Modular ADMM-Based Strategies for Optimized Compression, Restoration, and Distributed Representations of Visual Data

Yehuda Dar and Alfred M. Bruckstein

Contents

Abstract

Iterative techniques are a well-established tool in modern imaging sciences, allowing to address complex optimization problems via sequences of simpler computational processes. This approach has been significantly expanded in recent years by iterative designs where explicit solutions of optimization sub-problems were replaced by black-box applications of ready-to-use modules for

Y. Dar (✉)
Electrical and Computer Engineering Department, Rice University, Houston, TX, USA
e-mail: ydar@rice.edu

A. M. Bruckstein (✉)
Computer Science Department, Technion – Israel Institute of Technology, Haifa, Israel
e-mail: freddy@cs.technion.ac.il

K. Chen et al. (eds.), *Handbook of Mathematical Models and Algorithms in Computer Vision and Imaging*, https://doi.org/10.1007/978-3-030-03009-4_71-1

denoising or compression. These modular designs are conceptually simple, yet often achieve impressive results. In this chapter, we overview the concept of modular optimization for imaging problems by focusing on structures induced by the alternating direction method of multipliers (ADMM) technique and their applications to intricate restoration and compression problems. We start by emphasizing general guidelines independent of the module type used and only then derive ADMM-based structures relying on denoising and compression methods. The wide perspective on the topic should motivate extensions of the types of problems addressed and the kinds of black boxes utilized by the modular optimization. As an example for a promising research avenue, we present our recent framework employing black-box modules for distributed representations of visual data.

Keywords

Modular optimization · Alternating direction method of multipliers (ADMM) · Inverse problems · Signal compression · Distributed representations

Introduction

During the last several decades, significant attention and efforts were invested in establishing solutions for a wide variety of imaging problems. The proposed methods often rely on models and techniques adapted to visual signals and the relevant problem settings. Naturally, along the contemporary challenges and open questions of the field, there are excellent solutions to various fundamental problems that were extensively studied throughout the years. This situation suggests addressing currently open problems by exploring their relations to existing methods developed for basic tasks.

A lot of work has been devoted to fundamental problems such as denoising of a single image contaminated by additive white Gaussian noise and lossy compression of still images with respect to squared errors as quality assessment measures. Persistent and thorough studies of such basic problems (in their classical settings) led to excellent solutions that are believed to be nearly perfect (see, e.g., Chatterjee and Milanfar 2009). However, the techniques for many other imaging tasks are in various degrees of maturity that leave room for possibly considerable improvements. Examples for types of currently active research lines include jointly addressing multiple imaging tasks (Burger et al. 2018; Corona et al. 2019a,b; Dar et al. 2018a,b,c,d), restoration with uncertainty about the degradation operator (Lai et al. 2016; Bahat et al. 2017), image compression with respect to modern perceptual quality measures (Ballé et al. 2017; Laparra et al. 2017), and tasks (also fundamental ones such as denoising and compression) involving visual data beyond a single natural image (this includes video, hyperspectral, medical, etc.).

In this chapter, we overview a recent and fascinating approach for elegant utilization of existing knowledge and available imaging tools for complex problems of interest. The general idea is to define an optimization problem such that when

addressed using a specific iterative optimization technique, the resulting sequential algorithm calls for solving a subproblem corresponding to a fundamental task like denoising or compression. Then, the explicit solutions of the basic subproblem instances may be replaced by black-box applications of available methods, highly perfected over the years due to their prevalence and long-standing importