layout described by \(\{H_k, p_k\}_{k=1}^{K}\) were separated by the ADMM to distinct (and simpler) optimization tasks.

The optimization formulation in (5.8) coincides with the Lagrangian rate-distortion optimization utilized for standard compression tasks employing the usual (network independent) MSE distortion metric (here the effective Lagrange multiplier is \(\tilde{\lambda} = \frac{\beta N}{2}\)). Hence, we propose to replace the solution of (5.8) with a standard compression (and decompression) method – even one that does not exactly follow the Lagrangian optimization in (5.8). We refer to the standard compression and decompression as

\[
b(t) = \text{StandardCompress}(\hat{z}(t), \theta)
\]

\[
\hat{v}(t) = \text{StandardDecompress}(b(t))
\]

where \(\theta\) is a parameter generalizing the Lagrange multiplier part in regulating the rate-distortion tradeoff (see Algorithm 5.1). The last generalizations establish the proposed procedure as a generic methodology for optimizing any compression method to particular network layouts.

The optimization in (5.9) can be interpreted as an extended \(\ell_2\)-constrained deconvolution problem, here including a combination of several fidelity terms associated with the degradation operators \(\{H_k\}_{k=1}^{K}\). The analytic solution of (5.9) is

\[
\hat{z}(t) = \left( \sum_{k=1}^{K} p_k H_k^T H_k + \frac{\beta N}{2\lambda} I \right)^{-1} \left( \sum_{k=1}^{K} p_k H_k^T x + \frac{\beta N}{2\lambda} \tilde{v}(t) \right)
\]

exhibiting it as a linear combination of \(x\) and \(\tilde{v}(t)\). We define the parameter \(\tilde{\beta} \triangleq \frac{\beta N}{2\lambda}\) and use it in the generic method summarized in Algorithm 5.1.

5.3 Experimental Results

Let us demonstrate our method for optimizing the HEVC still-image compression standard (implemented in the software in [50]) to three possible blur operators degrading the decompressed image. The post-decompression linear operators \(H_1, H_2,\) and \(H_3\) correspond to shift-invariant Gaussian blur kernels (of \(15 \times 15\) pixels size) having
Figure 5.2: The regular and proposed methods applied for the ‘Flower and Bugs’ image. The PSNR values in this figure are for the individual reconstructions, i.e., using the regular MSE and not the expected one from (5.2) that is used in the rest of the chapter.
Algorithm 5.1 Generic Network-Optimized Compression

1: Inputs: $x$, $\theta$, $\tilde{\beta}$.
2: Initialize $t = 0$, $\hat{z}^{(0)} = x$, $u^{(1)} = 0$.
3: repeat
4: $t \leftarrow t + 1$
5: $\tilde{z}^{(t)} = \hat{z}^{(t-1)} - u^{(t)}$
6: $b^{(t)} = \text{StandardCompress}(\tilde{z}^{(t)}, \theta)$
7: $\tilde{v}^{(t)} = \text{StandardDecompress}(b^{(t)})$
8: $\hat{v}^{(t)} = \tilde{v}^{(t)} + u^{(t)}$
9: $\hat{z}^{(t)} = \left( \sum_{k=1}^{K} p_k H_k^T H_k + \tilde{\beta} I \right)^{-1} \left( \sum_{k=1}^{K} p_k H_k^T x + \tilde{\beta} \hat{v}^{(t)} \right)$
10: $u^{(t+1)} = u^{(t)} + (\hat{v}^{(t)} - \hat{z}^{(t)})$
11: until stopping criterion is satisfied
12: Output: The binary compressed data $b^{(t)}$.

standard deviations 0.6, 0.8, and 1, respectively, and usage probabilities of $p_1 = 0.6$, $p_2 = 0.3$, and $p_3 = 0.1$.

To evaluate our method we constructed three competing techniques also using HEVC image compression, and compared them to our method\textsuperscript{1}. The PSNR-bitrate curves of the examined methods (see, e.g., Fig. 5.3b) were created for each of the 12 examined images (see Table 5.1) by applying their HEVC component for 9 quantization parameter (QP) values equally-spaced between 1 to 41. The first competing approach is to regularly compress without any pre/post processing (while the decompression is still followed by the inevitable deterioration). As expected, this naive method performs poorly. The second competing procedure precedes the compression with deconvolution using the Expected Patch Log Likelihood (EPLL) method relying on a Gaussian Mixture Model (GMM) prior learned for natural images (see [23]). The EPLL implementation used here is with respect to a fidelity term corresponding to (5.2) and suitable parameter settings.

The third competing method is our pre-compensating compression from Chapter 3, optimized only for a single display (corresponding to the highest probability).

The implementation of the proposed method (Algorithm 5.1) uses a $\tilde{\beta}$ value based on the HEVC quantization parameter (the $\tilde{\beta}$ value here is 10 times the value formulated in Chapter 3). The stopping criterion was defined to a maximal number of 40 iterations or to end earlier when $\hat{v}^{(t)}$ and $\hat{z}^{(t)}$ converge or diverge (as described in Chapter 3).

The evaluation of the PSNR-bitrate curves summarized in Table 5.1 and exemplified for one image in Fig. 5.3b, showing that our method outperforms the other techniques at high bit-rates, where we achieve significant PSNR gains compared to the regular, the EPLL-based, and the single display optimization procedures. The average PSNR gains in Table 5.1 were computed based on the BD-PSNR metric [38, 54] for the high bit-rate range (here defined by QP values 1,6,11,16).

\textsuperscript{1}The Peak Signal-to-Noise Ratio (PSNR) here relies on the expected reconstruction MSE given in (5.2), i.e., $PSNR = 10 \log_{10} \left( \frac{P^2}{D(x,v)} \right)$ where $x$ and $v$ are the input and the decompressed signals, respectively, and $P$ is the maximal signal-value generally possible.
Figure 5.3: Methods evaluation for the image 'Flower and Bugs'.

Table 5.1: Method Evaluation for 12 Images

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Image</th>
<th>Average PSNR Gains at High Bit-Rates</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Proposed over</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Regular</td>
</tr>
<tr>
<td>TEST</td>
<td>Almonds</td>
<td>5.25</td>
</tr>
<tr>
<td>IMAGES</td>
<td>Cards</td>
<td>4.83</td>
</tr>
<tr>
<td>300x300</td>
<td>Duck toys</td>
<td>4.83</td>
</tr>
<tr>
<td></td>
<td>Garden table</td>
<td>4.70</td>
</tr>
<tr>
<td>[52]</td>
<td>House &amp; lawn</td>
<td>3.53</td>
</tr>
<tr>
<td>UCID</td>
<td>Tree</td>
<td>3.83</td>
</tr>
<tr>
<td>384x512</td>
<td>Garden</td>
<td>3.26</td>
</tr>
<tr>
<td></td>
<td>Teddy bear</td>
<td>4.96</td>
</tr>
<tr>
<td>[53]</td>
<td>Bears</td>
<td>4.73</td>
</tr>
<tr>
<td>Berkeley</td>
<td>Boats</td>
<td>4.05</td>
</tr>
<tr>
<td>481x321</td>
<td>Butterfly</td>
<td>4.85</td>
</tr>
<tr>
<td></td>
<td>Flower &amp; Bugs</td>
<td>4.83</td>
</tr>
</tbody>
</table>

In Figure 5.2 we present visual results for the compression of the 'Flower and Bugs' image (see Fig. 5.3a, where only a portion of the input image is presented due to lack of space). Figures 5.2a and 5.2e exhibit the decompressed images (before degradation) using the regular approach and the proposed method, respectively. Figures 5.2b-5.2d and 5.2f-5.2h show the three simulated displayed versions of the decompressed images. Evidently, we get significantly higher PSNR values (that correspond to the regular MSE measure) at a similar (slightly lower) bit-rate. Our method produces an overly-sharpened decompressed image (Fig. 5.2e) that is later balanced with the rendering blur, leading to better displayed images (Figs. 5.2f-5.2h).
Chapter 6

Restoration by Compression

In this chapter we explore the topic of complexity-regularized restoration, where the likelihood of candidate estimates are determined by their compression bit-costs. Using our ADMM-based methodology for intricate rate-distortion optimizations, we develop three practical methods for restoration using standard compression techniques. Two of the proposed methods rely on a new shift-invariant complexity regularizer, evaluating the total bit-cost of the signal shifted versions. We explain few of the main ideas of our approach using a theoretical analysis of complexity-regularized restoration of a Gaussian signal from deterioration of a linear shift-invariant operator and additive white Gaussian noise. Experiments for deblurring and inpainting of images using the JPEG2000 and HEVC techniques show good results.

6.1 Introduction

Signal restoration methods are often posed as inverse problems using regularization terms. While many solutions can explain a given degraded signal, using regularization will provide signal estimates based on prior assumptions on signals. One interesting regularization type measures the complexity of the candidate solution in terms of its compression bit-cost. Indeed, encoders (that yield the bit cost) rely on signal models and allocate shorter representations to more likely signal instances. This approach of complexity-regularized restoration is an attractive meeting point of signal restoration and compression, two fundamental signal-processing problems.

Numerous works [6, 1, 2, 3, 4, 7, 5] considered the task of denoising a signal corrupted by an additive white Gaussian noise using complexity regularization. In [1, 5], this idea is translated to practically estimating the clean signal by employing a standard lossy compression of its noisy version. However, more complex restoration problems (e.g., deblurring, super resolution, inpainting), involving non-trivial degradation operators, do not lend themselves to a straightforward treatment by compression techniques designed for the squared-error distortion measure. Moulin and Liu [68] studied the complexity regularization idea for general restoration problems, presenting a thorough
theoretical treatment together with a limited practical demonstration of Poisson denois-
ing based on a suitably designed compression method. Indeed, a general method for complexity-regularized restoration remained as an open question for a long while until our research presented here, proposing a generic and practical approach flexible in both the degradation model addressed and the compression technique utilized.

Our strategy for complexity-regularized signal restoration relies on the alternating direction method of multipliers (ADMM) approach \cite{15}, decomposing the difficult optimization problem into a sequence of easier tasks including $\ell_2$-regularized inverse problems and standard rate-distortion optimizations (with respect to a squared-error distortion metric). A main part of our methodology is to replace the rate-distortion optimization with standardized compression techniques enabling an indirect utilization of signal models used for efficient compression designs. Moreover, our method relates to various contemporary concepts in signal and image processing. The recent frameworks of Plug-and-Play Priors \cite{17, 18} and Regularization-by-Denoising \cite{20} suggest leveraging a Gaussian denoiser for more complicated restoration tasks, achieving impressive results (see, e.g., \cite{17, 18, 20, 64, 69, 19}). Essentially, our approach is the compression-based counterpart for denoising-based restoration concepts from \cite{17, 18, 20}.

Commonly, compression methods process the given signal based on its decomposition into non-overlapping blocks, yielding block-level rate-distortion optimizations based on block bit-costs. The corresponding complexity measure sums the bit-costs of all the non-overlapping blocks, however, note that this evaluation is shift sensitive. This fact motivates us to propose a shift-invariant complexity regularizer by quantifying the bit-costs of all the overlapping blocks of the signal estimate. This improved regularizer calls for our restoration procedure to use averaging of decompressed signals obtained from compressions of shifted signals. Our shift-invariant approach conforms with the Expected Patch Log-Likelihood (EPLL) idea \cite{23}, where a full-signal regularizer is formed based on a block-level prior in a way leading to averaging MAP estimates of shifted signal versions. Our extended method also recalls the cycle spinning concept, presented in \cite{70} for wavelet-based denoising. Additional resemblance is to the compression postprocessing techniques in \cite{71, 72} enhancing a given decompressed image by averaging supplementary compression-decompression results of shifted versions of the given image, thus, our method generalizes this approach to any restoration problem with an appropriate consideration of the degradation operator. Very recent works \cite{73, 74} suggested the use of compression techniques for compressive sensing of signals and images, but our approach examines other perspectives and settings referring to restoration problems as will be explained below.

In this chapter we present new algorithms, a respective theoretical analysis, and extensive experimental results. In this chapter we provide new results demonstrating the practical complexity-regularized restoration approach for image deblurring. While deblurring is a challenging restoration task, we present compelling results obtained using the JPEG2000 method and the image compression profile of the HEVC standard
An objective comparison to other deblurring techniques showed that the proposed HEVC-based implementation provides good deblurring results. Moreover, we also address the image inpainting problem using our ADMM-based approach in conjunction with the JPEG2000 and HEVC compression standards. In the examined settings we restore images from a severe degradation of 80% missing pixels. Interestingly, our compression-based image inpainting approach can be perceived as the dual concept of inpainting-based compression of images and videos suggested in, e.g., [10, 11, 12] and discussed also in [75].

Another prominent contribution of this chapter is the new theoretical study of the problem of complexity-regularized restoration, considering the estimation of a cyclo-stationary Gaussian signal from a degradation procedure consisting of a linear shift-invariant operator and additive white Gaussian noise. We gradually establish few equivalent optimization forms, emphasizing two main concepts for complexity-regularized restoration: the degraded signal should go through a simple inverse filtering procedure, and then should be compressed so that the decompression components will have a varying quality distribution determined by the degradation-filter energy-distribution. We explain how these ideas materialize in the practical approach we propose, thus, establishing a theoretical reasoning for the feasible complexity-regularized restoration.

This chapter is organized as follows. In section 6.2 we overview the settings of the complexity-regularized restoration problem. In section 6.3 we present the proposed practical methods for complexity-regularized restoration. In section 6.4 we theoretically analyze particular problem settings where the signal is a cyclo-stationary Gaussian process. In section 6.5 we provide experimental results for image deblurring and inpainting.

6.2 Complexity-Regularized Restoration: Problem Settings

6.2.1 Regularized Restoration of Signals

In this chapter we address the task of restoring a signal $x_0 \in \mathbb{R}^N$ from a degraded version, $y \in \mathbb{R}^M$, obeying the prevalent deterioration model:

$$y = Hx_0 + n$$  \hspace{1cm} (6.1)$$

where $H$ is a $M \times N$ matrix being a linear degradation operator (e.g., blur, pixel omission, decimation) and $n \in \mathbb{R}^M$ is a white Gaussian noise vector having zero mean and variance $\sigma_n^2$.

Maximum A-Posteriori (MAP) estimation is a widely-known statistical approach forming the restored signal, $\hat{x}$, via

$$\hat{x} = \arg\max_x p(x|y)$$  \hspace{1cm} (6.2)$$
where \( p(x|y) \) is the posterior probability. For the above defined degradation model (6.1), incorporating additive white Gaussian noise, the MAP estimate reduces to the form of

\[
\hat{x} = \arg\min_x \frac{1}{2\sigma_n^2} \|Hx - y\|^2_2 - \log p(x)
\] (6.3)

where \( p(x) \) is the prior probability that, here, evaluates the probability of the candidate solution.

Another prevalent restoration approach, embodied in many contemporary techniques, forms the estimate via the optimization

\[
\hat{x} = \arg\min_x \|Hx - y\|^2_2 + \mu s(x)
\] (6.4)

where \( s(x) \) is a general regularization function returning a lower value for a more likely candidate solution, and \( \mu \geq 0 \) is a parameter weighting the regularization effect. This strategy for restoration based on arbitrary regularizers can be interpreted as a generalization of the MAP approach in (6.3). Specifically, comparing the formulations (6.4) and (6.3) exhibits the regularization function \( s(x) \) and the parameter \( \mu \) as extensions of \((-\log p(x))\) and the factor \(2\sigma_n^2\), respectively.

Among the various regularization functions that can be associated with the general restoration approach in (6.4), we explore here the class of complexity regularizers measuring the required number of bits for the compressed representation of the candidate solution. The practical methods presented in this section focus on utilizing existing (independent) compression techniques, implicitly employing their underlying signal models for the restoration task.

### 6.2.2 Operational Rate-Distortion Optimization

The practical complexity-regularized restoration methods in this section are developed with respect to a compression technique obeying the following conceptual design. The signal is segmented to equally-sized non-overlapping blocks (each is consisted of \(N_b\) samples) that are independently compressed. The block compression procedure is modeled as a general variable-rate vector quantizer relying on the following mappings. The compression is done by the mapping \( C_b : \mathbb{R}^{N_b} \rightarrow B \) from the \(N_b\)-dimensional signal-block domain to a discrete set \( B_b \) of binary compressed representations (that may have different lengths). The decompression procedure is associated with the mapping \( F_b : B_b \rightarrow S_b \), where \( S_b \subset \mathbb{R}^{N_b} \) is a finite discrete set (a codebook) of block reconstruction candidates. For example, consider the block \( x_{block} \in \mathbb{R}^{N_b} \) that its binary compressed representation in \( B_b \) is given via \( b = C_b(x_{block}) \) and the corresponding reconstructed block in \( S_b \) is \( \hat{x}_{block} = F_b(b) \). Importantly, it is assumed that shorter codewords are coupled with block reconstructions that are, in general, more likely.

The signal \( x \) is compressed based on its segmentation into a set of blocks \( \{x_i\}_{i \in I} \)
(where \( \mathcal{I} \) denotes the index set of blocks in the non-overlapping partitioning of the signal). In addition we introduce the function \( R_b(z) \) that evaluates the bit-cost (i.e., the length of the binary codeword) for the block reconstruction \( z \in S_b \). Then, the operational rate-distortion optimization corresponding to the described architecture and a squared-error distortion metric is

\[
\{\tilde{x}_i\}_{i \in \mathcal{I}} = \arg\min_{\{v_i\}_{i \in \mathcal{I}} \in S_b} \sum_{i \in \mathcal{I}} \|x_i - v_i\|_2^2 + \lambda \sum_{i \in \mathcal{I}} R_b(v_i),
\]

where \( \lambda \geq 0 \) is a Lagrange multiplier corresponding to some total compression bit-cost. Importantly, the independent representation of non-overlapping blocks allows solving (6.5) separately for each block [24, 26].

Our mathematical developments require the following algebraic tools for block handling. The matrix \( P_i \) is defined to provide the \( i^{th} \) block from the complete signal via the standard multiplication \( P_i x = x_i \). Note that \( P_i \) can extract any block of the signal, even one that is not in the non-overlapping grid \( \mathcal{I} \). Accordingly, the matrix \( P^T_i \) locates a block in the \( i^{th} \) block-position in a construction of a full-sized signal and, therefore, lets to express the a complete signal as \( x = \sum_{i \in \mathcal{I}} P^T_i x_i \).

Now we can use the block handling operator \( P_i \) for expressing the block-based rate-distortion optimization in its corresponding full-signal formulation:

\[
\tilde{x} = \arg\min_{v \in S} \|x - v\|_2^2 + \lambda R(v).
\]

where \( S \) is the full-signal codebook, being the discrete set of candidate reconstructions for the full signal, defined using the block-level codebook \( S_b \) as

\[
S = \left\{ v \mid v = \sum_{i \in \mathcal{I}} P^T_i v_i, \ \{v_i\}_{i \in \mathcal{I}} \in S_b \right\}.
\]

Moreover, the regularization function in (6.6) is the total bit cost of the reconstructed signal defined for \( v \in S \) as \( R(v) \triangleq \sum_{i \in \mathcal{I}} R_b(P_i v) \).

### 6.2.3 Complexity-Regularized Restoration: Basic Optimization Formulation

While the regularized-restoration optimization in (6.4) is over a continuous domain, the operational rate-distortion optimization in (6.6) is a discrete problem with solutions limited to the set \( S \). Therefore, we extend the definition of the block bit-cost evaluation function such that it is defined for any \( z \in \mathbb{R}^{N_b} \) via

\[
\bar{R}_b(z) = \begin{cases} 
R_b(z) & , z \in S_b \\
\infty & , z \notin S_b
\end{cases},
\]

\[
(6.8)
\]
and the corresponding extension of the total bit-cost \( \bar{R}(x) \triangleq \sum_{i \in I} \bar{R}_b(P_i, x) \) is defined for any \( x \in \mathbb{R}^N \).

Now we define the complexity regularization function as

\[
\hat{R}(x) = \bar{R}(x)
\]

and the corresponding restoration optimization is

\[
\hat{x} = \arg\min_x \|Hx - y\|_2^2 + \mu \hat{R}(x).
\]

Due to the definition of the extended bit-cost evaluation function, \( \bar{R}(x) \), the solution candidates of (6.10) are limited to the discrete set \( S \) as defined in (6.7).

Examining the complexity-regularized restoration in (6.10) for the Gaussian denoising task, where \( H = I \), shows that the optimization reduces to the regular rate-distortion optimization in (6.6), namely, the compression of the noisy signal \( y \). However, for more complicated restoration problems, where \( H \) has an arbitrary structure, the optimization in (6.10) is not easy to solve and, in particular, it does not correspond to standard compression designs that are optimized for the regular squared-error distortion metric.

### 6.3 Proposed Methods

In this section we present three restoration methods leveraging a given compression technique. The proposed algorithms result from two different definitions for the complexity regularization function. While the first approach regularizes the total bit-cost of the non-overlapping blocks of the restored signal, the other two refer to the total bit-cost of all the overlapping blocks of the estimate.

#### 6.3.1 Regularize Total Complexity of Non-Overlapping Blocks

Here we establish a practical method addressing the optimization problem in (6.10) based on the alternating direction method of multipliers (ADMM) approach [15] (for additional uses see, e.g., [17, 18, 64, 19, 16]). The optimization (6.10) can be expressed also as

\[
\hat{x} = \arg\min_x \|Hx - y\|_2^2 + \mu \sum_{i \in \mathcal{I}} \bar{R}_b(P_i, x),
\]

where the degradation matrix \( H \), having a general structure, renders a block-based treatment infeasible.

Addressing this structural difficulty using the ADMM strategy [15] begins with introducing the auxiliary variables \( \{z_i\}_{i \in \mathcal{I}} \), where \( z_i \) is coupled with the \( i^{th} \) non-overlapping
Specifically, we reformulate the problem (6.11) into
\[
(\hat{x}, \{\hat{z}_i\}_{i \in \mathcal{I}}) = \arg\min_{x, \{z_i\}_{i \in \mathcal{I}}} \|Hx - y\|_2^2 + \mu \sum_{i \in \mathcal{I}} \bar{R}_b(z_i)
\]
\[
s.t. \quad z_i = P_i x, \quad \text{for} \quad i \in \mathcal{I}.
\]  
(6.12)

Then, reformulating the constrained optimization (6.12) using the augmented Lagrangian (in its scaled form) and the method of multipliers (see [15, Ch. 2]) leads to the following iterative procedure
\[
\left(\hat{x}^{(t)}, \{\hat{z}_i^{(t)}\}_{i \in \mathcal{I}}\right) = \arg\min_{x, \{z_i\}_{i \in \mathcal{I}}} \|Hx - y\|_2^2 + \mu \sum_{i \in \mathcal{I}} \bar{R}_b(z_i)
\]
\[
+ \beta \sum_{i \in \mathcal{I}} \left\|P_i x - z_i + u_i^{(t)}\right\|_2^2
\]  
(6.13)
\[
u_i^{(t+1)} = u_i^{(t)} + \left(P_i \hat{x}^{(t)} - \hat{z}_i^{(t)}\right), \quad i \in \mathcal{I},
\]  
(6.14)

where \(t\) is the iteration number, \(\beta\) is a parameter originating in the augmented Lagrangian, and \(u_i^{(t)} \in \mathbb{R}^{N_b}\) is the scaled dual variable corresponding to the \(i^{th}\) block (where \(i \in \mathcal{I}\)).

Each of the optimization variables in (6.13) participates only in part of the terms of the cost function and, therefore, employing one iteration of alternating minimization (see [15, Ch. 2]) leads to the ADMM form of the problem, where the included optimizations are relatively simple. Accordingly, the \(t^{th}\) iteration of the proposed iterative solution is
\[
\hat{x}^{(t)} = \arg\min_x \|Hx - y\|_2^2 + \frac{\beta}{2} \sum_{i \in \mathcal{I}} \left\|P_i x - \hat{z}_i^{(t)}\right\|_2^2
\]  
(6.15)
\[
\hat{z}_i^{(t)} = \arg\min_{z_i} \frac{\beta}{2} \left\|\hat{x}_i^{(t)} - z_i\right\|_2^2 + \mu \bar{R}_b(z_i), \quad i \in \mathcal{I}
\]  
(6.16)
\[
u_i^{(t+1)} = u_i^{(t)} + \left(P_i \hat{x}^{(t)} - \hat{z}_i^{(t)}\right), \quad i \in \mathcal{I},
\]  
(6.17)

where \(\hat{z}_i^{(t)} \triangleq \hat{z}_i^{(t-1)} - u_i^{(t)}\) and \(\hat{x}_i^{(t)} \triangleq P_i \hat{x}^{(t)} + u_i^{(t)}\) for \(i \in \mathcal{I}\).

The analytic solution of the first stage optimization in (6.15) is
\[
\hat{x}^{(t)} = \left(H^T H + \frac{\beta}{2} I\right)^{-1} \left(H^T y + \frac{\beta}{2} \sum_{i \in \mathcal{I}} P_i^T \hat{z}_i^{(t)}\right)
\]  
(6.18)

rendering this stage as a weighted averaging of the deteriorated signal with the block estimates obtained in the second stage of the previous iteration. While the analytic solution (6.18) explains the underlying meaning of the \(\ell_2\)-constrained deconvolution stage (6.15), it includes matrix inversion that, in general, may lead to numerical instabilities. Accordingly, in the implementation of the proposed method we suggest to address (6.15) via numerical optimization techniques (for example, we used the biconjugate gradients...
The optimizations in the second stage of each iteration (6.16) are rate-distortion optimizations corresponding to each of the non-overlapping blocks of the signal estimate \( \hat{x}^{(t)} \) obtained in the first stage. Accordingly, the set of block-level optimizations in (6.16) can be interpreted as a single full-signal rate-distortion optimization with respect to a Lagrange multiplier value of \( \lambda = 2\beta \). We denote the compression-decompression procedure that replaces (6.16) as

\[
\check{z}^{(t)} = \text{CompressDecompress}_\lambda \left( \hat{x}^{(t)} \right),
\]

where \( \check{x}^{(t)} \triangleq \sum_{i \in \mathcal{I}} P_i^T \hat{x}^{(t)}_i \) is the signal to compress, assembled from all the non-overlapping blocks, and \( \check{z}^{(t)} \) is the corresponding decompressed full signal. Moreover, by defining a full-sized scaled dual variable \( u^{(t)} \equiv \sum_{i \in \mathcal{I}} P_i^T u^{(t)}_i \) we get that \( \check{x}^{(t)} = \hat{x}^{(t)} + u^{(t)} \). Then, using the definitions established here we can translate the block-level computations (6.15)-(6.17) into the full-signal formulations described in Algorithm 6.1.

We further suggest using a standardized compression method as the compression-decompression operator (6.19). While many compression methods do not follow the exact rate-distortion optimizations we have in our mathematical development, we still encourage utilizing such techniques as an approximation for (6.16). Additionally, since many compression methods do not rely on Lagrangian optimization, their operating parameters may have different definitions such as quality parameters, compression ratios, or output bit-rates. Accordingly, we present the suggested algorithm with respect to a general compression-decompression procedure with output bit-cost directly or indirectly affected by a parameter denoted as \( \theta \). These generalizations are also implemented in the proposed Algorithm 6.1. In Section 6.5 we elaborate on particular settings of \( \theta \) that were empirically found appropriate for utilization of the HEVC and the JPEG2000 standard. In cases where the compression method significantly deviates from a Lagrangian optimization form, it can be useful to appropriately update the compression parameter in each iteration (this is the case for JPEG2000 as explained in Section 6.5).

Importantly, Algorithm 6.1 does not only restore the deteriorated input image, but also provides the signal estimate in a compressed form by employing the output of the compression stage of the last iteration.

### 6.3.2 Regularize Total Complexity of All Overlapping Blocks

Algorithm 6.1 emerged from complexity regularization measuring the total bit-cost of the estimate based on its decomposition into non-overlapping blocks (see Eq. (6.11)), resulting in a restored signal available in a compressed form compatible with the compression technique in use. Obviously, the above approach provides estimates limited to the discrete set of signals supported by the compression architecture, thus, having
Algorithm 6.1 Proposed Method Based on Total Complexity of Non-Overlapping Blocks

1: Inputs: $y$, $\beta$, $\theta$.
2: Initialize $\hat{z}^{(0)}$ (depending on the deterioration type).
3: $t = 1$ and $u^{(1)} = 0$
4: repeat
5: $\tilde{z}^{(t)} = \hat{z}^{(t-1)} - u^{(t)}$
6: Solve the $\ell_2$-constrained deconvolution: $\hat{x}^{(t)} = \arg\min_x \|Hx - y\|_2^2 + \frac{\beta}{2}\|x - \tilde{z}^{(t)}\|_2^2$
7: $\tilde{x}^{(t)} = \hat{x}^{(t)} + u^{(t)}$
8: $\tilde{z}^{(t)} = \text{CompressDecompress}_\theta(\tilde{x}^{(t)})$
9: $u^{(t+1)} = u^{(t)} + (\hat{x}^{(t)} - \tilde{z}^{(t)})$
10: $t \leftarrow t + 1$
11: until stopping criterion is satisfied

A somewhat reduced restoration ability with respect to methods providing estimates from an unrestricted domain of solutions. This observation motivates us to develop a complexity-regularized restoration procedure that provides good estimates from the continuous unrestricted domain of signals while still utilizing a standardized compression technique as its main component.

As before, our developments refer to a general block-based compression method relying on a codebook $S_b$ as a discrete set of block reconstruction candidates. We consider here the segmentation of the signal-block space, $\mathbb{R}^{N_b}$, given by the voronoi cells corresponding to the compression reconstruction candidates, namely, for each $c \in S_b$ there is a region

$$V_c \triangleq \left\{ w \in \mathbb{R}^{N_b} \mid c = \arg\min_{\tilde{c} \in S_b} \|w - \tilde{c}\|_2^2 \right\}$$

(6.20)

defining all the vectors in $\mathbb{R}^{N_b}$ that $c$ is their nearest member of $S_b$. We use the voronoi cells in (6.20) for defining an alternative extension to the bit-cost evaluation of a signal block (i.e., the new definition, $\tilde{R}_b^v(z)$, will replace $\bar{R}_b(z)$ given in (6.8) that was used for the development of Algorithm 6.1). Specifically, we associate a finite bit-cost to any $z \in \mathbb{R}^{N_b}$ based on the voronoi cell it resides in, i.e.,

$$\tilde{R}_b^v(z) = R_b(c) \text{ for } z \in V_c$$

(6.21)

where $R_b(c)$ is the regular bit-cost evaluation defined in Section 6.2.2 only for blocks in $S_b$.

The method proposed here emerges from a new complexity regularization function that quantifies the total complexity of all the overlapping blocks of the estimate. Using the extended bit-cost measure $\tilde{R}_b^v(\cdot)$, defined in (6.21), we introduce the full-signal
regularizer as
\[ s^*(x) = \sum_{i \in \mathcal{I}^*} \bar{r}_v(P_i x) \]  
(6.22)

where \( x \in \mathbb{R}^N \), and \( \mathcal{I}^* \) is a set containing the indices of all the overlapping blocks of the signal. The associated restoration optimization is
\[ \hat{x} = \arg\min_x \|Hx - y\|^2_2 + \mu \sum_{i \in \mathcal{I}^*} \bar{R}^v_i(P_i x). \]  
(6.23)

Importantly, in contrast to the previous subsection, the function \( s^*(x) \) evaluates the complexity of any \( x \in \mathbb{R}^N \) with a finite value and, thus, does not restrict the restoration to the discrete set of codebook-based constructions, \( \mathcal{S} \), defined in (6.7).

While the new regularizer in (6.23) is not separable into complexity evaluation of non-overlapping blocks, the ADMM approach can accommodate it as well. This is explained next. We define the auxiliary variables \( \{z_i\}_{i \in \mathcal{I}^*} \), where each \( z_i \) is coupled with the \( i \)th overlapping block. Then, the optimization (6.23) is expressed as
\[ (\hat{x}, \{\hat{z}_i\}_{i \in \mathcal{I}^*}) = \arg\min_{x, \{z_i\}_{i \in \mathcal{I}^*}} \|Hx - y\|^2_2 + \mu \sum_{i \in \mathcal{I}^*} \bar{R}^v_i(z_i) \]
\[ \text{s.t.} \quad z_i = P_i x, \quad \text{for} \ i \in \mathcal{I}^*. \]  
(6.24)

As in Section 6.3.1, employing the augmented Lagrangian (in its scaled form) and the method of multipliers results in an iterative solution provided by the following three steps in each iteration (as before \( t \) denotes the iteration number):
\[ \hat{x}^{(t)} = \arg\min_x \|Hx - y\|^2_2 + \beta \frac{1}{2} \sum_{i \in \mathcal{I}^*} \|P_i x - \hat{z}_i^{(t)}\|^2_2 \]  
(6.25)
\[ \hat{z}_i^{(t)} = \arg\min_{z_i} \beta \frac{1}{2} \|\hat{x}_i^{(t)} - z_i\|^2_2 + \mu \bar{R}^v_i(z_i), \quad i \in \mathcal{I}^* \]  
(6.26)
\[ u_i^{(t+1)} = u_i^{(t)} + (P_i \hat{x}^{(t)} - \hat{z}_i^{(t)}), \quad i \in \mathcal{I}^*, \]  
(6.27)

where \( u_i^{(t)} \) is the scaled dual variable for the \( i \)th block, \( \hat{z}_i^{(t)} \triangleq \hat{z}_i^{(t-1)} - u_i^{(t)} \) and \( \hat{x}_i^{(t)} \triangleq P_i \hat{x}^{(t)} + u_i^{(t)} \) for \( i \in \mathcal{I}^* \).

While the procedure above resembles the one from the former subsection, the treatment of overlapping blocks has different interpretations to the optimizations in (6.25) and (6.26). Indeed, note that the block-level rate-distortion optimizations in (6.26) are not discrete due to the extended bit-cost evaluation \( \bar{R}_v^v(\cdot) \) defined in (6.21). Due to the definition of \( \bar{R}_v^v(\cdot) \), the rate-distortion optimizations (6.26) can be considered as continuous relaxations of the discrete optimizations done by the practical compression technique. Since we intend using a given compression method without explicit knowledge of its underlying codebook, we cannot construct the voronoi cells defining \( \bar{R}_v^v(\cdot) \) and, thus, it is impractical to accurately solve (6.26). Consequently, we suggest to approximate
the optimizations (6.26) by the discrete forms of

$$\hat{z}^{(t)}_i = \arg\min_{z_i} \frac{\beta}{2} \|P^T i \hat{x}^{(t)} - z_i\|_2^2 + \mu \tilde{R}_b(z_i), \ i \in \mathcal{I}$$ \hspace{1cm} (6.28)

where $\tilde{R}_b(\cdot)$ is the discrete evaluation of the block bit-cost, defined in (6.8), letting to identify the problems as operational rate-distortion optimizations of the regular discrete form.

Each block-level rate-distortion optimization in (6.28) is associated with one of the overlapping blocks of the signal. Accordingly, we interpret this group of optimizations as multiple applications of a full-signal compression-decompression procedure, each associates to a shifted version of the signal (corresponding to different sets of non-overlapping blocks). Specifically, for a signal $x$ and a compression block-size of $N_b$ samples, there are $N_b$ shifted grids of non-overlapping blocks. For mathematical convenience, we consider here cyclic shifts such that the $j^{th}$ shift ($j = 1, ..., N_b$) corresponds to a signal of $N$ samples taken cyclically starting at the $j^{th}$ sample of $x$ (in practice other definitions of shifts may be used, e.g., see Section 6.5 for a suggested treatment of two-dimensional signals). We denote the $j^{th}$ shifted signal as $\text{shift}_j \{x\}$. Moreover, we denote the index set of blocks included in the $j^{th}$ shifted signal as $I_j$, hence, $\mathcal{I}^* = \bigcup_{j=1}^{N_b} \mathcal{I}_j$. Therefore, the decompressed blocks $\{\hat{z}^{(t)}_i\}_{i \in I^*}$ can be decomposed into $N_b$ subsets, $\{\hat{z}^{(t)}_i\}_{i \in I_j}$ for $j = 1, ..., N_b$, each contains non-overlapping blocks corresponding to a different shifted grid. Moreover, the $j^{th}$ set of blocks, $\{\hat{z}^{(t)}_i\}_{i \in \mathcal{I}_j}$, is associated with the full signal $\hat{z}_j^{(t)} \triangleq \sum_{i \in \mathcal{I}_j} P^T_i \hat{z}^{(t)}_i$. Then, the set of full signals $\{\hat{z}_j^{(t)}\}_{j=1}^{N_b}$ can be obtained by multiple full-signal compression-decompression applications, namely, for $j = 1, ..., N_b$: $\hat{z}_j^{(t)} = \text{CompressDecompress}_\lambda \left( \hat{x}_j^{(t)} \right)$, where the Lagrangian multiplier value is $\lambda = \frac{2\mu}{\beta}$ and

$$\hat{x}_j^{(t)} \triangleq \text{shift}_j \left\{ \hat{x}^{(t)} + \hat{u}^{(t)} \right\}$$ \hspace{1cm} (6.29)

is the compression input formed as the $j^{th}$ shift of $\hat{x}^{(t)}$ combined with the full-sized dual variable defined via

$$\hat{u}^{(t)} \triangleq \sum_{i \in \mathcal{I}_j} P^T_i \hat{u}^{(t)}_i$$ \hspace{1cm} (6.30)

assembled from the block-level dual variables corresponding to the $j^{th}$ grid of non-overlapping blocks. Notice that inverse shifts are required for obtaining the desired signals, i.e.,

$$\hat{z}_j^{(t)} = \text{shift}_j^{-1} \left\{ \hat{z}_j^{(t)} \right\}$$ \hspace{1cm} (6.31)

where $\text{shift}_j^{-1} \{\cdot\}$ is the inverse shift operator that (cyclically) shifts back the given
The deconvolution stage (6.25) of the iterative process can be rewritten as

$$\hat{x}^{(t)} = \arg\min_x \|Hx - y\|_2^2 + \frac{\beta}{2} \sum_{j=1}^{N_b} \left\|x - \tilde{z}^{j,(t)}\right\|_2^2$$

(6.32)

where the regularization part (the second term) considers the distance of the estimate from the $N_b$ full signals defined via

$$\tilde{z}^{j,(t)} \triangleq \hat{z}^{j,(t)} - u^{j,(t)}$$

(6.33)

for $j = 1, ..., N_b$, where $\hat{z}^{j,(t)}$ and $u^{j,(t)}$ were defined above. The analytic solution of the optimization (6.32) is

$$\hat{x}^{(t)} = \left(H^T H + \frac{\beta}{2} N_b\right)^{-1} \left(H^T y + \frac{\beta}{2} \sum_{j=1}^{N_b} \tilde{z}^{j,(t)}\right),$$

(6.34)

showing that the first stage of each iteration is a weighted averaging of the given deteriorated signal with all the decompressed signals (and the dual variables) obtained in the former iteration. It should be noted that the analytic solution (6.34) is developed here for showing the essence of the $\ell_2$-constrained deconvolution part of the method. Nevertheless, the possible numerical instabilities due to the matrix inversion appearing in (6.34) motivate the practical direct treatment of (6.25) via numerical optimization techniques.

Algorithm 6.2 summarizes the practical restoration method for a compression technique operated by the general parameter $\theta$ for determining the bit-cost (see details in Section 6.3.1).

The computational cost of Algorithm 6.2 stems from its reliance on repeated applications of compressions, decompressions, and $\ell_2$ - constrained deconvolution procedures. While the actual run-time depends on the computational complexity of the utilized compression technique, we can generally state that the total run-time will be of at least the run-time of compression and decompression processes for a total number of applications equal to the product of the number of iterations and the number of shifts considered.

The ADMM is known for promoting distributed optimization structures [15]. In Algorithm 6.2 the distributed nature of the ADMM is expressed in the separate optimization of each of the shifted block-grids (see stages 8-11). In particular, the dual variables $\{u^{j,(t)}\}_{j=1}^{N_b}$ associated with the various grids (see stages 5, 8, and 11 in Algorithm 6.2), are updated independently in stage 11 such that each considers only its respective $\tilde{z}^{j,(t)}$. However, the dual variables $\{u^{j,(t)}\}_{j=1}^{N_b}$ essentially refer to the same data based on different block-grids. Accordingly, we suggest to merge the independent dual variables.
Algorithm 6.2 Proposed Method Based on Total Complexity of All the Overlapping Blocks

1: Inputs: $y, \beta, \theta$.
2: Initialize $\{\hat{z}^{(0)}_j\}_{j=1}^{N_b}$ (depending on the deterioration type).
3: $t = 1$ and $\mathbf{u}^{(1)}_j = \mathbf{0}$ for $j = 1, ..., N_b$.
4: repeat
5: $\hat{z}^{(t)}_j = \hat{z}^{(t-1)}_j - \mathbf{u}^{(t)}_j$ for $j = 1, ..., N_b$
6: Solve the $\ell_2$-constrained deconvolution:
$$\hat{x}^{(t)} = \arg\min_{\mathbf{x}} \|H\mathbf{x} - y\|_2^2 + \frac{\beta}{2} \sum_{j=1}^{N_b} \|\mathbf{x} - \hat{z}^{(t)}_j\|_2^2$$
7: for $j = 1, ..., N_b$ do
8: $\hat{x}^{(t)}_{shifted} = shift_j\{\hat{x}^{(t)}_j + \mathbf{u}^{(t)}_j\}$
9: $\hat{z}^{(t)}_{shifted} = CompressDecompress_\theta(\hat{x}^{(t)}_{shifted})$
10: $\hat{z}^{(t)}_j = shift_{j}^{-1}\{\hat{z}^{(t)}_{shifted}\}$
11: $\mathbf{u}^{(t+1)}_j = \mathbf{u}^{(t)}_j + (\hat{x}^{(t)} - \hat{z}^{(t)}_j)$
12: end for
13: $t \leftarrow t + 1$
14: until stopping criterion is satisfied

to form a single, more robust, dual variable defined as

$$\mathbf{u}^{(t)}_{total} = \frac{1}{N_b} \sum_{j=1}^{N_b} \mathbf{u}^{(t)}_j$$ (6.35)

where the averaging tends to reduce particular artifacts that may appear due to specific block-grids. We utilize the averaged dual variable (6.35) to extend Algorithm 6.2 into Algorithm 6.3. Notice stages 5, 8, and 13 of Algorithm 6.3, where the averaged dual variable is used instead of the independent ones.

In Section 6.5 we further discuss practical aspects of the proposed Algorithms 6.1-6.3 and evaluate their performance for deblurring and inpainting of images.

### 6.4 Rate-Distortion Theoretic Analysis for the Gaussian Case

In this section we theoretically study the complexity-regularized restoration problem from the perspective of rate-distortion theory. While our analysis is focused on the particular settings of a cyclo-stationary Gaussian signal and deterioration caused by a linear shift-invariant operator and additive white Gaussian noise, the results clearly explain the main principles of complexity-regularized restoration.

In general, theoretical studies of rate-distortion problems for the Gaussian case provide to the signal processing practice optimistic beliefs about which design concepts perform well for the real-world non-Gaussian instances of the problems (see the excellent
Algorithm 6.3 Proposed Method Based on Total Complexity of All the Overlapping Blocks with Robust Dual Variables

1: Inputs: \( y, \beta, \theta \).
2: Initialize \( \{ \hat{z}^{j,(0)} \}_{j=1}^{N_b} \) (depending on the deterioration type).
3: \( t = 1 \) and \( u^{(1)}_{\text{total}} = 0 \).
4: \textbf{repeat}
5: \( \hat{z}^{j,(t)} = \hat{z}^{j,(t-1)} - u^{(t)}_{\text{total}} \) for \( j = 1, \ldots, N_b \).
6: Solve the \( \ell_2 \)-constrained deconvolution:
   \( \hat{x}^{(t)} = \arg\min_x \| Hx - y \|_2^2 + \frac{\beta}{2} \sum_{j=1}^{N_b} \| x - \hat{z}^{j,(t)} \|_2^2 \)
7: \textbf{for} \( j = 1, \ldots, N_b \) \textbf{do}
8: \( \tilde{x}^{j,(t)}_{\text{shifted}} = \text{shift}_j \{ \hat{x}^{(t)} + u^{(t)}_{\text{total}} \} \)
9: \( \hat{z}^{j,(t)}_{\text{shifted}} = \text{CompressDecompress}_\theta \left( \tilde{x}^{j,(t)}_{\text{shifted}} \right) \)
10: \( \hat{z}^{j,(t)} = \text{shift}^{-1}_j \left\{ \hat{z}^{j,(t)}_{\text{shifted}} \right\} \)
11: \( u^{j,(t+1)} = u^{j,(t)} + (\hat{x}^{(t)} - \hat{z}^{j,(t)}) \)
12: \textbf{end for}
13: \( u^{(t+1)}_{\text{total}} = \frac{1}{N_b} \sum_{j=1}^{N_b} u^{j,(t+1)} \)
14: \( t \leftarrow t + 1 \)
15: \textbf{until} stopping criterion is satisfied

discussion in [76, Sec. 3]). Moreover, theoretical and practical solutions may embody in a different way the same general concepts. Therefore, one should look for connections between theory and practice in the form of high-level analogies.

The optimal solution presented in this section considers the classical framework of rate-distortion theory and a particular, however, important case of a Gaussian signal and a linear shift-invariant degradation operator. Our rate-distortion analysis below will show that the optimal complexity-regularized restoration consists of the following two main ideas: pseudoinverse filtering of the degraded input, and compression with respect to a squared-error metric that is weighted based on the degradation-filter squared-magnitude (considering a processing in the Discrete Fourier Transform (DFT) domain). In Subsection 6.4.4 we explain how these two concepts connect to more general themes having different realizations in the practical approach proposed in Section 6.3.

In this section, consider the signal \( x \in \mathbb{R}^N \) modeled as a zero-mean cyclo-stationary Gaussian random vector with a circulant autocorrelation matrix \( R_x \), i.e., \( x \sim \mathcal{N}(0, R_x) \). The degradation model studied remains

\[
    y = Hx + n, \quad (6.36)
\]

where here \( H \) is a real-valued \( N \times N \) circulant matrix representing a linear shift-invariant deteriorating operation and \( n \sim \mathcal{N}(0, \sigma_n^2 I) \) is a length \( N \) vector of white Gaussian noise. Clearly, the degraded observation \( y \) is also a zero-mean cyclo-stationary Gaussian
random vector with a circulant autocorrelation matrix \( R_y = HR_xH^* + \sigma^2_nI \).

### 6.4.1 Prevalent Restoration Strategies

We precede the analysis of the complexity-regularized restoration with mentioning three well-known estimation methods. The restoration procedure is a function

\[
\hat{x} = f(y),
\]

where \( f \) maps the degraded signal \( y \) to an estimate of \( x \) denoted as \( \hat{x} \). In practice, one gets a realization of \( y \) denoted here as \( y_r \) and forms the corresponding estimate as \( \hat{x}_r = f(y_r) \).

#### Minimum Mean Squared Error (MMSE) Estimate

This restoration minimizes the expected MSE of the estimate, i.e.,

\[
f_{\text{MMSE}} = \arg\min_f E\left\{ \|x - f(y)\|_2^2 \right\},
\]

yielding that the corresponding estimate is the conditional expectation of \( x \) given \( y \)

\[
\hat{x}_{\text{MMSE}} = f_{\text{MMSE}}(y) = E\{x|y\}. \tag{6.39}
\]

Nicely, for the Gaussian case considered in this section, the MMSE estimate (6.39) reduces to a linear operator, presented below as the Wiener filter.

#### Wiener Filtering

The Wiener filter is also known as the Linear Minimum Mean Squared Error (LMMSE) estimate, corresponding to a restoration function of the form

\[
\hat{x} = f_{\text{Wiener}}(y) = Ay + b,
\]

optimized via

\[
\left\{ \hat{A}, \hat{b} \right\} = \arg\min_{A,b} E\left\{ \|x - (Ay + b)\|_2^2 \right\}. \tag{6.41}
\]

In our case, where \( x \) and \( y \) are zero mean, \( \hat{b} = 0 \) and

\[
\hat{A} = R_xH^* \left( HR_xH^* + \sigma^2_nI \right)^{-1}.
\]

If the distributions are Gaussian, this linear operator coincides with the optimal MMSE estimator.
Constrained Deconvolution Filtering

This approach considers a given degraded signal \( y_r = Hx_0 + n_r \), with the noise vector a realization of a random process while the signal \( x_0 \) is considered as a deterministic vector, with perhaps some known properties. Then, the restoration is carried out by minimizing a carefully-designed penalty function, \( g \), that assumes lower values for \( x \) vectors that fit the prior knowledge on \( x_0 \). Note that for a sufficiently large signal dimension we get that \( \| y_r - Hx_0 \|_2^2 = \| n_r \|_2^2 \approx N\sigma_n^2 \). The last result motivates to constrain the estimate, \( \hat{x} \), to conform with the known degradation model (6.36), by demanding the similarity of \( y_r - H\hat{x} \) to the additive noise term via the equality relation \( \| y_r - H\hat{x} \|_2^2 = N\sigma_n^2 \). The above idea is implemented in an optimization of the form

\[
\min_{\hat{x}} g(\hat{x})
\]

subject to \( \| y_r - H\hat{x} \|_2^2 = N\sigma_n^2 \). (6.43)

Our practical methods presented in Section 6.3 emerge from an instance of the constrained deconvolution optimization (6.43), in its Lagrangian version, where the penalty function \( g \) is the cost in bits measuring the complexity in describing the estimate \( \hat{x} \). In the remainder of this section, we study the complexity-regularized restoration problem from a statistical perspective.

6.4.2 The Complexity-Regularized Restoration Problem and its Equivalent Forms

Based on rate-distortion theory (e.g., see [77]), we consider the estimate of \( x \) as a random vector \( \hat{x} \in \mathbb{R}^N \) with the probability density function (PDF) \( p_{\hat{x}}(\hat{x}) \). The estimate characterization, \( p_{\hat{x}}(\hat{x}) \), is determined by optimizing the conditional PDF \( p_{\hat{x}|y}(\hat{x}|y) \), statistically representing the mapping between the given data \( y \) and the decompression result \( \hat{x} \). Moreover, the rate is measured as the mutual information between \( \hat{x} \) and \( y \), defined via

\[
I(y; \hat{x}) = \int p_{y,\hat{x}}(y, \hat{x}) \log \frac{p_{y,\hat{x}}(y, \hat{x})}{p_y(y) p_{\hat{x}}(\hat{x})} dy d\hat{x}.
\] (6.44)

Then, the basic form of the complexity-regularized restoration optimization is expressed as

**Problem 6.1 (Basic Form),**

\[
\min_{p_{\hat{x}|y}} I(y; \hat{x})
\]

subject to \( \mathbb{E} \left\{ \| y - H\hat{x} \|_2^2 \right\} = N\sigma_n^2 \). (6.45)

Here the estimate rate is minimized while maintaining suitability to the degradation model (6.36) using a distortion constraint set to achieve an a-priori known total noise
energy. In general, Problem 6.1 is complicated to solve since the distortion constraint considers \( \hat{x} \) through the degradation operator \( H \), while the rate is directly evaluated for \( \hat{x} \).

The shift invariant operator \( H \) is a circulant \( N \times N \) matrix, thus, diagonalized by the \( N \times N \) Discrete Fourier Transform (DFT) matrix \( F \). The \((k,l)\) component of the DFT matrix \((k,l=0,\ldots,N-1)\) is \( F_{k,l} = W_k^l \) where \( W_N = \frac{1}{\sqrt{N}} e^{-i2\pi/N} \). Then, the diagonalization of \( H \) is expressed as

\[
FHF^* = \Lambda_H, \tag{6.46}
\]

where \( \Lambda_H \) is a diagonal matrix formed by the components \( h^F_k \) for \( k = 0, \ldots, N-1 \). Using \( \Lambda_H \) we define the pseudoinverse of \( H \) as

\[
H^+ = F^* \Lambda_H^+ F, \tag{6.47}
\]

where \( \Lambda_H^+ \) is the pseudoinverse of \( \Lambda_H \), an \( N \times N \) diagonal matrix with the \( k \)th diagonal element:

\[
h_{k}^{F,+} = \begin{cases} 
\frac{1}{h_k^F}, & \text{for } h_k^F \neq 0 \\
0, & \text{for } h_k^F = 0.
\end{cases} \tag{6.48}
\]

We denote by \( N_H \) the number of nonzero diagonal elements in \( \Lambda_H \), the rank of \( H \).

The first main result of our analysis states that Problem 6.1, being the straightforward formulation for complexity-regularized restoration, is equivalent to the next problem.

**Problem 6.2** (Pseudoinverse-filtered input).

\[
\min_{\hat{p}} \ I(\hat{y};\hat{x}) \tag{6.49}
\]

subject to \( E \left\{ \| \mathbf{H}(\hat{y} - \hat{x}) \|_2^2 \right\} = N_H \sigma_n^2 \),

where

\[
\hat{y} = H^+ y \tag{6.50}
\]

is the pseudoinverse filtered version of the given degraded signal \( y \). One should note that Problem 6.2 has a more convenient form than Problem 6.1 since the distortion is an expected weighted squared error between the two random variables determining the rate. The equivalence of Problems 6.1 and 6.2 is proved in Section 6.6.1.

In this section, \( x \) is a cyclo-stationary Gaussian signal, hence, having a circulant autocorrelation matrix \( R_x \). Consequently, and also because \( H \) is circulant, the deteriorated signal \( y \) is also a cyclo-stationary Gaussian signal. Moreover, \( H^+ \) is also a circulant matrix, thus, by (6.50) the pseudoinverse filtering result, \( \hat{y} \), is also cyclo-stationary and...
zero-mean Gaussian. Specifically, the autocorrelation matrix of $\tilde{y}$ is

$$R_{\tilde{y}} = H^{+} R_y H^{+*} \quad (6.51)$$

$$= H^{+} R_x H^{+*} + \sigma_n^2 H^{+*} H,$$  

(6.52)

and, as a circulant matrix, it is diagonalized by the DFT matrix yielding the eigenvalues

$$\lambda_k(\tilde{y}) = \begin{cases} 
\lambda_k(x) + \frac{\sigma_n^2}{|h_k|^2}, & \text{for } h_k \neq 0 \\
0, & \text{for } h_k = 0.
\end{cases} \quad (6.53)$$

The DFT-domain representation of $\tilde{y}$ is

$$\tilde{y}^F = F\tilde{y}, \quad (6.54)$$

consisted of the coefficients $\{\tilde{y}_k^F\}_{k=0}^{N-1}$, being independent zero-mean Gaussian variables with variances corresponding to the eigenvalues in (6.53).

Transforming Problem 6.2 to the DFT domain, where $\tilde{y}$ becomes a set of independent Gaussian variables to be coded under a joint distortion constraint, simplifies the optimization structure to the following separable form (see proof sketch in Section 6.6.2).

**Problem 6.3** (Separable form in DFT domain).

$$\begin{aligned}
\min_{\{p_{\tilde{y}_k^F}, \tilde{x}_k^F\}} & \sum_{k=0}^{N-1} I(\tilde{y}_k^F; \tilde{x}_k^F) \\
\text{subject to} & \sum_{k=0}^{N-1} |h_k|^2 E\left\{ |\tilde{y}_k^F - \tilde{x}_k^F|^2 \right\} = N_H \sigma_n^2,
\end{aligned} \quad (6.55)$$

where $\{\tilde{x}_k^F\}_{k=0}^{N-1}$ are the elements of $\tilde{x}^F = F\tilde{x}$. Nicely, the separable distortion in Problem 6.3 considers each variable using a squared error that is weighted by the squared magnitude of the corresponding degradation-filter coefficient.

The rate-distortion function of a single Gaussian variable with variance $\sigma^2$ has the known formulation [77]:

$$R(D) = \left[ \frac{1}{2} \log \left( \frac{\sigma^2}{D} \right) \right]_+ \quad (6.56)$$

evaluating the minimal rate for a squared-error allowed reaching up to $D \geq 0$. In addition, the operator $[\cdot]_+$ is defined for real scalars as $[\alpha]_+ \triangleq \max\{\alpha, 0\}$, hence, $R(D) = 0$ for $D \geq \sigma^2$. Accordingly, the rate-distortion function of the Gaussian
variable $\tilde{y}_k^F$ is

$$R_k (D_k) = \left[ \frac{1}{2} \log \left( \frac{\lambda_k^F}{D_k} \right) \right]_+$$

(6.57)

where $D_k$ denotes the maximal squared-error allowed for this component. Now, similar to the famous case of jointly coding independent Gaussian variables with respect to a regular (non-weighted) squared-error distortion \[77\], we explicitly express Problem 6.3 as the following distortion-allocation optimization.

**Problem 6.4 (Explicit distortion allocation).**

$$\min_{D_0, \ldots, D_{N-1}} \sum_{k=0}^{N-1} \left[ \frac{1}{2} \log \left( \frac{\lambda_k^F}{D_k} \right) \right]_+$$

subject to

$$\sum_{k=0}^{N-1} |h_k^F|^2 D_k = N_H \sigma_n^2$$

$$D_k \geq 0 \quad k = 0, \ldots, N - 1.$$  

(6.58)

The optimal distortion-allocation satisfying the last optimization is

$$D_k^{opt} = \begin{cases} \sigma_n^2 / |h_k^F|^2, & \text{for } h_k^F \neq 0 \\ 0, & \text{for } h_k^F = 0 \end{cases}$$

(6.59)

and the associated optimal rates are

$$R_k^{opt} = \begin{cases} \frac{1}{2} \log \left( \frac{|h_k^F|^2 \lambda_k^{(x)}}{\sigma_n^2} + 1 \right), & \text{for } h_k^F \neq 0 \\ 0, & \text{for } h_k^F = 0 \end{cases}$$

(6.60)

Results (6.59) and (6.60) are proved in Section 6.6.3.

### 6.4.3 Demonstration of The Explicit Results

Let us exemplify the optimal rate-distortion results (6.59)-(6.60) for a cyclo-stationary Gaussian signal, $x$, having the circulant autocorrelation matrix presented in Fig. 6.1a, corresponding to the eigenvalues $\{\lambda_k^{(x)}\}_{k=0}^{N-1}$ (Fig. 6.1b) obtained by a DFT-based decomposition. We first examine the denoising problem, where the signal-domain degradation matrix is $H = I$ (Fig. 6.2a) and its respective DFT-domain spectral representation consists of $h_k^F = 1$ for any $k$ (see Fig. 6.2b). The additive white Gaussian noise has a sample variance of $\sigma_n^2 = 5$. Fig. 6.2c exhibits the optimal distortion allocation using a reverse-waterfilling diagram, where the signal-energy distribution $\{\lambda_k^{(x)}\}_{k=0}^{N-1}$ (black solid line) and the additive noise energy (the light-red region) defining together the noisy-signal energy level (purple solid line) corresponding to $\lambda_k^{(\tilde{y})} = \lambda_k^{(x)} + \sigma_n^2$. The blue dashed line in Fig. 6.2c shows the water level associated with the uniform distortion.
Figure 6.1: The autocorrelation of the cyclo-stationary Gaussian signal used in the demonstration. (a) The circulant autocorrelation matrix in the signal domain, and (b) the corresponding eigenvalues obtained using the DFT decomposition.

allocation. The optimal rate-allocation, corresponding to Fig. 6.2c and Eq. (6.60), is presented in Fig. 6.2d showing that more bits are spent on components with higher signal-to-noise ratios.

Another example considers the same Gaussian signal described in Fig. 6.1 and the noise level of $\sigma_n^2 = 5$, but here the degradation operator is the circulant matrix shown in Fig. 6.3a having a DFT-domain representation given in magnitude-levels in Fig. 6.3b exhibiting its frequency attenuation and amplification effects. The waterfilling diagram in Fig. 6.3c includes the same level of signal energy (black solid line) as in the denoising experiment, but the effective additive noise levels and the allocated distortions are clearly modulated in an inversely proportional manner by the squared magnitude of the degradation operator. For instance, frequencies corresponding to degradation-filter magnitudes lower than 1 lead to increase in the effective noise-energy addition and in the allocated distortion. The optimal rate allocation (Fig. 6.3d) is affected by the signal-to-noise ratio and by the squared-magnitude of the degradation filter (see also Eq. (6.60)), e.g., components that are attenuated by the degradation operator get less bits in the rate allocation.

6.4.4 Conceptual Relation to The Proposed Approach

As explained at the beginning of this section, theoretical and practical solutions may include different implementations of the same general ideas. Accordingly, connections between theory and practice should be established by pointing on high-level analogies. Our rate-distortion analysis (for a Gaussian signal and a LSI degradation operator) showed that the optimal complexity-regularized restoration relies on two prominent ideas: pseudoinverse filtering of the degraded input, and compression with respect to a squared-error metric that is weighted based on the degradation-filter squared-magnitude
Figure 6.2: Demonstrating the theoretic results for a denoising problem with a noise level of $\sigma^2_n = 5$. (a) The degenerated degradation operator in the signal domain $\mathbf{H} = \mathbf{I}$. (b) DFT-domain magnitude of the degradation filter. (c) Optimal waterfilling solution in DFT domain. (d) Optimal rate allocation in DFT domain.
Figure 6.3: Demonstrating the theoretic results for a restoration problem with a noise level of $\sigma_n^2 = 5$. (a) The degradation operator in the signal domain $H$. (b) DFT-domain magnitude of the degradation filter. (c) Optimal waterfilling solution in DFT domain. (d) Optimal rate allocation in DFT domain.
(considering the DFT-domain procedure). We will now turn to explain how these two concepts connect to more general themes having different realizations in the practical approach proposed in Section 6.3.

- **Design Concept #1:** Apply simple restoration filtering. The general idea of using an elementary restoration filter is implemented in the Gaussian case as pseudoinverse filtering. Correspondingly, our practical approach relies on a simple filtering mechanism, extending the pseudoinverse filter as explained next. Stage 6 of Algorithm 6.1 is an $\ell_2$-constrained deconvolution filtering that its analytic solution can be rewritten, using the relation $H^* (I - HH^*) = 0$, as (see proof in Section 6.6.4)

$$\hat{x}^{(t)} = \left( H^* H + \frac{\beta}{2} I \right)^{-1} \left( H^* H \tilde{y} + \frac{\beta}{2} \tilde{z}^{(t)} \right).$$

(6.61)

As before, $\tilde{y} = H^+ y$, i.e., the pseudoinverse-filtered version of $y$. The expression (6.61) can be interpreted as an initial pseudoinverse filtering of the degraded input, followed by a simple weighted averaging with $\tilde{z}^{(t)}$ (that includes the decompressed signal obtained in the last iteration). Evidently, the filtering in (6.61) is determined by the $\beta$ value, specifically, for $\beta = 0$ the estimate coincides with the pseudoinverse filtering solution and for a larger $\beta$ it is closer to $\tilde{z}^{(t)}$.

- **Design Concept #2:** Compress by promoting higher quality for signal-components matching to higher $h$-operator magnitudes. This principle is realized in the theoretic Gaussian case as weights attached to the squared-errors of DFT-domain components (see Problems 6.3 and 6.4). Since the weights, $|h_{F0}^2|, |h_{F1}^2|, \ldots, |h_{FN-1}^2|$, are the squared magnitudes of the corresponding degradation-filter coefficients, in the compression of the pseudoinverse-filtered input the distortion is spread unevenly being larger where the degradation filter-magnitude is lower. Remarkably, this concept is implemented differently in the proposed procedure (Algorithm 6.1) where regular compression techniques, optimized for the squared-error distortion measure, are applied on the filtering result of the preceding stage. We will consider the essence of the effective compression corresponding to these two stages together. Let us revisit (6.61), expressing stage 6 of Algorithm 6.1. Assuming $H$ is a circulant matrix, we can transform (6.61) into its Fourier domain representation

$$\hat{x}^{F,(t)} = \left( \Lambda_H^* \Lambda_H + \frac{\beta}{2} I \right)^{-1} \left( \Lambda_H^* \Lambda_H \tilde{y}^{F} + \frac{\beta}{2} \tilde{z}^{F,(t)} \right).$$

(6.62)

where $\hat{x}^{F,(t)}$ and $\tilde{z}^{F,(t)}$ are the Fourier representations of $\hat{x}^{(t)}$ and $\tilde{z}^{(t)}$, respectively.

\[\text{Since the differences between Algorithm 6.1 and Algorithms 6.2 and 6.3 are for a shift-invariance purpose, an issue that we do not concern in this section, we compare our theoretic results only to Algorithm 6.1.}\]
Furthermore, (6.62) reduces to the componentwise formulation

$$\hat{x}_k^{F,(t)} = \frac{|h_k^F|^2 \hat{y}_k^F + \beta \tilde{z}_k^{F,(t)}}{|h_k^F|^2 + \beta}$$

(6.63)

where $\hat{x}_k^{F,(t)}$ and $\tilde{z}_k^{F,(t)}$ are the $k^{th}$ Fourier coefficients of $\hat{x}^{F,(t)}$ and $\tilde{z}^{F,(t)}$, respectively. Equation (6.63) shows that signal elements (of the pseudoinverse-filtered input) corresponding to degradation-filter components of weaker energies will be retracted more closely to the respective components of $\tilde{z}_k^{F,(t)}$ – thus, will be farther from $\hat{y}_k^F$, yielding that the corresponding components in the standard compression applied in the next stage of this iteration will be of a relatively lower quality with respect their matching components of $\hat{y}_k^F$ (as they were already retracted relatively far from them in the preceding deconvolution stage).

To conclude this section, we showed that the main architectural ideas expressed in theory (for the Gaussian case) appear also in our practical procedure. The iterative nature of our methods (Algorithms 6.1-6.3) as well as the desired shift-invariance property provided by Algorithms 6.2-6.3 are outcomes of treating real-world scenarios such as non-Gaussian signals, general linear degradation operators, and computational limitations leading to block-based treatments – these all relate to practical aspects, hence, do not affect the fundamental treatment given in this section.

### 6.5 Experimental Results

In this section we present experimental results for image restoration. Our main study cases include deblurring and inpainting using the image-compression profile of the HEVC standard (in its BPG implementation [50]). We also provide evaluation of our method in conjunction with the JPEG2000 technique for the task of image deblurring.

We empirically found it sufficient to consider only a part of all the shifts, i.e., a portion of $I^*$. When using HEVC, the limited amount of shifts is compensated by the compression architecture that employs inter-block spatial predictions, thus, improves upon methods relying on independent block treatment. The shifts are defined by the rectangular images having their upper-left corner pixel relatively close to the upper-left corner of the full image, and their bottom-right corner pixel coincides with that of the full image. This extends the mathematical developments in Section 6.3 as practical compression handles arbitrarily sized rectangular images.

Many image regularizers have visual interpretation, for example, the classical image-smoothness evaluation. In our framework, the regularization part in (6.10) measures the complexity in terms of the compression bit-cost with respect to a specific compression architecture, designed based on some image model. Our complexity regularization also has a general visual meaning since, commonly, low bit-cost compressed images tend to be overly smooth or piecewise-smooth.
Figure 6.4: Empirical analysis for deblurring the Cameraman image using Algorithm 6.1 (i.e., without overlapping blocks and shifted grids). The parameter settings here are: $\mu = (6.67 \times 10^{-6})$ and $\beta = 0.01$.

Figure 6.5: Empirical analysis for deblurring the Cameraman image using Algorithm 6.2 considering 9 shifted block grids. The parameter settings here are:

$\mu = \frac{(6.67 \times 10^{-6})}{\text{number of shifts}}$ and $\beta = 0.01$.

6.5.1 Image Deblurring

Here we consider two deterioration settings taken from [78]. The first setting, denoted here also as ‘Set. 1’, considers a noise variance $\sigma_n^2 = 2$ coupled with a blur operator defined by the two-dimensional point-spread-function (PSF) $h(x_1, x_2) = 1/(1 + x_1^2 + x_2^2)$ for $x_1, x_2 = -7, ..., 7$, and zero-valued otherwise. The second setting, denoted here also as ‘Set. 2’ (named in [78] as ‘Scenario 3’), considers a noise variance $\sigma_n^2 \approx 0.3$ joint with a blur operator defined by the two-dimensional uniform blur PSF of size $9 \times 9$.

We precede the deblurring experiments with empirical evaluations of four important aspects of the proposed method.

Iterative Reduction of the Fundamental Restoration Cost

In Section 6.3 we established the basic optimization problems for restoration by regularizing the bit-costs of the non-overlapping and the overlapping blocks of the estimate (see (6.11) and (6.23), respectively). As explained above, these two fundamental optimization tasks cannot be directly addressed and, therefore, we developed the ADMM-based
Algorithms 6.1-6.3, that iteratively employ simpler optimization problems. Figures 6.4a and 6.5a demonstrate that, for appropriate parameter settings, the fundamental optimization cost reduces in each iteration. The provided figures also show the fidelity term, $\|Hx - y\|_2^2$, and the regularizing bit-cost (multiplied by $\mu$) of each iteration.

Iterative Improvement of the Restored Image

The fundamental optimization costs in (6.11) and (6.23) include the fidelity term $\|Hx - y\|_2^2$ that considers the candidate estimate $x$ and the given degraded signal $y$. However, the ultimate goal of the restoration process is to produce an estimate $x$ that will be close to the original (unknown!) signal $x_0$. It is common to evaluate proposed methods in experiment settings where $x_0$ is known and used only for the evaluation of the squared error $\|x - x_0\|_2^2$ or its corresponding PSNR. Accordingly, it is a desired property that our iterative methods will provide increment in the PSNR along the iterations and, indeed, Figures 6.4b and 6.5b show that this is achievable for appropriate parameter settings (the use of improper parameters may lead to unwanted decrease of the PSNR starting at some unknown iteration that, however, can be detected in many cases by heuristic divergence rules based on the dual variables used in the ADMM process). Interestingly, for some parameter settings, the PSNR may increase with the iterations, whereas the fundamental restoration cost will not necessarily consistently decrease. The last behavior may result from the fact the our optimization problem (with respect to a standard compression technique) is discrete, non-linear, and usually not convex and, therefore, the convergence guarantees of the ADMM [15] do not hold here for the fundamental restoration cost.

The Optimal Compression Parameter

Another question of practical importance is the value of the parameter $\theta$, determining the compression level of the standard technique utilized in the proposed Algorithms 6.1-6.3. Recall that the ADMM-based developments in Section 6.3 led to an iterative procedure including a stage of Lagrangian rate-distortion optimization operated for a Lagrange multiplier $\lambda \triangleq \frac{2\mu}{\beta}$ and, then, we replaced this optimization with application of a standard compression-decompression technique operated based on a parameter $\theta$. It is clear that $\theta$ is a function of $\lambda$. In the particular case where the standard compression has the Lagrangian form from our developments, then $\theta = \lambda$, however, this is not the general case. For an arbitrary compression technique, we assume that its parameter $\theta$ has $K$ possible values $\theta_1, ..., \theta_K$ (for example, the HEVC standard supports 52 values for its quantization parameter), then, for a given $\lambda \triangleq \frac{2\mu}{\beta}$ the required $\theta$ value in stage 8 of Algorithm 6.1 can be determined via

$$
\theta_{\lambda,\text{opt}}^{(t)} = \arg\min_{\theta \in \{\theta_1, ..., \theta_K\}} \|\tilde{x}^{(t)} - \tilde{z}^{(t)}_{\theta}\|_2^2 + \lambda \tau_{\text{tot},\theta}.
$$

(6.64)
Figure 6.6: Empirical analysis for deblurring the Cameraman image using Algorithm 6.2 with JPEG2000 considering 9 shifted block grids. The compression parameter is the compression ratio given to the JPEG2000 compression. The parameter settings here are: 9 shifts, $\mu = \frac{1.33 \times 10^{-4}}{\text{number of shifts}}$, $\beta = 25 \times 10^{-4}$.

where $\hat{z}_\theta^{(t)} = \text{CompressDecompress}_\theta (\tilde{x}^{(t)})$ is the decompressed signal and $r_{\text{tot},\theta}$ is the associated compression bit-cost. We present here experiments (see Figs. 6.4 and 6.5) for Algorithm 6.1 and 6.2 that in each iteration optimize the $\theta$ value corresponding to $\lambda \equiv \frac{2\mu}{\beta}$ based on procedures similar to (6.64). Nicely, it is shown in Figures 6.4c and 6.5c that, for the HEVC compression used here, the best $\theta$ values along the iterations are nearly the same (for a specific restoration task). This important property may be a result of the fact that HEVC extensively relies on Lagrangian rate-distortion optimizations (although in much more complex forms than those presented in Section 6.3). Accordingly, in order to reduce the computational load, in the experiments shown below we will use a constant compression parameter given as an input to our methods. Interestingly, when examining the optimal compression parameters (compression ratios in this case) for the JPEG2000 method that applies wavelet-based transform coding, there is a decrease in the optimal compression ratio along the iterations (see Fig. 6.6a). Accordingly, in order to reduce the computational load in the experiments below, we employed a predefined rule for reducing the JPEG2000 compression ratio along the iterations. Importantly, when we use the sub-optimal predefined rules for setting the compression parameter $\theta$ values (see Table 6.1 for the settings used in our evaluation comparisons in Table 6.2) we do not longer need to set a value for $\mu$ (since, in these cases, $\mu$ is practically unused).

**Restoration Improvement for Increased Number of Image Shifts**

In our experiments we noticed that the restoration quality improves as more shifts are used, however, at some point the added gain due to the added shifts becomes marginal. As an example to the benefits due to shifts see Figs. 6.4b and 6.5b, where the PSNR obtained for deblurring the Cameraman image using Algorithm 2 and 9 shifts is about
Table 6.1: Parameter Settings Used for the Deblurring and Inpainting Results in Tables 6.2 and 6.3

<table>
<thead>
<tr>
<th>Deblurring Algorithm 2 and 3 with HEVC</th>
<th>Maximal Number of Iterations</th>
<th>Number of Shifts</th>
<th>$\beta$</th>
<th>Compression Parameter $\theta$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>10 for ‘Set. 1’</td>
<td>400</td>
<td>$5 \times 10^{-3}$ for ‘Set. 1’</td>
<td>Fixed on 40</td>
</tr>
<tr>
<td></td>
<td>15 for ‘Set. 2’</td>
<td></td>
<td>$10^{-3}$ for ‘Set. 2’</td>
<td></td>
</tr>
<tr>
<td>Deblurring Algorithm 2 and 3 with JPEG2000</td>
<td>10</td>
<td>3600</td>
<td>$5 \times 10^{-3}$ for ‘Set. 1’</td>
<td>Start at 90, decrease in 10 in each iteration keep fix when arriving to 10</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>$25 \times 10^{-4}$ for ‘Set. 2’</td>
<td></td>
</tr>
<tr>
<td>Inpainting Algorithm 2 and 3 with HEVC</td>
<td>35</td>
<td>400</td>
<td>0.1</td>
<td>Fixed on 35 for Algorithm 2 Fixed on 40 for Algorithm 3</td>
</tr>
<tr>
<td>Inpainting Algorithm 2 with JPEG2000</td>
<td>10 for Algorithm 2</td>
<td>3600</td>
<td>0 for Algorithm 2</td>
<td>For Algorithm 2: Start at 78, decrease in 2 in each iteration keep fix when arriving to 25</td>
</tr>
<tr>
<td></td>
<td>20 for Algorithm 3</td>
<td></td>
<td>0.2 for Algorithm 3</td>
<td>For Algorithm 3: Start at 98, decrease in 2 in each iteration keep fix when arriving to 50</td>
</tr>
</tbody>
</table>

Table 6.2: Deblurring: PSNR Comparison (The Three Best Results in Each Column Appear in Bold Text)

<table>
<thead>
<tr>
<th></th>
<th>Cameraman 256x256</th>
<th>House 256x256</th>
<th>Lena 512x512</th>
<th>Barbara 512x512</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Set. 1</td>
<td>Set. 2</td>
<td>Set. 1</td>
<td>Set. 2</td>
</tr>
<tr>
<td><strong>Input PSNR</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ForWaRD [79]</td>
<td>22.23</td>
<td>20.76</td>
<td>25.61</td>
<td>24.11</td>
</tr>
<tr>
<td>SV-GSM [80]</td>
<td>28.99</td>
<td>28.10</td>
<td>32.96</td>
<td>33.67</td>
</tr>
<tr>
<td>BM3DDEB [81]</td>
<td>30.42</td>
<td>29.10</td>
<td>34.93</td>
<td>34.96</td>
</tr>
<tr>
<td>TVMM [82]</td>
<td>29.64</td>
<td>29.30</td>
<td>33.59</td>
<td>34.50</td>
</tr>
<tr>
<td>CGMK [83]</td>
<td>30.03</td>
<td>29.91</td>
<td>33.92</td>
<td>34.86</td>
</tr>
<tr>
<td>IDD-BM3D [78]</td>
<td>31.08</td>
<td>31.21</td>
<td>35.56</td>
<td>37.00</td>
</tr>
<tr>
<td>EPLL [23]</td>
<td>29.40</td>
<td>29.54</td>
<td>33.88</td>
<td>35.87</td>
</tr>
<tr>
<td>Proposed Algorithm 2 with JPEG2000</td>
<td>28.99</td>
<td>27.48</td>
<td>33.55</td>
<td>33.89</td>
</tr>
<tr>
<td>Proposed Algorithm 3 with JPEG2000</td>
<td>29.28</td>
<td>28.10</td>
<td>33.07</td>
<td>34.01</td>
</tr>
<tr>
<td>Proposed Algorithm 2 with HEVC</td>
<td>30.19</td>
<td>30.20</td>
<td>33.17</td>
<td>36.47</td>
</tr>
<tr>
<td>Proposed Algorithm 3 with HEVC</td>
<td>30.35</td>
<td>30.14</td>
<td>34.37</td>
<td>36.57</td>
</tr>
</tbody>
</table>
2dB higher than the PSNR obtained using Algorithm 1 (i.e., without additional shifts).

In our main evaluation, we examined the proposed Algorithms 2 and 3 for image deblurring in conjunction with the JPEG2000 and HEVC compression techniques (see the parameter settings in Table 6.1). Table 6.2 shows a comparison between various deblurring methods tested in the above two settings for four grayscale images. In 7 out of the 8 cases, the proposed Algorithms 2 or 3 utilizing the HEVC standard provided one of the best three results. Visual results are presented in Figures 6.7 and 6.8.

### 6.5.2 Image Inpainting

We presented in [45] experimental results for the inpainting problem, in its noisy and noiseless settings. Here we focus on the noiseless inpainting problem, where only pixel erasure occurs without an additive noise. The degradation is represented by a diagonal matrix $H$ of $N \times N$ size with main diagonal values of zeros and ones, indicating positions of missing and available pixels, respectively. Then, the product $Hx$ equals to an $N$-length vector where its $k^{th}$ sample is determined by $H$: if $H[k, k] = 0$ then it is zero, and for $H[k, k] = 1$ it equals to the corresponding sample of $x$. The structure of the pixel erasure operator lets us to simplify the optimization in step 6 of Algorithm 6.2. We note that $H$ is a square diagonal matrix and, therefore, $H^T = H$ and $H^T y$ is equivalent to a vector $y$ with zeroed components according to $H$'s structure. Additional useful relation is $H^T H = H$. Consequently, step 6 of Algorithm 6.2 facilitates a componentwise computation that is interpreted to form the $k^{th}$ sample of $\hat{x}^{(t)}$ as

\[
\hat{x}^{(t)}[k] = \begin{cases} 
  y[k] + \beta \sum_{j=1}^{N_b} \tilde{z}^{(t)}[j][k] & \text{for } H[k, k] = 1 \\
  \frac{1}{N_b} \sum_{j=1}^{N_b} \tilde{z}^{(t)}[j][k] & \text{for } H[k, k] = 0 
\end{cases} 
\]  

(6.65)

We initialize the shifted images $\{\tilde{z}^{(0)}[j]\}_{j=1}^{N_b}$ as the given image with the missing pixels set as the corresponding local averages of the available pixels in the respective $7 \times 7$ neighborhoods. When the iterative processing ends, we use the fact that the available pixels are noiseless and set them in the reconstructed image. The rest of the procedure remains as before.

We present here implementations of Algorithms 6.2 and 6.3 utilizing the JPEG2000 and HEVC image compression (the parameter settings are described in Table 6.1). We consider the experimental settings from [84], where 80% of the pixels are missing (see Fig. 6.9b and 6.10b). Five competing inpainting methods are considered: cubic interpolation of missing pixels via Delaunay triangulation (using Matlab’s ‘griddata’ function); inpainting using sparse representations of patches of $16 \times 16$ pixels based on an overcomplete DCT (ODCT) dictionary (see method description in [85, Ch. 15]); using patch-group transformation [86]; based on patch clustering [87]; and via patch reordering.

---

2The results in Table 6.2 for the methods from [79, 80, 81, 82, 83, 78] were taken as is from [78].
Figure 6.7: The deblurring experiment (settings #2) for the Cameraman image (256 × 256). (a) The underlying image. (b) Degraded image (20.76 dB). (c) Restored image using Algorithm 3 with JPEG2000 compression (28.10 dB). (d) Restored image using Algorithm 3 with HEVC compression (30.14 dB).
Figure 6.8: The deblurring experiment (settings #2) for the Barbara image (512 × 512).
(a) The underlying image. (b) Degraded image (22.49 dB). (c) Restored image using Algorithm 3 with JPEG2000 compression (25.18 dB). (d) Restored image using Algorithm 3 with HEVC compression (27.72 dB).
The PSNR values of images restored using the above methods (taken from [84]) are provided in Table 6.3 together with our results. For two images our HEVC-based implementation of Algorithm 3 provides the highest PSNR values. Visually, Figures 6.9d and 6.10d exhibit the effectiveness of our method in repairing the vast amount of absent pixels.

### 6.6 Proofs for the Rate-Distortion Theoretic Analysis

#### 6.6.1 Equivalence of Problems 6.1 and 6.2

We start by showing the equality between the distortion constraints of Problems 6.1 and 6.2. We develop the distortion of Problem 6.1 as follows:

\[
\|y - H\hat{x}\|_2^2 = \|(I - HH^+) y + HH^+ y - H\hat{x}\|_2^2 \\
= \|(I - HH^+) y + H(H^+ y - \hat{x})\|_2^2 \\
= \|(I - HH^+) y\|_2^2 + \|H(H^+ y - \hat{x})\|_2^2 \\
+ (H^+ y - \hat{x})^* H^* (I - HH^+) y \\
+ y^* (I - HH^+)^* H (H^+ y - \hat{x}) \\
= \|(I - HH^+) y\|_2^2 + \|H(H^+ y - \hat{x})\|_2^2
\]  

(6.66)

where the last equality follows from

\[
H^* (I - HH^+) = 0
\]

(6.67)

that can be easily proved, e.g., by using the DFT-based diagonalization of \(H\) and \(H^+\).
Figure 6.9: The inpainting experiment (80% missing pixels) for the Barbara image (512 × 512). (a) The original image. (b) Deteriorated image. (c) Restored image using Algorithm 3 with JPEG2000 compression (24.83 dB) (d) Restored image using Algorithm 3 with HEVC compression (28.80 dB).
Figure 6.10: The inpainting experiment (80% missing pixels) for the House image (256 × 256). (a) The original image. (b) Deteriorated image. (c) Restored image using Algorithm 3 with JPEG2000 compression (30.50 dB) (d) Restored image using Algorithm 3 with HEVC compression (33.10 dB).
The first term in (6.66) can be further developed:

\[
\| (I - HH^+) y \|^2_2 = \| (I - F^* \Lambda_H FF^* \Lambda_H^+ F) y \|^2_2 \\
= \| F^* (I - \Lambda_H \Lambda_H^+) F y \|^2_2 \\
= \| (I - \Lambda_H \Lambda_H^+) y_F \|^2_2 \\
= \sum_{k:h_k^F=0} |y_k^F|^2 \\
= \sum_{k:h_k^F=0} |n_k^F|^2
\]  

where \( y_F \triangleq F y \) is the DFT-domain representation of \( y \) (correspondingly, we use these notations to any vector), and the last equality is implied from the DFT-component relation \( y_k^F = h_k^F x_k^F + n_k^F \) that reduces to \( y_k^F = n_k^F \) for components with \( h_k^F = 0 \). Consequently,

\[
E \left\{ \| (I - HH^+) y \|^2_2 \right\} = E \left\{ \sum_{k:h_k^F=0} |n_k^F|^2 \right\} \\
= (N - N_H) \sigma_n^2
\]  

where \( N_H \) was defined in Section 6.4.2 as the rank of \( H \). Accordingly, and also using (6.66), the distortion constraint of Problem 6.1, i.e.,

\[
E \left\{ \| y - H \hat{x} \|^2_2 \right\} = N \sigma_n^2
\]  

equals to (recall that \( \tilde{y} = H^+ y \))

\[
E \left\{ \| H (\tilde{y} - \hat{x}) \|^2_2 \right\} = N_H \sigma_n^2,
\]

that is, the distortion constraint of Problem 6.2.

We now turn to prove the equivalence of Problems 6.1 and 6.2. Our proof sketch conforms with common arguments in rate-distortion function proofs (see [77]): first, we lower bound the mutual information \( I(y; \hat{x}) \), which is the cost function of Problem 6.1; then, we provide a statistical construction achieving the lower bound while obeying the distortion constraint.

The proposed lower bound for \( I(y; \hat{x}) \) is established by noting that \( \tilde{y} = H^+ y \) and, therefore, the data processing inequality [77] implies here that

\[
I(y; \hat{x}) \geq I(\tilde{y}; \hat{x})
\]

where \( I(\tilde{y}; \hat{x}) \) is the cost function of Problem 6.2. The relation in (6.72) is known to be attained with equality when \( y \) and \( \hat{x} \) are independent given \( \tilde{y} \). The next construction shows that this is indeed the case.
We will now show the achievability of the lower bound in (6.72) by describing a two-stage setting that statistically represents $\tilde{y}$ as an outcome of $\hat{x}$, and $y$ as a consequence of $\tilde{y}$. This layout is an instance of the construction concept known as the (backward) test channel [77]. The first stage of our construction is based on

$$\hat{x} \sim \mathcal{N}(0, \mathbf{H}^+ \mathbf{R}_y \mathbf{H}^{++} - \sigma_n^2 \mathbf{H}^+ \mathbf{H}^{++})$$  \hspace{1cm} (6.73)

$$z \sim \mathcal{N}(0, \sigma_n^2 \mathbf{H}^+ \mathbf{H}^{++})$$  \hspace{1cm} (6.74)

where $\hat{x}$ and $z$ are independent. Consequently, we define

$$\tilde{y} = \hat{x} + z,$$  \hspace{1cm} (6.75)

implying $\tilde{y} \sim \mathcal{N}(0, \mathbf{H}^+ \mathbf{R}_y \mathbf{H}^{++})$ that, indeed, conforms with $\tilde{y} = \mathbf{H}^+ y$ where $y \sim \mathcal{N}(0, \mathbf{R}_y)$. Moreover, the construction (6.73)-(6.75) yields

$$E\left\{ \| \mathbf{H} (\tilde{y} - \hat{x}) \|_2^2 \right\} = E\left\{ \| z \|_2^2 \right\}$$

$$= E\{ z^* \mathbf{H}^* \mathbf{H} z \}$$

$$= E\{ \text{Trace} \{ z^* \mathbf{H}^* \mathbf{H} z \} \}$$

$$= E\{ \text{Trace} \{ \mathbf{H} z z^* \mathbf{H}^* \} \}$$

$$= \text{Trace} \{ \mathbf{H} \mathbf{R}_y \mathbf{H}^* \}$$

$$= \sigma_n^2 \cdot \text{Trace} \{ \mathbf{H} \mathbf{H}^+ \mathbf{H}^{++} \}$$

$$= \sigma_n^2 \cdot \text{Trace} \{ \mathbf{P}_H \mathbf{P}_H^* \}$$

$$= \sigma_n^2 N_H$$  \hspace{1cm} (6.76)

where $\mathbf{P}_H \triangleq \mathbf{H} \mathbf{H}^+$ is the matrix projecting onto the range of $\mathbf{H}$, note it is also a circulant matrix diagonalized by the DFT matrix to the diagonal matrix $\mathbf{A}_{\mathbf{P}_H} \triangleq \mathbf{A}_H \mathbf{A}_H^*$. The last computation of the trace is due to the structure of $\mathbf{A}_{\mathbf{P}_H}$, having ones in main-diagonal entries corresponding to the DFT-domain indices of the range of $\mathbf{H}$, and zeros elsewhere. The result in (6.76) shows that the distortion constraint (6.71) is satisfied.

Let us consider the second stage of the construction, awaiting to prove that $y$ and $\hat{x}$ are independent given $\tilde{y}$. We precede the construction with examining the following decomposition of $y$

$$y = \mathbf{P}_H y + (\mathbf{I} - \mathbf{P}_H) y$$

$$= \mathbf{H} \mathbf{H}^+ y + (\mathbf{I} - \mathbf{P}_H) (\mathbf{H} x + \mathbf{n})$$

$$= \mathbf{H} \mathbf{y} + (\mathbf{I} - \mathbf{P}_H) \mathbf{n}$$  \hspace{1cm} (6.77)

where the second equality uses the degradation model, and the third equality is due to $(\mathbf{I} - \mathbf{P}_H) \mathbf{H} = \mathbf{0}$. Importantly, Eq. (6.77) describes $y$ as a linear combination of two
independent random vectors: \( \tilde{y} \) and \( (I - P_H)n \). Since \( \tilde{y} \) and \( (I - P_H)n \) are Gaussian random vectors, their independence is proved by showing they are uncorrelated via

\[
E \{ (I - P_H)n\tilde{y}^* \} = E \{ (I - P_H)ny^*H^+ \}
\]

\[
= E \{ (I - F^*A_{P_H}F) ny^*F^*A_{P_H}^+F \}
\]

\[
= E \{ F^* (I - A_{P_H}) Fn (Fy)^* A_{P_H}^+F \}
\]

\[
= F^* E \{ (I - A_{P_H}) n^F (A_{P_H}^+y^F)^* \} F
\]

(6.78)

\[
= F^*0F = 0
\]

where in (6.78) we used the facts that \( (I - \Lambda P_H)\) \( n^F \) is a DFT-domain vector with zeros in components corresponding to the range of \( H \), and \( A_{P_H}^+y^F \) is a DFT-domain vector with zeros in entries corresponding to the nullspace of \( H \), hence, these zero patterns yield the outer-product matrix which is all zeros.

The decomposition in (6.77) motivates us to consider \( y \) to emerge from \( \tilde{y} \) via the statistical relation

\[
y = H\tilde{y} + w
\]

(6.79)

where \( w \sim \mathcal{N}(0, \sigma_n^2 (I - P_H)(I - P_H)^*) \) and is independent of \( \tilde{y} \) (and also of \( z \)). Note that \( w \) takes the role of \( (I - P_H)n \) appearing in (6.77), e.g., they have the same distribution. One can further examine the suitability of the construction (6.79) to the considered problem by noting it satisfies the distortion constraint of Problem 6.1, namely,

\[
E \left\{ \|y - H\hat{x}\|^2 \right\} = E \left\{ \|H\tilde{y} + w - H\hat{x}\|^2 \right\}
\]

(6.80)

\[
= E \left\{ \|H(\tilde{y} + z) + w - H\hat{x}\|^2 \right\}
\]

\[
= E \left\{ \|Hz + w\|^2 \right\}
\]

\[
= E \left\{ \|Hz\|^2 \right\} + E \left\{ \|w\|^2 \right\}
\]

\[
= \sigma_n^2 \text{Trace} \{HH^+H^+H^* \}
\]

\[
+ \sigma_n^2 \text{Trace} \{(I - P_H)(I - P_H)^* \}
\]

\[
= \sigma_n^2 \text{Trace} \{P_HP_H^* \}
\]

\[
+ \sigma_n^2 \text{Trace} \{(I - P_H)(I - P_H)^* \}
\]

\[
= \sigma_n^2 \cdot N_H + \sigma_n^2 \cdot (N - N_H)
\]

\[
= N\sigma_n^2
\]

as required. Furthermore, the constructed \( y \) indeed obeys \( y \sim \mathcal{N}(0, R_y) \). Specifically,
its autocorrelation matrix stems from the calculation

\[
E \{yy^*\} = E \{(H\hat{y} + w)(H\hat{y} + w)^*\} = E \{(H\hat{y}) (H\hat{y})^*\} + E \{ww^*\} = HH^*R_yH^{++}H^* + R_w
\]

\[
= P_H (HR_xH^* + \sigma_n^2 I) P_H^* + R_w
\]

\[
= P_H HR_xH^*P_H^* + \sigma_n^2 P_H^* P_H + R_w
\]

\[
= HR_xH^* + \sigma_n^2 P_H^* P_H + \sigma_n^2 (I - P_H) (I - P_H)^*
\]

\[
= HR_xH^* + \sigma_n^2 I
\]

\[
= R_y
\]

where we used the auxiliary result

\[
P_H P_H^* + (I - P_H) (I - P_H)^*
\]

\[
= F^* \Lambda P_H \Lambda P_H^* F + F^* (I - \Lambda P_H) (I - \Lambda P_H)^* F
\]

\[
= F^* \Lambda P_H F + F^* (I - \Lambda P_H) F
\]

\[
= I.
\]

Joining the two stages of the construction, presented in (6.75) and (6.79), exhibits \(\hat{x} \rightarrow \hat{y} \rightarrow y\) as a Markov chain and, therefore, \(y\) and \(\hat{x}\) are independent given \(\hat{y}\). This evident construction turns (6.72) into

\[
I (y; \hat{x}) = I (\hat{y}; \hat{x})
\]

that completes proving the equivalence of Problems 6.1 and 6.2.

**6.6.2 Equivalence of Problems 6.2 and 6.3**

The rate-distortion function for a Gaussian source with memory (i.e., correlated components) is usually derived in the Principle Component Analysis (PCA) domain where the components are independent Gaussian variables (see, e.g., [88]). In our case, where the signal is cyclo-stationary, the PCA is obtained using the DFT matrix. As in the usual case,

\[
I (\hat{y}; \hat{x}) = I (\hat{y}^F; \hat{x}^F)
\]

\[
= \sum_{k=0}^{N-1} I (\hat{y}_k^F; \hat{x}_k^F)
\]

where (6.84) emerges from the reversibility of the transformation, and (6.85) is due to the independence of the \(\{\hat{y}_k^F\}_{k=0}^{N-1}\) components [77] and the backward-channel construction (see Section 6.6.1) that can be translated into a form of independent DFT-domain
component-level channels.

The main difference from the well-known rate-distortion analysis is that here, in Problem 6.2, the distortion constraint is not a regular squared error – but a weighted one, that will be developed next. Since DFT is a unitary transformation, its energy preservation property yields

\[ E \left\{ \parallel \mathbf{H} (\mathbf{\tilde{y}} - \mathbf{\hat{x}}) \parallel_2^2 \right\} = E \left\{ \parallel \mathbf{A}_H (\mathbf{\tilde{y}}^F - \mathbf{\hat{x}}^F) \parallel_2^2 \right\} = \sum_{k=0}^{N-1} |h_k^F|^2 E \left\{ |\mathbf{\tilde{y}}_k - \mathbf{\hat{x}}_k^F|^2 \right\} \]  

(6.86)

where the last equality is due to the diagonal structure of \( \mathbf{A}_H \). Hence, we got that the two expected-distortion expressions in Problems 6.2 and 6.3 are equal.

### 6.6.3 Solution of Problem 6.4

For a start, the transition between Problem 6.3 and Problem 6.4 is analogous to the familiar case of jointly coding a set of independent Gaussian variables [77]. Accordingly, and also due to lack of space, we do not elaborate here on this problem-equivalence proof.

Problem 6.4 is compelling as it is a distortion-allocation optimization, where the distortion levels \( \{D_k\}_{k=0}^{N-1} \) are allocated under the joint distortion constraint. We address Problem 6.4 via its Lagrangian form (temporarily ignoring the constraints of non-negative distortions)

\[
\min_{D_0,\ldots,D_{N-1}} \sum_{k=0}^{N-1} \left[ \frac{1}{2} \log \left( \frac{\lambda(\mathbf{\tilde{y}}_k)}{D_k} \right) \right] + \mu \sum_{k=0}^{N-1} |h_k^F|^2 D_k
\]

(6.87)

where \( \mu \geq 0 \) is the Lagrange multiplier. Recalling that some components may correspond to \( h_k^F = 0 \) and, by (6.53), also \( \lambda_k = 0 \) – meaning they are deterministic variables. These deterministic components do not need to be coded (i.e., \( R_k = 0 \)) while still attaining \( D_k = 0 \). Accordingly, the Lagrangian optimization (6.87) is updated into

\[
\min_{\{D_k\}_{k=0}^{N-1}} \sum_{k:h_k^F \neq 0} \left[ \frac{1}{2} \log \left( \frac{\lambda(\mathbf{\tilde{y}}_k) + \frac{\sigma^2_k}{|h_k^F|^2}}{D_k} \right) \right] + \mu \sum_{k:h_k^F \neq 0} |h_k^F|^2 D_k
\]

(6.88)

where we used the expression from (6.53), and assumed that the distortions are small enough such that the operator \( [\cdot]_+ \) can be omitted (a correct assumption as will be later shown). Now, the optimal \( D_k \) value can be determined by equating the respective derivative of the Lagrangian cost to zero, leading to allocated distortion (still as a
function of $\mu$)

$$D_{k}^{opt} = \frac{1}{2 \ln (2) \mu |h_F^k|^2}$$

for $k : h_F^k \neq 0$ \hfill (6.89)

and by setting the $\mu$ satisfying the total distortion constraint from Problem 6.4 we get

$$D_{k}^{opt} = \frac{\sigma_n^2}{|h_F^k|^2}$$

for $k : h_F^k \neq 0$. \hfill (6.90)

Expressing a nonuniform distortion-allocation (for components with nonzero $h_F^k$), being inversely proportional to the weights $\{|h_F^k|^2\}_{k=0}^{N-1}$. One should note that the assumption on small-enough distortions is satisfied as $D_{k}^{opt} \leq \lambda^{(y)}_k$ for any $k$ obeying $h_F^k \neq 0$, and that all the distortions are non-negative as required. The optimal distortions established here (for $k$ where $h_F^k \neq 0$) are set in the rate formula (6.56), providing the optimal rate allocation

$$R_{k}^{opt} = \begin{cases} 
\frac{1}{2} \log \left( |h_F^k|^2 \frac{\lambda^{(y)}}{\sigma_n^2} + 1 \right), & \text{for } h_F^k \neq 0 \\
0, & \text{for } h_F^k = 0.
\end{cases} \hfill (6.91)$$

### 6.6.4 Equivalent Form of Stage 6 of Algorithm 6.1

The analytic solution of Stage 6 of Algorithm 6.1 is considered here with the conjugate-transpose operator, $^*$, extending the regular transpose:

$$\hat{x}^{(t)} = \left( H^*H + \frac{\beta}{2} I \right)^{-1} \left( H^*y + \frac{\beta}{2} \hat{z}^{(t)} \right). \hfill (6.92)$$

We note that

$$H^*y = H^* \left( HH^*y + (I - HH^*) y \right)$$

$$= H^*Hy + H^* (I - HH^*) y$$

$$= H^*Hy$$

where the last equality results from the relation $H^* (I - HH^*) = 0$. Consequently, (6.92) becomes

$$\hat{x}^{(t)} = \left( H^*H + \frac{\beta}{2} I \right)^{-1} \left( H^*Hy + \frac{\beta}{2} \hat{z}^{(t)} \right), \hfill (6.94)$$

which is the form presented in (6.61).
Chapter 7

Compression Postprocessing via Sequential Denoising

In this chapter we propose a novel postprocessing method for reducing artifacts in compressed images. The task is formulated as a regularized inverse problem, that is 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 simplify the inversion step by representing the nonlinear compression-decompression procedure using a linear approximation. Furthermore, we provide a comprehensive mathematical analysis for linear approximation of simplified quantization and transform-coding operations. We demonstrate our approach for image compression and the presented experimental results show impressive gains, improving upon state-of-the-art postprocessing results for leading image compression standards.

7.1 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., [89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 100, 101, 102, 103, 104, 105, 106, 107, 108, 109] 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 [110]: 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 [110]: enhancement of the deteriorated signal by smoothing its artifacts (e.g., [91, 89]), and restoration of the original signal samples (e.g., [103, 94]).

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 [91, 89, 101, 93]), 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. [16] proposed to efficiently solve regularized inverse-problems in image processing using the Alternating Direction Method of Multipliers (ADMM) [15]. Their approach decouples the inversion and the regularization parts of the optimization problem, which is in turn iteratively solved. Venkatakrishnan et al. [17] 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 the Plug-and-Play Priors framework. Furthermore, as denoising algorithms relying on sparse models were found to be highly effective ones (e.g., K-SVD [111, 112], BM3D [113]), we utilize a leading denoiser from this category. The Plug-and-Play Priors framework was proposed for general inverse problems and was specifically demonstrated for reconstruction of tomographic images. The novelty of our work with respect to the original Plug-and-Play approach is that we apply it for the task of compression-artifact
reduction. Moreover, we utilize it to address an inverse-problem for a forward-model that is non-linear and non-differentiable.

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 that, a thorough demonstration for image compression is provided. Specifically, we show results for the leading image compression standards: JPEG [114], JPEG2000 [115] and the still-image profile of the HEVC [29, 116], offering the state-of-the-art performance [117]. 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 chapter is organized as follows. In section 7.2 the proposed postprocessing method is presented. In section 7.3, the compression linearization is mathematically analyzed for simplified cases of quantization and transform coding. Section 7.4 presents image-compression experimental results and compares them to those of competitive techniques.

7.2 The Proposed Postprocessing Strategy

7.2.1 Problem Formulation using ADMM

Let us consider a signal $x \in \mathbb{R}^N$ that undergoes a compression-decompression procedure, $C : \mathbb{R}^N \rightarrow \mathbb{R}^N$, resulting in the reconstructed signal $y = C(x)$. 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 $x$ from the reconstructed $y$ using the following regularized inverse-problem:

$$\hat{x} = \arg \min_x ||y - C(x)||_2^2 + \beta s(x), \quad (7.1)$$

where $s(\cdot)$ is a regularizer, which can be associated with a given Gaussian denoiser, weighted by the parameter $\beta$. For example, assuming that the image is piecewise constant promotes the utilization of the popular total-variation regularizer, $s(x) = ||x||_{TV}$ [118].

One should note that $y$ and $C(x)$ are two signals reconstructed from compression, and therefore, the fidelity term in Equation (7.1) expresses their distance. Notice that
this is substantially different from $\|y - x\|_2^2$—whereas the latter has compression artifacts as error, the one we deploy represents a milder distortion. Throughout this chapter we shall assume for simplicity that the distortion between the two reconstructions, $y$ and $C(x)$, is modeled as a white additive Gaussian noise, leading to the $\ell_2$ term used here. We should note, however, that our scheme could be improved by using a better modeling of the reconstructed-signal error, such as an $\ell_\infty$ on the transform coefficients w.r.t. the quantization step-size (in the case of transform coding).

Similar to [16] and [17], we develop an iterative algorithm for the solution of (7.1). We start by applying variable splitting that yields the following equivalent form of (7.1):

$$
\min_{x,v} \|y - C(x)\|_2^2 + \beta s(v) \quad (7.2)
$$

subject to $x = v$,

where $v \in \mathbb{R}^N$ is an additional vector due to the split. The constrained problem (7.2) is addressed by forming an augmented Lagrangian and its corresponding iterative solution (of its scaled version) via the method of multipliers [15, ch. 2], where the $i^{th}$ iteration consists of

$$(\hat{x}_i, \hat{v}_i) = \arg \min_{x,v} \|y - C(x)\|_2^2 + \beta s(v) + \frac{\lambda}{2} \|x - v + u_i\|_2^2 \quad (7.3)$$

$$u_{i+1} = u_i + (\hat{x}_i - \hat{v}_i). \quad (7.6)$$

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

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

Approximating the joint optimization of $x$ and $v$ in (7.3), using one iteration of alternating minimization, results in the iterative solution in the ADMM form, where the $i^{th}$ iteration consists of

$$\hat{x}_i = \arg \min_x \|y - C(x)\|_2^2 + \frac{\lambda}{2} \|x - \tilde{x}_i\|_2^2 \quad (7.4)$$

$$\hat{v}_i = \arg \min_v \frac{\lambda}{2} \|v - \tilde{v}_i\|_2^2 + \beta s(v) \quad (7.5)$$

$$u_{i+1} = u_i + (\hat{x}_i - \hat{v}_i). \quad (7.6)$$

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

The regularization step (7.5) is of the form of a Gaussian denoising optimization-problem (of a noise level determined by $\beta/\lambda$) and therefore can be viewed as applying a denoising algorithm to the signal $\tilde{v}_i$. More specifically, this corresponds to assuming that $\tilde{v}_i = v + w$, where $w$ is an i.i.d zero-mean Gaussian vector with variance $1/\lambda$ (and a corresponding distribution function denoted as $p_w(w)$). In addition, $v$ is assumed
to be drawn from a distribution $p_s(v)$ that is proportional to $\exp(-\beta s(x))$. Then, the Maximum A-Posteriori (MAP) estimator of $v$ from its (white Gaussian) noisy version $\tilde{v}_i$ is formed as

$$\tilde{v}_i = \arg\max_v \log p_w(\tilde{v}_i - v) + \log p_s(v), \quad (7.7)$$

which for the above defined distribution functions, $p_w(\cdot)$ and $p_s(\cdot)$, is equivalent to (7.5) and, thus, establishes the latter as a Gaussian denoising procedure. Indeed, the Plug-and-Play Priors framework [17] suggests exactly this strategy, replacing (7.5) with an independent denoiser; even one that does not explicitly have in its formulation a minimization problem of the form of (7.5). The deployment of a favorable denoiser introduces valuable practical benefits to the design of the proposed postprocessing procedure, and yields a powerful generic method.

### 7.2.2 Linear Approximation of the Compression-Decompression Procedure

Due to the high nonlinearity of $C(x)$, we further simplify the forward-model step (7.4) using a first-order Taylor approximation of the compression-decompression function around $\tilde{x}_{i-1}$, i.e.,

$$C_{lin}(x) = C(\tilde{x}_{i-1}) + \frac{dC(z)}{dz} \bigg|_{z=\tilde{x}_{i-1}} \cdot (x - \tilde{x}_{i-1}) \quad (7.8)$$

where $\frac{dC(z)}{dz} \bigg|_{z=\tilde{x}_{i-1}}$ is the $N \times N$ Jacobian matrix of the compression-decompression at the point $\tilde{x}_{i-1}$.

Since the approximation of the Jacobian, $\frac{dC(z)}{dz}$, 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.

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 [119].

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:

$$
\frac{dC(z)}{dz_k} = \frac{1}{|S_\delta|} \sum_{\delta \in S_\delta} \frac{C(z + \delta \cdot e_k) - C(z - \delta \cdot e_k)}{2\delta},
$$

(7.9)

where $e_k$ is the $k^{th}$ standard direction vector, and $S_\delta$ 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 (7.8) is reasonable in a small neighborhood around the approximating point $\hat{x}_{i-1}$. Accordingly, we further constrain the distance of the solution from the linear-approximation point by modifying (7.4) to

$$
\hat{x}_i = \arg \min_x \|y - C_{lin}(x)\|_2^2 + \frac{\lambda}{2} \|x - \bar{x}_i\|_2^2 + \mu \|x - \bar{x}_{i-1}\|_2^2.
$$

(7.10)

The proposed generic method is summarized in Algorithm 7.1.

**Algorithm 7.1 The Proposed Postprocessing Method**

1: $\hat{x}_0 = y$, $\hat{v}_0 = y$
2: $i = 1$, $u_1 = 0$
3: repeat
4: Approximate $C_{lin}(\cdot)$ around $\hat{x}_{i-1}$ using (7.8) and (7.9)
5: $\tilde{x}_i = \hat{v}_{i-1} - u_i$
6: $\hat{x}_i = \arg \min_x \|y - C_{lin}(x)\|_2^2 + \frac{\lambda}{2} \|x - \bar{x}_i\|_2^2 + \mu \|x - \bar{x}_{i-1}\|_2^2$
7: $\tilde{v}_i = \hat{x}_i + u_i$
8: $\hat{v}_i = \text{Denoise}_{\beta/\lambda}(\tilde{v}_i)$
9: $u_{i+1} = u_i + (\hat{x}_i - \hat{v}_i)$
10: $i \leftarrow i + 1$
11: until stopping criterion is satisfied

### 7.3 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.

### 7.3.1 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 \( \delta \) as

\[
\eta(x_0, \delta) = [x_0 - \delta, x_0 + \delta].
\]

The studied approximation takes the general linear form of

\[
\tilde{q}(x) = ax + b
\]

where \( a \) and \( b \) are the linearization parameters. The approximation in (7.12) introduces an error that can be measured via the Mean-Squared-Error (MSE) over the local interval \( \eta(x_0, \delta) \):

\[
LMSE^{SQ}(a, b; \eta(x_0, \delta)) \triangleq \frac{1}{2\delta} \int_{x_0-\delta}^{x_0+\delta} (q(x) - \tilde{q}(x))^2 dx
\]

Substituting (7.12) in (7.13), then demanding parameter optimality by

\[
\frac{\partial}{\partial a} LMSE^{SQ}(a, b; \eta(x_0, \delta)) = 0
\]

\[
\frac{\partial}{\partial b} LMSE^{SQ}(a, b; \eta(x_0, \delta)) = 0,
\]

\[\text{1}\] Here we mathematically study the problem for a given quantization function that is used for calculating the approximation, and therefore a single \( \delta \) value is sufficient. However, in our generic algorithm we empirically utilize a set of \( \delta \) values in Equation (7.9) since the compression function is unknown.

\[\text{2}\] We study the linearization error as a function of the approximation-interval size, which is determined by \( \delta \). Clearly, by setting a sufficiently small \( \delta \) 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 \( \delta \). Therefore, we should bear in mind throughout the following derivation that we do not present here an explicit method for selecting the value of \( \delta \) but an analysis of the local approximation-error of the quantizer as a function of \( \delta \). 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.
leads to the following optimal parameters:

\[ a^* = \frac{3}{\delta^2} (L_a - L_b x_0) \]  
(7.15)

\[ b^* = L_b - \frac{3x_0}{\delta^2} (L_a - L_b x_0) \]  
(7.16)

where we defined

\[ L_a \triangleq \frac{1}{2\delta} \int_{x_0-\delta}^{x_0+\delta} x q(x) \, dx \]  
(7.17)

\[ L_b \triangleq \frac{1}{2\delta} \int_{x_0-\delta}^{x_0+\delta} q(x) \, dx. \]  
(7.18)

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

### 7.3.2 The Case of Two-Level Quantization

We start by studying the elementary two-level quantizer that takes the form of a step function (Fig. 7.1a) as follows:

\[ q_2(x) = \begin{cases} 
-\frac{1}{2}, & \text{for } x \leq 0 \\
\frac{1}{2}, & \text{for } x > 0 
\end{cases} \]  
(7.19)

where the two output levels, \( r_0 = -\frac{1}{2} \) and \( r_1 = \frac{1}{2} \), 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 (7.19) 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 \( q_2(x) \) is locally fixed on \( r_0 \) or \( r_1 \), respectively, and therefore

\[ L_a = \frac{1}{2\delta} \int_{x_0-\delta}^{x_0+\delta} x r_i dx = x_0 r_i \]  
(7.20)

\[ L_b = \frac{1}{2\delta} \int_{x_0-\delta}^{x_0+\delta} r_i dx = r_i \]  
(7.21)

for the respective \( i \in \{0, 1\} \). Then setting (7.20) and (7.21) in (7.15) and (7.16), 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 (7.15)-(7.16) for this scenario requires to decompose the integrals (7.17)-(7.18) to the two decision regions, yielding the following optimal linearization parameters:

\[ a^* = \frac{3}{4\delta} \left( 1 - \left(\frac{x_0}{\delta}\right)^2 \right) \]  
\[ b^* = \frac{3x_0}{4\delta} \left( \left(\frac{x_0}{\delta}\right)^2 - \frac{1}{3} \right) \]  

and, using (7.13), the corresponding error (for $\delta > |x_0|$) is

\[ \text{LMSE}_{SQ}(a^*, b^*; \eta(x_0, \delta)) = \frac{1}{16} \left( 1 + 3 \left(\frac{x_0}{\delta}\right)^2 \right) \left( 1 - \left(\frac{x_0}{\delta}\right)^2 \right). \]  

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

\[ \lim_{\delta \to \infty} a^* = 0 \]  
\[ \lim_{\delta \to \infty} b^* = 0. \]  

This asymptotic fitting to a constant-valued function is also expressed in the numerical results in Fig. 7.2a-7.2b.

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 $\delta$ 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 $\delta$. This interesting observation is a special case of a more general behavior where a constant error value is achieved for any $(x_0, \delta)$ pair that is on the line $\delta = c|x_0|$ for some $c \in [0, \infty)$. 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. 7.2.

The numerical results also demonstrate the following behavior of the approximation as function of $\delta$. 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.

\footnote{Recall that for $\delta < |x_0|$ the approximation local-MSE is zero.}
Figure 7.1: Examples of scalar quantizers.

Figure 7.2: Optimal linearization of a normalized two-level quantizer as function of the local interval center ($x_0$) and length ($\delta$).
7.3.3 The Case of Multi-Level Uniform Quantization

Let us extend the above analysis to a multi-level uniform quantizer in the mid-riser form [120, p. 137] (Fig. 7.1b):

\[
q_u(x) = \lfloor x \rfloor + \frac{1}{2},
\]

(7.27)

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.

The optimal local linear approximation for this uniform quantizer is obtained by calculating (7.15)-(7.16) for the formula in (7.27). 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 7.3.2) 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 (7.22)-(7.23). 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:

\[
q_u(x) = \sum_{\tau = -\infty}^{\infty} q_2(x - \tau),
\]

(7.28)

where \(q_2(\cdot)\) was defined in (7.19). Then, using (7.28) we can develop (7.17)-(7.18) to the following forms:

\[
L_u^a = \sum_{\tau = -\infty}^{\infty} L_\tau^a, \quad (7.29)
\]

\[
L_u^b = \sum_{\tau = -\infty}^{\infty} L_\tau^b, \quad (7.30)
\]

where \(L_\tau^a\) and \(L_\tau^b\) 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

\[ a^*_u = \sum_{\tau=-\infty}^{\infty} a^*_\tau \]

(7.31)

\[ b^*_u = \sum_{\tau=-\infty}^{\infty} b^*_\tau, \]

(7.32)

where \( a^*_\tau \) and \( b^*_\tau \) are the optimal linearization parameters for \( q_2(x-\tau) \) and are obtainable by shifting the expressions in (7.22)-(7.23). This analytic relation between the linearization of the uniform and the two-level quantizers is clearly exhibited in the numerical results (see Fig. 7.3) in the form of a periodic structure.

The numerical calculations (Fig. 7.3) also show convergence to the global approximation parameters

\[ \lim_{\delta \to \infty} a^*_u = 1 \]

(7.33)

\[ \lim_{\delta \to \infty} b^*_u = 0, \]

(7.34)

which imply \( \lim_{\delta \to \infty} C_{lin}(x) = x \). In order to explain the results in Fig. 7.3, we return to the interpretation of a multi-level quantizer as a sum of shifted two-level quantizers (as expressed in Eq. (7.28)). 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. 7.3 to Fig. 7.2 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. 7.3 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. 7.3c) 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 [121], 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 \( i = 0, \pm1, \pm2, \ldots \)), where only the two adjacent non-differentiable points affect the linearization. This can also be shown analytically by setting the decompositions in (7.28) and (7.31)-(7.32) into (7.13),
resulting in

\[
LMSE_{SQ}^{\tau}(a^\tau, b^\tau; \eta(x_0, \delta)) = \\
\sum_{\tau = -\infty}^{\infty} LMSE_{SQ}^{\tau}(a^\tau, b^\tau; \eta(x_0, \delta)) \\
+ \frac{1}{2\delta} \sum_{\tau, \nu = -\infty}^{x_0+\delta} \int_{x_0-\delta}^{x_0+\delta} (q_2(x - \tau) - a^\tau_x x - b^\tau_x) \times \\
(q_2(x - \nu) - a^\nu_x x - b^\nu_x) dx,
\]

where \(LMSE_{SQ}^{\tau}(a^\tau, b^\tau; \eta(x_0, \delta))\) is the optimal LMSE for \(q_2(x - \tau)\) as available by shifting the expression in (7.24).

### 7.3.4 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

\[ C(x) = UQ(U^T x), \tag{7.36} \]

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

\[ Q(x) = \begin{bmatrix} q(x_1) \\ \vdots \\ q(x_N) \end{bmatrix} \tag{7.37} \]

where \( q(\cdot) \) is a single-variable scalar quantization function as studied above, and \( x_i \) is the \( i^{th} \) component of the vector \( x \). 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 (7.36), it imposes its non-differentiable nature on \( C(x) \). Let us consider the linear approximation of \( C(x) \) around the point \( x_0 \in \mathbb{R}^N \) in a limited neighborhood of a high-dimensional cube defined by \( \delta \) as

\[ \eta(x_0, \delta) = \{ x \mid \| x - x_0 \|_\infty \leq \delta \}. \tag{7.38} \]

The approximation takes the general multidimensional linear form of

\[ \tilde{C}(x) = Ax + b \tag{7.39} \]

where \( A \in \mathbb{R}^{N \times N} \) and \( b \in \mathbb{R}^N \) are the linearization parameters. The local MSE of approximating the transform-coding procedure around \( x_0 \) is defined as

\[ L\text{MSE}^{TC}(A, b; \eta(x_0, \delta)) \triangleq \frac{1}{|\eta(x_0, \delta)|} \int_{\eta(x_0, \delta)} \| C(x) - \tilde{C}(x) \|_2^2 \, dx \tag{7.40} \]

By substituting (7.39) in (7.40) and using the energy-preservation property of unitary transforms, we get the equivalent error expression in the transform-domain

\[ \frac{1}{|\hat{\eta}(x_0, \delta)|} \int_{\hat{\eta}(x_0, \delta)} \| Q(\hat{x}) - \hat{A}\hat{x} - \hat{b} \|_2^2 \, d\hat{x} \tag{7.41} \]

124
where
\[ \hat{x} = U^T x \quad (7.42) \]
\[ \hat{x}_0 = U^T x_0 \]
\[ \hat{A} = U^T A U \]
\[ \hat{b} = U^T b \]
\[ |\hat{\eta}(x_0, \delta)| = |\eta(x_0, \delta)|. \]

and the rotated approximation area (or volume), \( \hat{\eta} \), is defined around \( \hat{x}_0 \) 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, \( x \), by identical scalar quantization of its components, \( x_i \):

\[
LMSE^{SQV}(A, b; \bar{\eta}) \triangleq \frac{1}{|\bar{\eta}|} \int_{\bar{\eta}} \| Q(x) - Ax - b \|_2^2 dx
= \frac{1}{|\bar{\eta}|} \sum_{i=1}^{N} \int_{\hat{\eta}} (q(x_i) - a_i^T \hat{x} - b_i)^2 dx
\]

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

Equations (7.41) and (7.43) clearly show that the MSE of approximating transform-coding (7.41) reduces to the linearization-error of scalar quantization of the transform coefficients, namely,

\[
LMSE^{TC}(A, b; \eta(x_0, \delta))
= LMSE^{SQV}(\hat{A}, \hat{b}; \hat{\eta}(x_0, \delta))
= \frac{1}{|\hat{\eta}(x_0, \delta)|} \sum_{i=1}^{N} \int_{\hat{\eta}(x_0, \delta)} (q(\hat{x}_i) - \hat{a}_i^T \hat{x} - \hat{b}_i)^2 d\hat{x}
\]

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

While the separability of \( Q(\cdot) \) was utilized to have integrals in (7.44) 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, \( \eta_U(x_0, \delta) \), such that its transform-domain counterpart is aligned with the axes (see Fig. 7.4):

\[
\hat{\eta}_U(x_0, \delta) = \{ \hat{x} \ | \ ||\hat{x} - \hat{x}_0||_\infty \leq \delta \}. \quad (7.45)
\]
Figure 7.4: 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.

Note that $\eta_{U}(x_0, \delta)$ 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 $C(x)$. This is obtainable by finding the optimal transform-domain parameters $\hat{A}^*$ and $\hat{b}^*$ and then transforming them back to the signal domain. Following this strategy we first pose the componentwise optimality demands in the transform domain:

$$\frac{\partial}{\partial \hat{a}_{ij}} \text{LMSE}^{SQV}(\hat{A}, \hat{b}; \hat{\eta}_{U}(x_0, \delta)) = 0 \text{ for } i,j = 1,\ldots,N$$
$$\frac{\partial}{\partial \hat{b}_i} \text{LMSE}^{SQV}(\hat{A}, \hat{b}; \hat{\eta}_{U}(x_0, \delta)) = 0 \text{ for } i = 1,\ldots,N$$

(7.46)

Some calculations show that the solution satisfying the optimality conditions consists of a diagonal matrix $\hat{A}^*$ (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 $\left[\hat{x}_0(i) - \delta, \hat{x}_0(i) + \delta\right]$ as generally given in (7.15)-(7.16). Then, the signal-domain parameters are given as

$$\hat{A}^* = U \hat{A}^* U^T = \sum_{i=1}^{N} \hat{a}_{ii}^* u_i u_i^T$$

(7.47)

$$\hat{b}^* = U \hat{b}^*$$

(7.48)

where the last equality in (7.47) is due to the diagonality of $\hat{A}^*$ and $u_i$ denotes the $i^{th}$ column of $U$. The corresponding optimal error is equivalent in the signal and transform domains.
domains, hence can be expressed in a simplified form as

\[
LMSE^{TC}(A^*, b^*; \eta_U(x_0, \delta)) = LMSE^{SQV}(\hat{A}^*, \hat{b}^*; \hat{\eta}_U(x_0, \delta))
\]

\[
= \frac{1}{2\delta} \sum_{i=1}^{N} \int_{\hat{x}_0(i)-\delta}^{\hat{x}_0(i)+\delta} \left( q(\hat{x}_i) - \hat{a}_{ii}^* \hat{x}_i - \hat{b}_i^* \right)^2 d\hat{x}
\]

\[
= \sum_{i=1}^{N} LMSE^{SQ}(\hat{a}_{ii}^*, \hat{b}_i^*; \eta(\hat{x}_0(i), \delta))
\]

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 \(\hat{x}_0(i)\) located differently with respect to the quantization lattice.

Let us exemplify the latter analysis on a transform coder of two-component signals (i.e., \(x \in \mathbb{R}^2\)), that scalarly applies the normalized two-level quantizer that was studied above (see Eq. (7.19)) on the two components in the domain of the 45°-rotation matrix that takes the 2x2 form of \(U_{\pi/4} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & -1 \\ 1 & 1 \end{bmatrix}\). The approximation is around \(x_0 = U_{\pi/4} \hat{x}_0\) in a 45°-rotated square neighborhood defined by the \(\eta_U(x_0, \delta)\) (see Fig. 7.4). We further define the first component of \(\hat{x}_0\) to vary and fix the second on the value of 15, i.e., \(\hat{x}_0 = \left[ \hat{x}_0^{(1)} \ 15 \right]^T\). The overall linearization error as function of \(\delta\) and \(\hat{x}_0^{(1)}\) (Fig. 7.5) shows that it combines the errors of the scalar linearization of the transform coefficients (Figs. 7.6e-7.6f). The corresponding parameters in the signal domain (where the matrix \(A\) is not necessarily diagonal) are given in Fig. 7.7. 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. 7.5).

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 \(\eta_U(x_0, \delta)\) neighborhood, where \(\hat{A}^*\) is diagonal. Accordingly, in this case, quantization of each transform-coefficient is linearized separately over a one-dimensional interval of size \(2\delta\). 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 \(x_0\) 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 \( 2\delta \) value is still greater than \( \Delta_i \), however relatively closer, hence, the corresponding \( \hat{a}_{ii}^* \) values fluctuate. Finally, the \( i > K \) coefficients have quantization steps that are greater than \( 2\delta \), and accordingly, \( \hat{a}_{ii}^* \approx 0 \). The latter qualitative analysis lets us to interpret the local-approximation of the transform-coder, \( \hat{A}^* \), as a low-pass filter that depends on \( \delta \) and the approximation point. Furthermore, the numerical results (Fig. 7.8) 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 \( \delta \) the filter has a all-stop (Fig. 7.8a) or all-pass (Fig. 7.8d) behavior, respectively.

We conclude by considering the signal-domain filter \( A^* \) related to \( \hat{A}^* \) by the inverse-transformation in (7.47). When the compression utilizes the Discrete Fourier Transform (DFT) and the approximation is over \( \eta_U(x_0, \delta) \), then the diagonal matrix \( \hat{A}^* \) yields a circulant \( A^* \). While the latter involves complex-valued calculations, coding using Discrete Cosine Transform (DCT) keeps the procedure over the reals. The signal-domain filter, \( A^* \), of the DCT-based coding is exemplified in Fig. 7.9 showing an approximately Toeplitz structure.

### 7.4 Experimental Results

In this section we demonstrate the performance of the proposed postprocessing method by presenting results obtained in conjunction with various compression methods. We start by considering the simplistic compression procedures of scalar quantization and
Figure 7.6: 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 \( \hat{A} \), and (c)-(d) show the values of \( \hat{b} \)'s components. (e)-(f) show the corresponding approximation errors of the two-transform domain elements.
Figure 7.7: 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 $A$, and (e)-(f) show the values of $b$'s components.
Figure 7.8: Interpretation of a diagonal $\hat{A}^*$ as a transform-domain filter that depends on $\delta$. Here $N = 32$ and the $i^{th}$ quantization step is $\Delta_i = 2^{i/4}$.

Figure 7.9: Interpretation of $A^*$ as a signal-domain filter that depends on $\delta$. Presented here for the case of $\delta = 50$ from Fig. 7.8c, incorporated in a DCT-based coding.
one-dimensional transform coding. Then, we proceed to the leading image compression standards: JPEG, JPEG2000 and the recent HEVC.

In all experiments we use the BM3D method [113] as the denoiser. Since the proposed technique uses a well established denoiser as a subroutine, we compare our method with a single application of this denoiser as a postprocessing procedure. 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. More specifically, this oracle denoiser optimizes its output PSNR based on the knowledge of the precompressed image, a capability that cannot be applied in a real postprocessing task.

The computational complexity of our method is mainly determined by the complexity levels of the utilized denoiser and the Jacobian estimation procedure. The latter further depends on the implementation of the compression-decompression method, as it is repeatedly applied according to (7.9). In addition, equation (7.9) exhibits also the effect of the size of the set $S_\delta$ utilized for approximating a single column of the Jacobian. Since the number of Jacobian columns is as the number of signal samples (denoted as $N$), a straightforward computation of the Jacobian is costly and requires $N$ calculations of (7.9). Furthermore, the Jacobian matrix is of $N \times N$ size, and is often too large to allow accurate solution of (7.10). Fortunately, the computational requirements of the estimation of the entire Jacobian matrix 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. Moreover, the block-diagonal structure of the Jacobian allows to decompose the computation of (7.10) to handle each block separately. Furthermore, this block-diagonal structure can be assumed even for compression methods that do not conform with it (e.g., JPEG2000), and thus somewhat compromising the postprocessing result, in order to offer a reasonable run-time. The (possibly assumed) block size of the compression procedure is denoted here as $B_H \times B_W$, and yields a Jacobian with blocks of size $B_H B_W \times B_H B_W$ along its main diagonal.

The code was implemented in Matlab. While the settings differ for the various compression methods, a similar stopping criterion is applied. In (7.3) we introduced the scaled dual-variable of the $i^{th}$ iteration, $u_i \in \mathbb{R}^n$. We here denote $\Delta u_i = \frac{1}{N} \|u_i - u_{i-1}\|_1$ and set the algorithm termination conditions to be at one of the following: $\Delta u_i < 0.05$, $\Delta u_i > \Delta u_{i-1}$ or some maximal number of iterations attained.

The remaining parameters are set for each compression method as specified in Table 7.1. While the relation between the parameters to the compression method is complex, one can claim that the parameters express the non-differentiable nature of the compression function. For example, HEVC compression, which is an intricate compression method, needs smaller $\delta$ values in the Jacobian approximation and a higher $\mu$, both constraining the linearization to be more local than for the other simpler compression methods. Furthermore, the parameter settings consider the compression
Table 7.1: Experimental settings for the examined compression methods

<table>
<thead>
<tr>
<th>Compression Method</th>
<th>Affecting Factors</th>
<th>Max. Iterations</th>
<th>( S_\delta ) = {0.1\Delta k }_{k=1}^5</th>
<th>( \lambda )</th>
<th>( \beta )</th>
<th>( \mu )</th>
<th>( B_l \times B_R )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Scalar Quantization</td>
<td>( r ) bit-rate (bpp)</td>
<td>6</td>
<td>( \hat{\Delta} = \Delta )</td>
<td>0.01</td>
<td>500 \cdot 2^{-2r}</td>
<td>5 \cdot 10^{-4} \cdot 2^b_{in}</td>
<td>1 \times 1</td>
</tr>
<tr>
<td>Simplistic Transform Coding</td>
<td>( \Delta ) quantizer step size (in transform domain)</td>
<td>10</td>
<td>( \hat{\Delta} = \Delta )</td>
<td>0.03</td>
<td>Aligned approx. area: 200 \cdot 2^{-b_{in}}</td>
<td>Rotated approx. area: 500 \cdot 2^{-b_{in}}</td>
<td>5 \cdot 10^{-5} \cdot 2^{b_{in}}</td>
</tr>
<tr>
<td>JPEG</td>
<td>( r ) bit-rate (bpp)</td>
<td>8</td>
<td>( \Delta = 135/r )</td>
<td>0.15</td>
<td>2 \cdot r^{-1}</td>
<td>0.04 \cdot 2^r</td>
<td>8 \times 8</td>
</tr>
<tr>
<td>JPEG2000</td>
<td>( r ) bit-rate (bpp)</td>
<td>8</td>
<td>( \Delta = 100/r )</td>
<td>0.15</td>
<td>5 \cdot r^{-1}</td>
<td>0.3 \cdot 2^r</td>
<td>8 \times 8</td>
</tr>
<tr>
<td>HEVC</td>
<td>( r ) bit-rate (bpp)</td>
<td>8</td>
<td>( \Delta = 50/r )</td>
<td>0.15</td>
<td>5 \cdot r^{-1}</td>
<td>0.3 \cdot 2^r</td>
<td>64 \times 64</td>
</tr>
</tbody>
</table>

Table 7.2: Simplistic Experiment: PSNR Comparison for Scalar Quantization

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

bit-rate, as this quantity reflects the complexity of the given image with respect to the specific compression procedure. Accordingly, the formulas in Table 7.1 were empirically determined to provide an adequate performance.

7.4.1 Simplistic Compression Procedures

Scalar Quantization

We begin with the elementary compression procedure of applying uniform scalar quantization (as formulated in (7.27)) on the signal samples. Motivated by the analysis in section 7.3.3, we define here the approximation interval to be no longer than the quantization step \( \Delta \), as considering only the nearest non-differentiable point yields a useful linearization. Accordingly, the derivative (which is scalar here) is approximated using (7.9) and \( S_\delta = \{0.1\Delta k \}_{k=1}^5 \). Our technique achieved impressive PSNR improvements (Table 7.2) over the entire bit-rate range, and consistently passed the oracle denoiser. Visually, the false-contouring artifacts were significantly reduced (Fig. 7.10).
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 $U_{\pi/4} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & -1 \\ 1 & 1 \end{bmatrix}$, 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 in section 7.3.4: a square area aligned with the axes (i.e., $\eta(x_0, \delta)$), and a $45^\circ$-rotated squared area (i.e., $\hat{\eta}_U(x_0, \delta)$) that allows to calculate the linearization in the transform domain and then transforming the parameters back to the signal domain using (7.47)-(7.48). Both options used square areas of the same size by setting $S_\delta = \{0.1\Delta k\}^5_{k=1}$, where $\Delta$ is the quantizer step size in the transform domain. 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 7.3). The latter observation will motivate us to use aligned-cubic approximation areas also for more complex compression techniques that will follow next.

7.4.2 JPEG

This well known standard [114] is a relatively straight-forward 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 7.4.1, our postprocessing
Table 7.3: Simplistic Experiment: PSNR Comparison for Transform Coding of Two-Component Vectors

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

Figure 7.11: Reconstruction of Lena (512x512) from JPEG compression at 0.363bpp.

method is expected to provide good results here. Indeed, the experiments show the impressive gains of the suggested method that compete with the prominent techniques from [106, 107] (see Table 7.4). The comparison to [106, 107] indirectly considers additional methods such as [89, 91, 103, 122] that were already surpassed by [106] and/or [107]. Moreover, while many competitive methods (e.g., [89, 91, 103, 106]) are mainly intended to low bit-rate compression, our method handles the entire bit-rate range and excels for medium and high bit-rates (Table 7.4). The thorough evaluation here is based on PSNR values, as well as on the perceptual metric of Structural Similarity (SSIM) [62].

Since JPEG applies transform coding on non-overlapping 8x8 blocks, its Jacobian matrix is indeed block diagonal. In addition, the sufficiently small blocks provide a computationally efficient structure that does not need to be simplified further. Conse-
<table>
<thead>
<tr>
<th>Image 512x512</th>
<th>Bit-Rate</th>
<th>JPEG</th>
<th>Oracle Denoiser</th>
<th>Foi et al. [106]</th>
<th>Zhang et al. [107]</th>
<th>Proposed Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lena</td>
<td>0.173</td>
<td>27.33</td>
<td>0.7367</td>
<td>28.77</td>
<td>0.7887</td>
<td>29.07</td>
</tr>
<tr>
<td></td>
<td>0.245</td>
<td>30.41</td>
<td>0.8183</td>
<td>31.82</td>
<td>0.8555</td>
<td>31.97</td>
</tr>
<tr>
<td></td>
<td>0.363</td>
<td>32.96</td>
<td>0.8735</td>
<td>34.12</td>
<td>0.8932</td>
<td>34.06</td>
</tr>
<tr>
<td></td>
<td>0.511</td>
<td>34.76</td>
<td>0.9036</td>
<td>35.64</td>
<td>0.9130</td>
<td>35.55</td>
</tr>
<tr>
<td></td>
<td>0.638</td>
<td>35.81</td>
<td>0.9188</td>
<td>36.52</td>
<td>0.9238</td>
<td>36.44</td>
</tr>
<tr>
<td></td>
<td>0.807</td>
<td>36.86</td>
<td>0.9314</td>
<td>37.42</td>
<td>0.9334</td>
<td>37.34</td>
</tr>
<tr>
<td></td>
<td>1.157</td>
<td>38.54</td>
<td>0.9476</td>
<td>38.91</td>
<td>0.9474</td>
<td>38.82</td>
</tr>
<tr>
<td>Barbara</td>
<td>0.227</td>
<td>23.86</td>
<td>0.6642</td>
<td>25.40</td>
<td>0.7142</td>
<td>24.97</td>
</tr>
<tr>
<td></td>
<td>0.338</td>
<td>25.70</td>
<td>0.7710</td>
<td>27.26</td>
<td>0.8038</td>
<td>26.54</td>
</tr>
<tr>
<td></td>
<td>0.537</td>
<td>28.25</td>
<td>0.8559</td>
<td>30.03</td>
<td>0.8761</td>
<td>28.91</td>
</tr>
<tr>
<td></td>
<td>0.764</td>
<td>30.89</td>
<td>0.9060</td>
<td>32.49</td>
<td>0.9175</td>
<td>31.42</td>
</tr>
<tr>
<td></td>
<td>0.938</td>
<td>32.54</td>
<td>0.9273</td>
<td>33.94</td>
<td>0.9338</td>
<td>33.02</td>
</tr>
<tr>
<td></td>
<td>1.149</td>
<td>34.22</td>
<td>0.9442</td>
<td>35.34</td>
<td>0.9464</td>
<td>34.64</td>
</tr>
<tr>
<td></td>
<td>1.552</td>
<td>36.88</td>
<td>0.9625</td>
<td>37.65</td>
<td>0.9619</td>
<td>37.22</td>
</tr>
<tr>
<td>Boat</td>
<td>0.187</td>
<td>25.56</td>
<td>0.6563</td>
<td>26.70</td>
<td>0.6962</td>
<td>26.75</td>
</tr>
<tr>
<td></td>
<td>0.291</td>
<td>28.13</td>
<td>0.7580</td>
<td>29.19</td>
<td><strong>0.7883</strong></td>
<td>29.14</td>
</tr>
<tr>
<td></td>
<td>0.460</td>
<td>30.49</td>
<td>0.8301</td>
<td>31.37</td>
<td>0.8471</td>
<td>31.29</td>
</tr>
<tr>
<td></td>
<td>0.663</td>
<td>32.35</td>
<td>0.8691</td>
<td>33.14</td>
<td>0.8785</td>
<td>33.04</td>
</tr>
<tr>
<td></td>
<td>0.825</td>
<td>33.50</td>
<td>0.8880</td>
<td>34.16</td>
<td>0.8926</td>
<td>34.07</td>
</tr>
<tr>
<td></td>
<td>1.035</td>
<td>34.63</td>
<td>0.9042</td>
<td>35.19</td>
<td>0.9062</td>
<td>35.10</td>
</tr>
<tr>
<td></td>
<td>1.497</td>
<td>36.43</td>
<td>0.9289</td>
<td>36.91</td>
<td>0.9292</td>
<td>36.82</td>
</tr>
<tr>
<td>Bridge</td>
<td>0.231</td>
<td>23.06</td>
<td>0.5713</td>
<td>23.76</td>
<td>0.5848</td>
<td>23.79</td>
</tr>
<tr>
<td></td>
<td>0.409</td>
<td>25.13</td>
<td>0.7108</td>
<td>25.71</td>
<td>0.7175</td>
<td>25.72</td>
</tr>
<tr>
<td></td>
<td>0.688</td>
<td>27.01</td>
<td>0.8093</td>
<td>27.48</td>
<td>0.8188</td>
<td>27.50</td>
</tr>
<tr>
<td></td>
<td>1.003</td>
<td>28.50</td>
<td>0.8634</td>
<td>28.88</td>
<td>0.8603</td>
<td>28.88</td>
</tr>
<tr>
<td></td>
<td>1.261</td>
<td>29.54</td>
<td>0.8917</td>
<td>29.95</td>
<td>0.8993</td>
<td>29.89</td>
</tr>
<tr>
<td></td>
<td>1.579</td>
<td>30.80</td>
<td>0.9166</td>
<td>31.23</td>
<td>0.9153</td>
<td>31.09</td>
</tr>
<tr>
<td></td>
<td>2.197</td>
<td>33.32</td>
<td>0.9400</td>
<td>33.78</td>
<td>0.9474</td>
<td>33.57</td>
</tr>
<tr>
<td>Peppers</td>
<td>0.176</td>
<td>27.17</td>
<td>0.7078</td>
<td>28.75</td>
<td>0.7748</td>
<td>29.04</td>
</tr>
<tr>
<td></td>
<td>0.246</td>
<td>30.14</td>
<td>0.7839</td>
<td>31.50</td>
<td>0.8273</td>
<td>31.69</td>
</tr>
<tr>
<td></td>
<td>0.361</td>
<td>32.44</td>
<td>0.8354</td>
<td>33.45</td>
<td>0.8564</td>
<td>33.52</td>
</tr>
<tr>
<td></td>
<td>0.513</td>
<td>33.92</td>
<td>0.8651</td>
<td>34.61</td>
<td>0.8730</td>
<td>34.64</td>
</tr>
<tr>
<td></td>
<td>0.650</td>
<td>34.77</td>
<td>0.8809</td>
<td>35.30</td>
<td>0.8829</td>
<td>35.31</td>
</tr>
<tr>
<td></td>
<td>0.832</td>
<td>35.59</td>
<td>0.8949</td>
<td>35.96</td>
<td>0.8941</td>
<td>35.94</td>
</tr>
<tr>
<td></td>
<td>1.203</td>
<td>36.79</td>
<td>0.9335</td>
<td>37.03</td>
<td>0.9111</td>
<td>37.01</td>
</tr>
</tbody>
</table>

Table 7.4: JPEG: Result Comparison
sequently, the run-time of the JPEG postprocessing (Matlab implementation) is rather reasonable and is usually about 1-2 minutes for a 512x512 image.

### 7.4.3 JPEG2000

This efficient standard\(^{115}\) 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 (via the Kakadu software\(^{123}\)) without any tiling. However, the Jacobian was estimated by assuming independent 8x8 blocks, where this reduced accuracy yielded considerable relief in the computational burden (the postprocessing took 5-8 minutes for a 512x512 image). The reconstruction PSNR of our method reached up to 0.7dB improvement of the JPEG2000 output (e.g., see Fig. 7.12). The results in\(^{108,109}\) were provided to postprocessing of low bit-rate compression. Therefore, we first compare our results to these from\(^{108,109}\) (Table 7.5) according to their experimental settings, and then show results for higher bit-rates where our method is even more effective (Table 7.6). Table 7.5 exhibits that our method outperforms\(^{108}\) and competitive with the technique from\(^{109}\). In addition, our results for higher bit-rates (Table 7.6) compete with the oracle denoiser. These results are encouraging since the oracle denoiser needs the precompressed image and, therefore, is not suitable for the common compression applications. Furthermore, the results in Tables 7.5 and 7.6 establish our technique as suitable for a wide range of bit-rates. The restoration results visually demonstrated the artifact reduction using our method, specifically, handling of the ringing artifact (Fig. 7.12).

### 7.4.4 HEVC

This state-of-the-art coding standard offers a profile of still-image compression\(^{29,116}\). 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 size 64x64, and thus the corresponding run-time was higher than for the previous compression methods. More specifically, postprocessing an HEVC-compressed image took several hours, in contrast to few minutes as needed for the previous compression methods. Accordingly, we stress that our purpose here is to demonstrate the conceptual suitability of our method to compression techniques that are significantly more intricate than transform coding.

The results here are for HEVC-compression using the software library in\(^{50}\). Again,
Figure 7.12: Reconstruction of Barbara (512x512) from JPEG2000 compression at 0.40bpp.
Table 7.5: JPEG2000: Comparison of PSNR Gains at Low Bit-Rates

<table>
<thead>
<tr>
<th>Image 512x512</th>
<th>Bit Rate $^2$</th>
<th>JPEG2000 $^3$</th>
<th>Zhai et al. $[108]$</th>
<th>Kwon et al. $[109]$</th>
<th>Proposed Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lena</td>
<td>0.10</td>
<td>29.86</td>
<td>0.03</td>
<td>0.50</td>
<td>0.40</td>
</tr>
<tr>
<td></td>
<td>0.15</td>
<td>31.63</td>
<td>-0.34</td>
<td>0.54</td>
<td>0.39</td>
</tr>
<tr>
<td></td>
<td>0.20</td>
<td>33.01</td>
<td>0.08</td>
<td>NA</td>
<td>0.47</td>
</tr>
<tr>
<td>Peppers</td>
<td>0.10</td>
<td>29.58</td>
<td>0.31</td>
<td>0.47</td>
<td>0.47</td>
</tr>
<tr>
<td></td>
<td>0.15</td>
<td>31.33</td>
<td>0.04</td>
<td>0.62</td>
<td>0.41</td>
</tr>
<tr>
<td></td>
<td>0.20</td>
<td>32.51</td>
<td>-0.27</td>
<td>NA</td>
<td>0.46</td>
</tr>
<tr>
<td>Bridge</td>
<td>0.10</td>
<td>22.81</td>
<td>-0.05</td>
<td>0.04</td>
<td>0.06</td>
</tr>
<tr>
<td></td>
<td>0.15</td>
<td>23.76</td>
<td>-0.13</td>
<td>0.10</td>
<td>0.09</td>
</tr>
<tr>
<td></td>
<td>0.20</td>
<td>24.34</td>
<td>-0.17</td>
<td>NA</td>
<td>0.11</td>
</tr>
</tbody>
</table>

1 The results presented in $[108, 109]$ for JPEG2000 postprocessing are for somewhat different versions of the widely-used Lena, Peppers and Bridge images. Accordingly, in this table, and only here, we refer to these images that were specified in $[109]$.

2 The bit-rate here is the input given to the Kakadu software, and is not necessarily the accurate output bit-rate.

3 As explained in $[109]$, the PSNR values of the compression using the Kakadu software are slightly different in the various papers. Accordingly, the comparison is for the PSNR gains.

our postprocessing results reached up to 0.3dB gain in PSNR and often exceeded the oracle denoiser, as shown in the PSNR and SSIM comparison in Table 7.7. Figure 7.13 visually demonstrates our method’s treatment of the delicate artifacts of the HEVC. To the best of our knowledge, no other artifact-reduction techniques for the HEVC still-image profile have been proposed yet, as it is a recent standard.

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.

7.5 Extension to Postprocessing of Intra-Frame Video Coding

Let us extend our work presented above in this chapter and propose a postprocessing technique for video signals compressed using intra-frame coding methods, where each frame is coded independently of the rest of the sequence. Our proposed method is iterative – in each step it solves an optimization problem that involves the compression-decompression operator, and applies a Gaussian video denoiser. As video frames are separately coded, the treatment of the compression-decompression operator elegantly reduces to individually considering single frames, and thus, the corresponding optimization can be efficiently solved. Contrastingly, the spatio-temporal structure of the signal is utilized by incorporating an efficient video Gaussian-denoiser (such as V-BM4D $[124]$...
<table>
<thead>
<tr>
<th>Image 512x512</th>
<th>Bit Rate</th>
<th>JPEG2000</th>
<th>Oracle Denoiser</th>
<th>Proposed Method</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>PSNR</td>
<td>SSIM</td>
<td>PSNR</td>
</tr>
<tr>
<td>Lena</td>
<td>0.30</td>
<td>34.87</td>
<td>0.8983</td>
<td>35.20</td>
</tr>
<tr>
<td></td>
<td>0.40</td>
<td>36.12</td>
<td>0.9142</td>
<td><strong>36.43</strong></td>
</tr>
<tr>
<td></td>
<td>0.50</td>
<td>37.26</td>
<td>0.9255</td>
<td><strong>37.49</strong></td>
</tr>
<tr>
<td></td>
<td>0.60</td>
<td>37.97</td>
<td>0.9326</td>
<td><strong>38.16</strong></td>
</tr>
<tr>
<td></td>
<td>0.70</td>
<td>38.69</td>
<td>0.9392</td>
<td><strong>38.87</strong></td>
</tr>
<tr>
<td>Barbara</td>
<td>0.30</td>
<td>29.16</td>
<td>0.8506</td>
<td><strong>30.00</strong></td>
</tr>
<tr>
<td></td>
<td>0.40</td>
<td>30.79</td>
<td>0.8791</td>
<td><strong>31.70</strong></td>
</tr>
<tr>
<td></td>
<td>0.50</td>
<td>32.16</td>
<td>0.9049</td>
<td><strong>33.09</strong></td>
</tr>
<tr>
<td></td>
<td>0.60</td>
<td>33.30</td>
<td>0.9199</td>
<td><strong>34.20</strong></td>
</tr>
<tr>
<td></td>
<td>0.70</td>
<td>34.37</td>
<td>0.9296</td>
<td><strong>35.17</strong></td>
</tr>
<tr>
<td>Boat</td>
<td>0.30</td>
<td>30.87</td>
<td>0.8204</td>
<td>31.22</td>
</tr>
<tr>
<td></td>
<td>0.40</td>
<td>32.29</td>
<td>0.8508</td>
<td>32.61</td>
</tr>
<tr>
<td></td>
<td>0.50</td>
<td>33.32</td>
<td>0.8710</td>
<td>33.58</td>
</tr>
<tr>
<td></td>
<td>0.60</td>
<td>34.17</td>
<td>0.8858</td>
<td>34.44</td>
</tr>
<tr>
<td></td>
<td>0.70</td>
<td>34.90</td>
<td>0.8975</td>
<td>35.13</td>
</tr>
<tr>
<td>Bridge</td>
<td>0.30</td>
<td>25.42</td>
<td><strong>0.6922</strong></td>
<td>25.50</td>
</tr>
<tr>
<td></td>
<td>0.40</td>
<td>26.36</td>
<td><strong>0.7510</strong></td>
<td>26.41</td>
</tr>
<tr>
<td></td>
<td>0.50</td>
<td>27.24</td>
<td>0.7852</td>
<td>27.35</td>
</tr>
<tr>
<td></td>
<td>0.60</td>
<td>27.89</td>
<td>0.8143</td>
<td>28.02</td>
</tr>
<tr>
<td></td>
<td>0.70</td>
<td>28.50</td>
<td>0.8391</td>
<td>28.64</td>
</tr>
<tr>
<td>Peppers</td>
<td>0.30</td>
<td>34.11</td>
<td>0.8587</td>
<td>34.37</td>
</tr>
<tr>
<td></td>
<td>0.40</td>
<td>35.05</td>
<td>0.8722</td>
<td><strong>35.26</strong></td>
</tr>
<tr>
<td></td>
<td>0.50</td>
<td>35.80</td>
<td>0.8833</td>
<td><strong>35.96</strong></td>
</tr>
<tr>
<td></td>
<td>0.60</td>
<td>36.34</td>
<td>0.8955</td>
<td><strong>36.44</strong></td>
</tr>
<tr>
<td></td>
<td>0.70</td>
<td>36.83</td>
<td>0.9056</td>
<td>36.91</td>
</tr>
</tbody>
</table>
Table 7.7: HEVC: Result Comparison

<table>
<thead>
<tr>
<th>Image 128x128</th>
<th>Bit Rate</th>
<th>HEVC</th>
<th>Oracle Denoiser</th>
<th>Proposed Method</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>PSNR</td>
<td>SSIM</td>
<td>PSNR</td>
<td>SSIM</td>
</tr>
<tr>
<td>Lena</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.177</td>
<td>25.79</td>
<td>0.7278</td>
<td>25.91</td>
<td>0.7331</td>
</tr>
<tr>
<td>0.340</td>
<td>28.91</td>
<td>0.8357</td>
<td>28.97</td>
<td>0.8349</td>
</tr>
<tr>
<td>0.639</td>
<td>32.71</td>
<td>0.9180</td>
<td>32.84</td>
<td>0.9181</td>
</tr>
<tr>
<td>1.046</td>
<td>36.31</td>
<td>0.9577</td>
<td>36.44</td>
<td>0.9572</td>
</tr>
<tr>
<td>Barbara</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.120</td>
<td>27.55</td>
<td>0.7658</td>
<td>27.71</td>
<td>0.7726</td>
</tr>
<tr>
<td>0.206</td>
<td>30.22</td>
<td>0.8420</td>
<td>30.39</td>
<td>0.8471</td>
</tr>
<tr>
<td>0.401</td>
<td>33.30</td>
<td>0.9163</td>
<td>33.48</td>
<td>0.9181</td>
</tr>
<tr>
<td>0.746</td>
<td>36.81</td>
<td>0.9554</td>
<td>37.04</td>
<td>0.9570</td>
</tr>
<tr>
<td>Boat</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.212</td>
<td>25.29</td>
<td>0.7566</td>
<td>25.33</td>
<td>0.7568</td>
</tr>
<tr>
<td>0.416</td>
<td>28.54</td>
<td>0.8640</td>
<td>28.59</td>
<td>0.8614</td>
</tr>
<tr>
<td>0.735</td>
<td>32.44</td>
<td>0.9346</td>
<td>32.55</td>
<td>0.9352</td>
</tr>
<tr>
<td>1.186</td>
<td>36.51</td>
<td>0.9682</td>
<td>36.61</td>
<td>0.9684</td>
</tr>
<tr>
<td>Bridge</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.198</td>
<td>25.08</td>
<td>0.7172</td>
<td>25.11</td>
<td>0.7142</td>
</tr>
<tr>
<td>0.393</td>
<td>28.10</td>
<td>0.8287</td>
<td>28.10</td>
<td>0.8287</td>
</tr>
<tr>
<td>0.746</td>
<td>31.64</td>
<td>0.9151</td>
<td>31.64</td>
<td>0.9151</td>
</tr>
<tr>
<td>1.289</td>
<td>35.68</td>
<td>0.9653</td>
<td>35.69</td>
<td>0.9653</td>
</tr>
<tr>
<td>Peppers</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.1323</td>
<td>28.40</td>
<td>0.8328</td>
<td>28.53</td>
<td>0.8409</td>
</tr>
<tr>
<td>0.2285</td>
<td>31.51</td>
<td>0.8947</td>
<td>31.71</td>
<td>0.8993</td>
</tr>
<tr>
<td>0.3813</td>
<td>34.73</td>
<td>0.9330</td>
<td>34.92</td>
<td>0.9354</td>
</tr>
<tr>
<td>0.6284</td>
<td>38.01</td>
<td>0.9563</td>
<td>38.18</td>
<td>0.9569</td>
</tr>
</tbody>
</table>

Here the images are the 128x128 portions taken from the center of the 256x256 images.
Figure 7.13: Reconstruction of Lena (128x128 Portion) from HEVC compression at 0.639bpp.

or BM4D [125]), which is applied on frame groups.

The distributed coding approach suggests to separately encode dependent data elements, and to compensate this suboptimality via a complicated decoder that utilizes the inter-component relations by jointly reconstructing them. Accordingly, the low-complexity intra-frame video encoder together with the extended decoder, defined as the intra-frame decoder followed by the proposed spatio-temporal postprocessing, constitute an elementary distributed video-coding system [126].

7.5.1 The Proposed Postprocessing Method

The Video Signal and Intra-Frame Coding

Let us consider a video signal consisting of $T$ frames, each has a spatial resolution of $W$ pixel width and $H$ pixel height. Accordingly, the column-stack form of the signal is denoted here as $x \in \mathbb{R}^N$, where $N = T \cdot W \cdot H$ is the total number of samples in the signal. The signal $x$ is, in fact, a vertical concatenation of the column-stack form of its $T$ frames, i.e.,

$$x = \begin{bmatrix} x^{(1)} \\ \vdots \\ x^{(T)} \end{bmatrix}$$  \hspace{1cm} (7.50)

where $x^{(i)} \in \mathbb{R}^{N_f}$ is the column-stack form of the $i^{th}$ frame ($i = 1, ..., T$), and $N_f = W \cdot H$ is the number of pixels in a single frame.

The video signal $x$ undergoes a compression-decompression procedure, $C : \mathbb{R}^N \rightarrow \mathbb{R}^N$, 

142
resulting in the reconstructed signal $y = C(x)$. More specifically, we examine here an intra-frame coding procedure, where each frame is separately compressed-decompressed via the same procedure, $C_f : \mathbb{R}^{N_f} \rightarrow \mathbb{R}^{N_f}$. Accordingly, the reconstructed video satisfies

$$y = C(x) = \begin{bmatrix} C_f(x^{(1)}) \\ \vdots \\ C_f(x^{(T)}) \end{bmatrix}.$$  \hfill (7.51)

### 7.5.2 Problem Formulation using ADMM

For lossy compression methods an error is introduced of magnitude that depends on the bit-budget, the specific-signal characteristics, and the compression algorithm. We aim at restoring the precompressed signal $x$ from the reconstruction $y$ using the following regularized inverse-problem:

$$\hat{x} = \arg\min_x \|y - C(x)\|_2^2 + \beta s(x),$$  \hfill (7.52)

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

Similar to [17] and [16], we develop an iterative algorithm for the solution of (7.52). We start by applying variable splitting that yields the following equivalent form of (7.52):

$$\min_{x,v} \|y - C(x)\|_2^2 + \beta s(v) \quad \text{s.t.} \quad x = v,$$  \hfill (7.53)

where $v \in \mathbb{R}^N$ is an additional vector due to the split. The constrained problem (7.53) is addressed by forming an augmented Lagrangian and its corresponding iterative solution (of its scaled version) via the method of multipliers [15, ch. 2], where the $i^{th}$ iteration consists of

$$(\hat{x}_i, \hat{v}_i) = \arg\min_{x,v} \|y - C(x)\|_2^2 + \beta s(v) + \frac{\lambda}{2} \|x - v + u_i\|_2^2$$  \hfill (7.54)

where $u_i \in \mathbb{R}^n$ is the scaled dual-variable and $\lambda$ is an auxiliary parameter, both introduced in the Lagrangian.

Please note the following notation remark for a general vector $u$. Whereas $u_i$ stands for vector $u$ in the $i^{th}$ iteration, $u_j^j$ represents the $j^{th}$ component (a scalar) of the vector $u$. In addition, $u^{(j)}$ denotes the $j^{th}$ frame of a video signal $u$ and, accordingly, $u_i^{(j)}$ denotes the $j^{th}$ frame of the video signal $u_i$. Approximating the joint optimization of $x$ and $v$ in (7.54), using one iteration of alternating minimization, results in the iterative solution in the ADMM form, where
the $i^{th}$ iteration consists of

$$\hat{x}_i = \arg \min_{x} \|y - C(x)\|^2 + \frac{\lambda}{2} \|x - \hat{x}_i\|^2 \tag{7.55}$$

$$\hat{v}_i = \arg \min_{v} \frac{\lambda}{2} \|v - \hat{v}_i\|^2 + \beta s(v) \tag{7.56}$$

$$u_{i+1} = u_i + (\hat{x}_i - \hat{v}_i). \tag{7.57}$$

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

The regularization step (7.56) is of the form of a Gaussian denoising optimization-problem (of a noise level determined by $\beta/\lambda$) and therefore can be viewed as applying a denoising algorithm to the signal $\hat{v}_i$. Indeed, the Plug-and-Play Priors framework [17] suggests exactly this strategy, replacing (7.56) with an independent denoiser; even one that does not explicitly have in its formulation a minimization problem of the form of (7.56), i.e., it replaces (7.56) with a denoising operation, $\hat{v}_i = \text{Denoise}_{\beta/\lambda}(\hat{v}_i)$. The deployment of a favorable denoiser introduces valuable practical benefits to the design of the proposed postprocessing procedure, and yields a powerful generic method. Moreover, by choosing an effective video denoiser that utilizes the inter-frame relations, our postprocessing becomes a spatio-temporal procedure attempting to repair the inefficiency of the intra-frame encoder.

**Linear Approximation of the Intra-Frame Compression-Decompression Procedure**

Due to the high nonlinearity of $C(x)$, we further simplify the forward-model step (7.55) using a first-order Taylor approximation of the compression-decompression function around $\hat{x}_{i-1}$, i.e.,

$$C_{lin}(x) = C(\hat{x}_{i-1}) + \frac{dC(z)}{dz} \bigg|_{z=\hat{x}_{i-1}} \cdot (x - \hat{x}_{i-1}) \tag{7.58}$$

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

Since the approximation of the Jacobian, $\frac{dC(z)}{dz}$, deeply influences the restoration result and the computational cost, this is a delicate task. First, $C$ is a nonlinear and even nondifferentiable function as the compression relies on quantization and/or thresholding. Second, we provide here a generic technique, and therefore do not explicitly consider the compression-decompression formulation except from utilizing its intra-frame coding structure.

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 via a mathematical analysis provided in Section 7.3. As we might be approximating the derivative in the neighborhood of a nondifferential 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:

$$
\frac{dC(z)}{dz_k} = \frac{1}{|S_\delta|} \sum_{\delta \in S_\delta} \frac{C(z + \delta \cdot e_k) - C(z - \delta \cdot e_k)}{2\delta},
$$

(7.59)

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

The straightforward computation of the Jacobian according to (7.59) is costly, especially for video signals, as it heavily depends on the total number of samples, $N$. Particularly, the Jacobian matrix has $N$ columns, each should be computed using (7.59) via $2 \cdot |S_\delta|$ compression-decompression applications. In addition, the Jacobian matrix of $C$ is of $N \times N$ size and, thus, may also impose practical difficulties in solving (7.55).

Fortunately, the intra-frame coding procedure has the form in (7.51), where the frames are individually encoded-decoded using the procedure $C_f$. Consequently, the Jacobian matrix has the block-diagonal form of

$$
\begin{bmatrix}
\left. \frac{dC_f(w)}{dw_k} \right|_{w = \hat{x}_{i-1}^{(1)}} & 0 & \cdots & 0 \\
0 & \left. \frac{dC_f(w)}{dw_k} \right|_{w = \hat{x}_{i-1}^{(2)}} & \cdots & \vdots \\
\vdots & 0 & \ddots & 0 \\
0 & \cdots & 0 & \left. \frac{dC_f(w)}{dw_k} \right|_{w = \hat{x}_{i-1}^{(T)}}
\end{bmatrix},
$$

(7.60)

where $\left. \frac{dC_f(w)}{dw_k} \right|_{w = \hat{x}_{i-1}^{(j)}}$ is the $N_f \times N_f$ Jacobian matrix of the frame compression-decompression procedure, $C_f$, around the $N_f$-dimensional point, $\hat{x}_{i-1}^{(j)}$, that corresponds to the $j^{th}$ frame of $\hat{x}_{i-1}$. The block-diagonal form in (7.60) lets us to concurrently compute $T$ Jacobian columns, each belonging to a different video frame. Namely, instead of using (7.59) for $z = \hat{x}_{i-1}$, we employ the following relation:

$$
\sum_{\delta \in S_\delta} C \left( z + \delta \sum_{j=0}^{T-1} e_{(jN_f+k)} \right) - C \left( z - \delta \sum_{j=0}^{T-1} e_{(jN_f+k)} \right) \frac{1}{2\delta},
$$

(7.61)

where $e_{(jN_f+k)}$ is the $(jN_f+k)^{th}$ standard direction vector, and $\left. \frac{dC_f(w)}{dw_k} \right|_{w = \hat{x}_{i-1}^{(j)}}$ is the $k^{th}$ column ($k = 1, \ldots, N_f$) of the Jacobian matrix corresponding to encoding-decoding the $j^{th}$ frame of the video signal $z$. Accordingly, the complete Jacobian matrix can be
formed, for \( z = \tilde{x}_{i-1} \), following the structure in (7.60).

Due to the high nonlinearity of \( C \), the linear approximation (7.58) is reasonable in a small neighborhood around the approximating point \( \tilde{x}_{i-1} \). Accordingly, we further constrain the distance of the solution from the linear-approximation point by modifying (7.55) to

\[
\tilde{x}_i = \arg \min_x \| y - C_{lin}(x) \|_2^2 + \frac{\lambda}{2} \| x - \tilde{x}_i \|_2^2 + \mu \| x - \tilde{x}_{i-1} \|_2^2.
\] (7.62)

The intra-frame coding structure in (7.51) and the corresponding block-diagonal form of its Jacobian matrix (7.60) allow us to decompose the linear form given in (7.58) for the entire video signal, and to write

\[
C_{lin}(x) = \begin{bmatrix}
C_{f,lin}(x^{(1)}) \\
\vdots \\
C_{f,lin}(x^{(T)})
\end{bmatrix}
\] (7.63)

where \( C_{f,lin} \) is the linear approximation of the frame compression-decompression procedure, formulated for the \( j^{th} \) frame (\( j = 1, \ldots, T \)) of the video \( x \) as

\[
C_{f,lin}(x^{(j)}) = C_f(\tilde{x}_{i-1}^{(j)}) + \left. \frac{dC_f(w)}{dw} \right|_{w = \tilde{x}_{i-1}^{(j)}} \cdot (x^{(j)} - \tilde{x}_{i-1}^{(j)}).
\] (7.64)

Accordingly, the optimization problem in (7.62), defined for the entire video signal, reduces to a set of distinct optimizations, each considering a single frame. More specifically, the problem of the \( j^{th} \) frame (\( j = 1, \ldots, T \)) is formulated as

\[
\tilde{x}_{i}^{(j)} = \arg \min_{x^{(j)}} \| y - C_{f,lin}(x^{(j)}) \|_2^2 + \frac{\lambda}{2} \| x^{(j)} - \tilde{x}_{i}^{(j)} \|_2^2 + \mu \| x^{(j)} - \tilde{x}_{i-1}^{(j)} \|_2^2.
\] (7.65)

The results of the frame-level optimizations are concatenated to form the entire spatio-temporal solution \( \tilde{x}_i \).

The proposed method for postprocessing intra-frame coded video signals is summarized in Algorithm 7.2.

**Further Reduction of the Computational Complexity**

The computational complexity of our method is mainly determined by the complexity levels of the utilized denoiser and the Jacobian estimation procedure. The latter further depends on the implementation of the compression-decompression method, as it is repeatedly applied according to (7.61). In addition, Equation (7.61) exhibits also the effect of the size of the set \( S_\delta \) utilized for jointly approximating \( T \) columns of the Jacobian of the video compression-decompression procedure, \( C \). Since the number of columns in the Jacobian of the frame compression-decompression procedure, \( C_f \), is the number of frame samples (denoted as \( N_f \)), computation of the Jacobian in
**Algorithm 7.2** The Proposed Postprocessing Method

1: \( \hat{x}_0 = y, \hat{v}_0 = y \)
2: \( i = 1, u_1 = 0 \)
3: repeat
4:   Approximate \( C_{f,lin}(\cdot) \) around \( \hat{x}_{i-1}^{(j)} \) \( (j = 1, ..., T) \) using (7.64) and (7.61)
5:   \( \hat{x}_i = \hat{v}_{i-1} - u_i \)
6:   Form \( \hat{x}_i \) by solving for \( j = 1, ..., T \):
   \[
   \hat{x}_i^{(j)} = \arg\min_{x^{(j)}} \| y - C_{f,lin}(x^{(j)}) \|_2^2 + \frac{\lambda}{2} \| x^{(j)} - \hat{x}_i^{(j)} \|_2^2 + \mu \| x^{(j)} - \hat{x}_{i-1}^{(j)} \|_2^2
   \]
7:   \( \hat{v}_i = \hat{x}_i + u_i \)
8:   \( \hat{v}_i = \text{Denoise}_{\beta/\lambda}(\hat{v}_i) \)
9:   \( u_{i+1} = u_i + (\hat{x}_i - \hat{v}_i) \)
10: \( i \leftarrow i + 1 \)
11: until stopping criterion is satisfied

(7.60) may still be costly, as it requires \( N_f \) calculations of (7.61). Fortunately, the computational requirements for estimating the Jacobian matrix can be further relaxed for many compression methods that operate independently on non-overlapping blocks within each frame. In this case, the Jacobian of the frame compression-decompression procedure, \( \frac{dC_f(w)}{dw} \), 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. Moreover, the block-diagonal structure of the Jacobian allows to decompose the computation of (7.65) to handle each spatial-block separately. The details of this computational simplification are inherited from the discussion given above for linearization of separately coded frames.

In addition, the block-diagonal structure of \( \frac{dC_f(w)}{dw} \) can be assumed even for compression methods that do not conform with it (e.g., Motion-JPEG2000), and thus somewhat compromising the postprocessing result, in order to offer a reasonable run-time. The (possibly assumed) spatial-block size of the compression procedure is denoted here as \( B_H \times B_W \), and yields a Jacobian with blocks of size \( B_H B_W \times B_H B_W \) along its main diagonal.

### 7.5.3 Experimental Results

In this section we demonstrate the performance of the proposed postprocessing method by presenting results obtained in conjunction with the well-known Motion-JPEG2000 standard [127]. Motion-JPEG2000 is an intra-frame coding method that naturally extends the JPEG2000 still-image compression standard [115] to video signals. More specifically, each frame in the sequence is independently encoded using a wavelet-based transform coding procedure, applied on large spatial tiles (of at least \( 128 \times 128 \) pixels).

We use the BM4D method [125] as the denoiser. While the BM4D was designed to denoise volumetric data, it was also established in [125] as suitable for video denoising.
(especially for low-motion sequences). Since the proposed postprocessing technique uses a well established denoiser as a subroutine, we compare our method with a single application of this denoiser as a postprocessing procedure. This competing approach is further strengthened by endorsing the denoiser with an oracle capability by searching for the best parameter in terms of maximal average frame-PSNR (AFPSNR) result. More specifically, this oracle denoiser optimizes its output AFPSNR based on the knowledge of the precompressed video, a capability that cannot be applied in a real postprocessing task.

The code was implemented in Matlab. The following stopping criterion was applied. In (7.54) we introduced the scaled dual-variable of the $i^{th}$ iteration, $u_i \in \mathbb{R}^N$. We here denote $\Delta u_i = \frac{1}{N} \|u_i - u_{i-1}\|_1$ and set the algorithm termination conditions to be at one of the following: $\Delta u_i < 0.05$, $\Delta u_i > \Delta u_{i-1}$ or maximal number of five iterations attained. The remaining parameters are set as follows. The derivatives are approximated according to a spatial block-size $B_H \times B_W$ of $8 \times 8$ pixels and $S_k = \left\{ 27 \cdot k \times (\text{bpp})^{-1} \right\}_{k=1}^5$, where bpp is the bit-rate of the group of frames. The components in the optimization problems are weighted by $\lambda = 0.15$, $\beta = b_0 \times 5 \cdot 10^{-4} \times (\text{bpp})^{-1}$ and $\mu = 0.3 \times 2^{0.1875 - \text{bpp}}$, where $b_0$ is a parameter depending on the temporal characteristics of the processed video. We set $b_0 = 10$ for low-motion videos (e.g., Akiyo, News and Hall Monitor), and $b_0 = 1$ for sequences of higher motion-levels (such as Highway and Ice). Note that the parameter settings can be further improved, e.g., the parameter $b_0$ can be automatically determined as a function of the average squared difference of frames in the postprocessed sequence.

We evaluated our method for several video sequences at CIF resolution (frame size of $352 \times 288$ pixels), where a group of 16 frames forms the spatio-temporal signal to consider. The quality of the reconstructed-from-compression and the postprocessed videos is measured in terms of average frame-PSNR (AFPSNR) of the sequence. The reconstruction AFPSNR of our method reached up to 0.9 dB improvement of the Motion-JPEG2000 output (Table 7.8). In addition, our results compete with the oracle denoiser (see Table 7.8). These results are encouraging since the oracle denoiser needs the precompressed video and, therefore, is not suitable for the common compression applications. Furthermore, the results in Table 7.8 establish our technique as suitable for a wide range of bit-rates. The restoration results visually demonstrated the artifact reduction using our method, specifically, handling of the ringing artifact (see Figures 7.14-7.15).
Table 7.8: Result Comparison for Motion-JPEG2000

<table>
<thead>
<tr>
<th>Video</th>
<th>Bit Rate</th>
<th>Motion-JPEG2000 AFPSNR</th>
<th>Oracle Denoiser AFPSNR</th>
<th>Proposed Method AFPSNR</th>
</tr>
</thead>
<tbody>
<tr>
<td>Akiyo</td>
<td>0.08</td>
<td>30.76</td>
<td>31.08</td>
<td>31.07</td>
</tr>
<tr>
<td></td>
<td>0.09</td>
<td>31.76</td>
<td>32.17</td>
<td>32.14</td>
</tr>
<tr>
<td></td>
<td>0.12</td>
<td>33.08</td>
<td>33.58</td>
<td>33.53</td>
</tr>
<tr>
<td></td>
<td>0.16</td>
<td>35.08</td>
<td>35.80</td>
<td>35.71</td>
</tr>
<tr>
<td></td>
<td>0.27</td>
<td>38.61</td>
<td>39.19</td>
<td>39.28</td>
</tr>
<tr>
<td>News</td>
<td>0.08</td>
<td>25.41</td>
<td>25.65</td>
<td>25.68</td>
</tr>
<tr>
<td></td>
<td>0.09</td>
<td>26.31</td>
<td>26.53</td>
<td>26.56</td>
</tr>
<tr>
<td></td>
<td>0.12</td>
<td>27.35</td>
<td>27.68</td>
<td>27.68</td>
</tr>
<tr>
<td></td>
<td>0.16</td>
<td>28.97</td>
<td>29.58</td>
<td>29.54</td>
</tr>
<tr>
<td></td>
<td>0.27</td>
<td>32.08</td>
<td>32.76</td>
<td>32.77</td>
</tr>
<tr>
<td>Hall Monitor</td>
<td>0.08</td>
<td>25.89</td>
<td>26.17</td>
<td>26.17</td>
</tr>
<tr>
<td></td>
<td>0.09</td>
<td>26.80</td>
<td>27.13</td>
<td>27.15</td>
</tr>
<tr>
<td></td>
<td>0.12</td>
<td>27.97</td>
<td>28.45</td>
<td>28.44</td>
</tr>
<tr>
<td></td>
<td>0.16</td>
<td>29.78</td>
<td>30.57</td>
<td>30.51</td>
</tr>
<tr>
<td></td>
<td>0.27</td>
<td>33.29</td>
<td>34.34</td>
<td>34.26</td>
</tr>
<tr>
<td>Highway</td>
<td>0.08</td>
<td>33.36</td>
<td>33.98</td>
<td>34.00</td>
</tr>
<tr>
<td></td>
<td>0.09</td>
<td>34.42</td>
<td>34.83</td>
<td>35.01</td>
</tr>
<tr>
<td></td>
<td>0.12</td>
<td>35.64</td>
<td>35.75</td>
<td>36.25</td>
</tr>
<tr>
<td></td>
<td>0.16</td>
<td>37.36</td>
<td>36.73</td>
<td>37.86</td>
</tr>
<tr>
<td></td>
<td>0.27</td>
<td>39.98</td>
<td>37.69</td>
<td>40.13</td>
</tr>
<tr>
<td>Ice</td>
<td>0.08</td>
<td>28.25</td>
<td>28.88</td>
<td>28.67</td>
</tr>
<tr>
<td></td>
<td>0.09</td>
<td>29.35</td>
<td>30.09</td>
<td>29.91</td>
</tr>
<tr>
<td></td>
<td>0.12</td>
<td>30.67</td>
<td>31.47</td>
<td>31.39</td>
</tr>
<tr>
<td></td>
<td>0.16</td>
<td>32.47</td>
<td>32.88</td>
<td>33.16</td>
</tr>
<tr>
<td></td>
<td>0.27</td>
<td>35.49</td>
<td>34.77</td>
<td>36.32</td>
</tr>
</tbody>
</table>

The best results (up to a difference of 0.05dB in the average frame-PSNR) are marked in bold text.

Figure 7.14: Reconstruction of the 5th frame of Highway (CIF) from Motion-JPEG2000 compression at 0.08bpp.
Figure 7.15: Reconstruction of News (CIF) from Motion-JPEG2000 compression at 0.27bpp.
Chapter 8

Conclusion

8.1 Thesis Summary

In this thesis, several topics related to compression and restoration of data are studied. Our findings are potentially useful in a wide variety of practical situations. The contributions of our work can be summarized as follows:

1. **An iterative-optimization methodology for intricate compression problems.**

   We proposed an optimization framework for compression problems of a complex form due to unusual, but practically important, distortion metrics. Our iterative approach employs the alternating direction method of multipliers (ADMM) to cast the problem into a sequence of easier tasks. Remarkably, the utilization of variable splitting, included in the ADMM, leads to an optimization process with a repeated stage interpreted as an application of a standard compression technique. Our new compression framework has a wide variety of possible applications. A family of problems involving compression in conjunction to processing/degradation operators were extensively studied in this work. Additional promising utilizations of our approach are described below in the future research directions section.

2. **System optimization from the compression standpoint.**

   In this part we studied a common system structure where lossy compression is employed as an intermediate stage among other processing components. We addressed the problem that whereas the valuable distortion is determined by the system’s input and output signals, the compression design commonly disregards the system structure and, by that, provides an overall sub-optimal rate-distortion performance. We proposed a compression methodology for an operational rate-distortion optimization considering a known system layout, modeled using linear operators and noise. Using the ADMM technique, we showed that the design of the new globally-optimized compression reduces to a standard compression of a "system adjusted" signal.
3. **Extension of image and video compression standards to optimize complete acquisition-rendering systems.**

We demonstrate our system optimization approach for image and video compression using the state-of-the-art HEVC standard, showing significant gains by the adjustment to various acquisition-rendering system structures. The examined layouts include an acquisition model of a linear low-pass filter and an additive white noise preceding the compression, and a rendering stage modeled as a linear operator applied on the decompressed signal. We further considered a multimedia distribution network where the user display devices are modeled as several linear blur operators, each associated with its probability to be used. As another application, we used our methodology to pre-compensate for the post-decompression motion-blur perceived while viewing videos on LCD devices.

4. **A new viewpoint on the information-theoretic problem of remote source coding.**

Our system-optimized compression framework provides a new perspective on the classic information-theoretic problem of remote source coding (also known as noisy source coding). The contribution in this aspect stems from our deterministic settings (contrasting the usual information theoretic statistical frameworks), promoting a new distortion metric resembling fidelity terms used in restoration methods of constrained deconvolution forms. The iterative optimization nature of our method and the utilization of standard compression designs also shed a new light on the remote source coding problem. In addition, we explained the main concepts of our method using a rate-distortion theoretic analysis for the case of a Gaussian signal and linear shift-invariant system operators.

5. **New methods and analysis for restoration based on compression.**

In this section we explored the topic of complexity-regularized restoration, where the likelihood of candidate estimates are determined by their compression bit-costs. Using our ADMM-based approach we developed three practical methods for restoration using standard compression techniques. Two of the proposed methods rely on a new shift-invariant complexity regularizer, evaluating the total bit-cost of the signal shifted versions. We explained few of the main ideas of our approach using an insightful theoretical analysis of complexity-regularized restoration of a Gaussian signal from deterioration of a linear shift-invariant operator and additive white Gaussian noise. Experiments for deblurring and inpainting of images using the JPEG2000 and HEVC technique showed good results.

6. **A denoising-based postprocessing approach for decompressed images.**

In the last section of this thesis, we depart from the main proposed methodology and present a novel postprocessing method for reducing artifacts in compressed
images. The task was 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 experiments showing impressive gains, that improve upon state-of-the-art postprocessing results for leading image compression standards.

8.2 Future Research Directions

In this thesis we established several new general ideas and promising research paths that can be further explored. We believe that the proposed iterative-optimization approach for challenging compression problems can be further developed to form a significant alternative to the Lagrangian rate-distortion optimization [24, 25] that is widely used in contemporary image and video compression designs [26, 27, 28, 29]. To achieve this, the research proposed in this thesis should be extended in two paths: first, studying additional compression problems that cannot be addressed using the formerly known rate-distortion optimization techniques. Such problems may consider optimizing compression with respect to perceptual distortion metrics or to the commonly used postfiltering enhancement filters (e.g., optimizing compression with respect to deblocking filter applied after decompression). The second suggested path may include theoretical and practical studies of the iterative nature of the proposed approach, analyzing convergence properties and parameter accuracy importance. In addition, the benefits and shortcomings of using iterative optimization strategies other than ADMM can be explored.

Our system-optimized compression framework can be also further extended to other architectures involving compression. One may study systems containing non-linear or locally-varying operators. Moreover, our demonstrations for acquisition-rendering systems of visual signals can be developed also for other signal types (e.g., audio signals).

Another research path established in this thesis and worth further exploration is concerned with the utilization of compression for the purpose of restoration, and vice versa. We showed that standard compression techniques can be directly integrated as a main building block of a restoration process, and also that existing denoising methods can be directly employed in a post-decompression processing stage to improve compression. In addition, our system-aware compression approach relies on repeated $\ell_2$-constrained deconvolutions hinting on possible future improvements by using efficient restoration methods instead of the simple deconvolution process. We believe that further
research along this line may reveal additional optimization structures leading to valuable compression and restoration methods.
Appendix A

A Tree-based Coding Method for One-Dimensional Signals

We consider a coding procedure that relies on a nonuniform segmentation of the signal based on a binary tree structure. The method presented here is influenced by the general framework given in [25] for optimizing tree-structures, and by the rate-distortion Lagrangian optimization in [24, 26].

We consider the coding of a $M$-length vector $w \in \mathbb{R}^M$, where $M = 2^{d_0}$ for some positive integer $d_0$. The procedure starts with a full $d$-depth binary-tree ($d \leq d_0$), which is the initial tree, describing a uniform partitioning of the vector components $w_k$, $k = 0, ..., M - 1$, into $2^d$ sub-vectors of $M \cdot 2^{-d}$ length. The segmentation of the vector is represented by the leaves of the binary tree: the sub-vector location and length are determined by the leaf place in the tree, in particular, the sub-vector length is defined by the tree-level that the leaf belongs to. The examined nonuniform segmentations are induced by all the trees obtained by repeatedly pruning neighboring-leaves having the same parent node. The initial $d$-depth full-tree together with all its pruned subtrees form the set of relevant trees, denoted here as $T_d$.

The leaves of a tree $T \in T_d$ form a set denoted as $L(T)$, where the number of leaves is referred to as $|L(T)|$. Accordingly, the tree $T$ represents a (possibly) nonuniform partitioning of the $M$-length vector into $|L(T)|$ segments. A leaf $l \in L(T)$ resides in the $h(l)$ level of the tree and corresponds to the indexing interval $[a_{left}(l), a_{right}(l)]$ of length $\Delta(l) = M \cdot 2^{-h(l)}$. A segment, corresponding to the leaf $l \in L(T)$, is represented by its average value

$$\hat{w}(l) = \frac{1}{\Delta(l)} \sum_{k=a_{left}(l)}^{a_{right}(l)} w_k$$

that is further uniformly quantized using $q_b = 8$ bits. The quantized sample corresponding to the $l$th leaf (segment) is denoted as $\hat{w}^Q(l)$. This coding structure leads to reconstruction squared-error induced by the tree $T \in T_d$ and calculated based on its
leaves, $L(T)$, via

$$E^2 (T) = \sum_{l \in L(T)} \sum_{k \in a^r_{(l)}} (w_k - \hat{w}^Q_{(l)})^2.$$  \hfill (A.2)

For a given signal $w \in \mathbb{R}^M$ and a budget of $\rho$ bits, one can formulate the optimization of a tree-structured nonuniform coding as

$$\min_{T \in T_d} E^2 (T)$$
$$\text{subject to } q_b |L(T)| = \rho,$$  \hfill (A.3)

namely, the optimization searches for the tree associated with a bit-cost of $\rho$ bits that provides minimal reconstruction squared-error. The unconstrained Lagrangian form of (A.3) is

$$\min_{T \in T_d} \{ E^2 (T) + \nu (q_b |L(T)|) \} ,$$  \hfill (A.4)

where $\nu \geq 0$ is a Lagrange multiplier that corresponds to $q_b |L(T)| = \rho$. However, it should be noted that due to the discrete nature of the problem such $\nu$ does not necessarily exist for any $\rho$ value (see details, e.g., in [25, 128]). The problem (A.4) can also be written as

$$\min_{T \in T_d} \left\{ \sum_{l \in L(T)} \sum_{k \in a^r_{(l)}} (w_k - \hat{w}^Q_{(l)})^2 + \nu q_b |L(T)| \right\} .$$  \hfill (A.5)

Importantly, since the representation intervals do not overlap, the contribution of a leaf, $l \in L(T)$, to the Lagrangian cost is

$$C (l) = \sum_{k \in a^r_{(l)}} (w_k - \hat{w}^Q_{(l)})^2 + \nu q_b.$$  \hfill (A.6)

The discrete optimization problem (A.5) of optimizing the tree for a given signal and a Lagrange multiplier $\nu$ is practically addressed as follows. Start from the full $d$-depth tree and determine the corresponding segments and their quantized samples, squared errors, and contributions to the Lagrangian cost (A.6). Go through the tree levels from bottom and up, in each tree level find the pairs of neighboring leaves having the same parent node and evaluate the pruning condition: if

$$C (\text{left child}) + C (\text{right child}) > C (\text{parent})$$  \hfill (A.7)

is true, then prune the two leaves – implying that two segments are merged to form...
a single sub-vector of double length (thus, the total bit-cost is reduced by \( q_b \)). If the condition (A.7) is false, then the two leaves (and the associated representation segments) are kept. This evaluation is continued until reaching a level where no pruning is done, or when arriving to the root of the tree.
Bibliography


[39] RTINGS.com, “Motion blur of TVs,”


[50] F. Bellard, “BPG 0.9.6,”


[54] G. Valenzise, “Bjontegaard metric (Matlab function),”


[60] “Xiph.org video test media,”


[98] T. Tillo and G. Olmo, “Data-dependent pre- and postprocessing multiple

artifacts reduction using variational methods : Algorithms and experimental
study,” in IEEE International Conference on Acoustics, Speech and Signal

artifacts of wavelet-based image,” in IEEE International Conference on

tion,” Signal Processing: Image Communication, vol. 27, no. 6, pp. 663–677,
2012.

[102] A. Zakhor, “Iterative procedures for reduction of blocking effects in transform
pp. 91–95, 1992.

to reduce blocking artifacts of block discrete cosine transform compressed
432, 1993.

[104] A. Nosratinia, “Enhancement of JPEG-compressed images by re-application
of jpeg,” Journal of VLSI signal processing systems for signal, image and

[105] A. Nosratinia, “Postprocessing of JPEG-2000 images to remove compression
2003.

for high-quality denoising and deblocking of grayscale and color images,”

reduction by overlapped-block transform coefficient estimation with block
2013.

based block-shift filtering for deblocking and deringing,” Journal of Visual


[123] “Kakadu software 7.7,”


For Gaussian symbols. In addition, we present a series of rate-distortion theory experiments that demonstrate the effectiveness of the method in reducing the bit-rates for optimization purposes. We show that our method outperforms existing techniques in terms of subjective quality improvements for JPEG2000 and HEVC, using standard compression techniques. We also analyze the potential of our method for image deblurring, showing its ability to reconstruct images with high-quality results. Heuristics and complexity issues are addressed within the framework of our proposed regularization scheme, which is based on the analysis of bit-rates and Gaussian symbols. The method is validated through subjective experiments, showing its potential for real-world applications. Technion - Computer Science Department - Ph.D. Thesis PHD-2018-13 - 2018.
תקציר

דחיסה ושחזור זה שני פעולות המוגנות בבסיסה של תחומי מידע ענびים. דחיסה זו היא פונקציה מפוארת על תכונה מבנית ו多媒体ית (lossy compression), בה היא מאפשרת חסימה של תכונה מספרית המחזורית ותتحقכים בدخولיה. דחיסה זו היא קביעה של מידע על תכונה מבניתphemライン (trade-off) של מבנה המידע. דחיסה זו נשענת על חסימה של תכונה מספרית מחזורית ותتحقכים בدخولיה, ומ과정 השליטה על תכונה מבניתphemライン (trade-off) של מבנה המידע.

בחזרה עד דיוקים ושגיאות המסייעות בהקטנת כמות המידע הנדרשת ליצירת תכונה מבניתphemライン (trade-off) של מבנה מידע. דחיסה זו היא קביעה של נתונים של תכונה מבניתphemライン (trade-off) של מבנה מידע. דחיסה זו נשענת על חסימה של תכונה מספרית מחזורית ותتحقכים בدخولיה, ומ processo של קביעה של נתונים של תכונה מבניתphemeline (trade-off) של מבנה מידע. דחיסה זו נשענת על חסימה של תכונה מספרית מחזורית ותتحقכים בدخولיה, ומ processo של קביעה של נתונים של תכונה מבניתphemeline (trade-off) של מבנה מידע. דחיסה זו נשענת על חסימה של תכונה מספרית מחזורית ותتحقכים בدخولיה, ומ processo של קביעה של נתונים של תכונה מבניתphemeline (trade-off) של מבנה מידע. דחיסה זו נשענת על חסימה של תכונה מספרית מחזורית ותتحقכים בدخولיה, ומ processo של קביעה של נתונים של תכונה מבניתphemeline (trade-off) של מבנה מידע. דחיסה זו נשענת על חסימה של תכונה מספרית מחזורית ותتحقכים בدخولיה, ומ processo של קביעה של נתונים של תכונה מבניתphemeline (trade-off) של מבנה מידע. דחיסה זו נשענת על חסימה של תכונה מספרית מחזורית ותتحقכים בدخولיה, ומпроceso של קביעה של נתונים של תכונה מבניתphemeline (trade-off) של מבנה מידע. דחיסה זו נשענת על חסימה של תכונה מספרית מחזורית ותتحقכים בدخولיה, ומпроceso של קביעה של נתונים של תכונה מבניתphemeline (trade-off) של מבנה מידע. דחיסה זו נשענת על חסימה של תכונה מספרית מחזורית ותتحقכים בدخولיה, ומпроceso של קביעה של נתונים של תכונה מבניתphemeline (trade-off) של מבנה מידע. דחיסה זו נשענת על חסימה של תכונה מספרית מחזורית ותتحقכים בدخولיה, ומпроceso של קביעה של נתונים של תכונה מבניתphemeline (trade-off) של מבנה מידע. דחיסה זו נשענת על חסימה של תכונה מספרית מחזורית ותتحقכים בدخولיה, ומпроceso של קביעה של נתונים של תכונה מבניתphemeline (trade-off) של מבנה מידע. דחיסה זו נשענת על חסימה של תכונה מספרית מחזורית ותتحقכים בدخولיה, ומпроceso של קביעה של נתונים של תכונה מבניתphemeline (trade-off) של מבנה מידע. דחיסה זו נשענת על חסימה של תכונה מספרית מחזורית ותتحقכים בدخولיה, ומпроceso של קביעה של נתונים של תכונה מבניתphemeline (trade-off) של מבנה מידע. דחיסה זו נשענת על חסימה של תכונה מספרית מחזורית ותتحقכים בدخولיה, ומпроceso של קביעה של נתונים של תכונה מבניתphemeline (trade-off) של מבנה מידע. דחיסה זו נשענת על חסימה של תכונה מספרית מחזורית ותتحقכים בدخولיה, ומпроceso של קביעה של נתונים של תכונה מבניתphemeline (trade-off) של מבנה מידע. דחיסה זו נשענת על חסימה של תכונה מספרית מחזורית ותتحقכים בدخولיה, ומпроceso של קביעה של נתונים של תכונה מבניתphemeline (trade-off) של מבנה מידע. דחיסה זו נשענת על חסימה של תכונה מספרית מחזורית ותتحقכים בدخولיה, ומпроceso של קביעה של נתונים של תכונה מבניתphemeline (trade-off) של מבנה מידע. דחיסה זו נשענת על חסימה של תכונה מספרית מחזורית ותتحقכים בدخولיה, ומпроceso של קביעה של נתונים של תכונה מבניתphemeline (trade-off) של מבנה מידע. דחיסה זו נשענת על חסימה של תכונה מספרית מחזורית ותتحقכים בدخولיה, ומпроceso של קביעה של נתונים של תכונה מבניתphemeline (trade-off) של מבנה מידע. דחיסה זו נשענת על חסימה של תכונה מספרית מחזורית ותتحقכים בدخولיה, ומпроceso של קביעה של נתונים של תכונה מבניתphemeline (trade-off) של מבנה מידע. דחיסה זו נשענת על חסימה של תכונה מספרית מחזורית ותتحقכים בدخولיה, ומпроceso של קביעה של נתונים של תכונה מבניתphemeline (trade-off) של מבנה מידע. דחיסה זו נשענת על חסימה של תכונה מספרית מחזורית ותتحقכים בدخولיה, ומпроceso של קביעה של נתונים של תכונה מבניתphemeline (trade-off) של מבנה מידע. דחיסה זו נשענת על חסימה של תכונה מספרית מחזורית ותتحقכים בدخولיה, ומпроceso של קביעה של נתונים של תכונה מבניתphemeline (trade-off) של מבנה מידע. דחיסה זו נשענת על חסימה של תכונה מספרית מחזורית ותتحقכים בدخولיה, ומпроceso של קביעה של נתונים של תכונה מבניתphemeline (trade-off) של מבנה מידע. דחיסה זו נשענת על חסימה של תכונה מספרית מחזורית ותتحقכים בدخولיה, ומпроceso של קביעה של נתונים של תכונה מבניתphemeline (trade-off) של מבנה מידע. דחיסה זו נשענת על חסימה של תכונה מספרית מחזורית ותتحقכים בدخولיה, ומпроceso של קביעה של נתונים של תכונה מבניתphemeline (trade-off) של מבנה מידע. דחיסה זו נשענת על חסימה של תכונה מספרית מחזורית ותتحقכים בدخولיה, ומпроceso של קביעה של נתונים של תכונה מבניתphemeline (trade-off) של מבנה מידע. דחיסה זו נשענת על חסימה של תכונה מספרית מחזורית ותتحقכים בدخولיה, ומпроceso של קביעה של נתונים של תכונה מבניתphemeline (trade-off) של מבנה מידע. דחיסה זו נשענת על חסימה של תכונה מספרית מחזורית ותتحقכים בدخولיה, ומпроceso של קביעה של נתונים של תכונה מבניתphemeline (trade-off) של מבנה מידע. דחיסה זו נשענת על חסימה של תכונה מספרית מחזורית ותتحقכים בدخولיה, ומпроceso של קביעה של נתונים של תכונה מבניתphemeline (trade-off) של מבנה מידע. דחיסה זו נשענת על חסימה של תכונה מספרית מחזורית ותتحقכים בدخولיה, ומпроceso של קביעה של נתונים של תכונה מבניתphemeline (trade-off) של מבנה מידע. דחיסה זו נשענת על חסימה של תכונה מספרית מחזורית ותتحقכים בدخولיה, ומпроceso של קביעה של נתונים של תכונה מבניתphemeline (trade-off) של מבנה מידע. דחיסה זו נשענת על חסימה של תכונה מספרית מחזורית ותتحقכים בدخولיה, ומпроceso של קביעה של נתונים של תכונה מבניתphemeline (trade-off) של מבנה מידע. דחיסה זו נשענת על חסימה של תכונה מספרית מחזורית ותتحقכים בدخولיה, ומпроceso של קביעה של נתונים של תכונה מבניתphemeline (trade-off) של מבנה מידע. דחיסה זו נשענת על חסימה של תכונה מספרית מחזורית ותتحقכים בدخولיה, ומпроceso של קביעה של נתונים של תכונה מבניתphemeline (trade-off) של מבנה מידע. דחיסה זו נשענת על חסימה של תכונה מספרית מחזורית ותتحقכים בدخولיה, ומпроceso של קביעה של נתונים של תכונה מבניתphemeline (trade-off) של מבנה מידע. דחיסה זו נשען...
המחקר נערך בהנחיית פרופסור אלפרד ברוקשטיין וד'ר ברוקשטיין מפקולטה למדעי המחשב.

תודעה

ברצוני להביע את תודהי העומק-dismissible' ד'ר ברוקשטיין וד'ר ברוקשטיין מפקולטה למדעי המחשב על הנחיותם המטせות והטיפשות של לדמי ה더טוטור. אני מעריך את שהואolated על המחמות, לעובד עםAndy והד iidim,j את ארנון, כל אחד בנפרד, על כלים שהם של האסфאות והמשניות במוחקך ואת הנאתו במעבדה ולתחבויות רקב מתמיד. אני מעריך את חם马桶ים ומי מטסיים על האקדמיה והמעבדה لبنוסף, ברצוני להודות להמכוות על עובדה ואמורתי על התמונות, על כל שבחותם קורין התמונות.

הסיבות האונטולוגיות למדעים ומקולים.

אני מודע לאיתורים, על כל האספת התמונות להארץ הישנה, הערכים והיוותות שלдумать.

הממחנה היינו חשבוניםивать דבדכי ענבר היה זה.
שיטות חישובmonary לאיתות
וקשריהם לבעיות שחזור

חבור על מחקר

לשמ מיולי חקליל חידישות לכתבת החזון
דוקטור לפילוסופיה

יהודה דר

יווח ל tecnología – מכון טכנולוגי לישראל
תשייר התשע”א – חיפה
2018
שיטות חדשות לדחיסת אוטות
וקשריהם לבחינת שטוחות

יחודת ז"ר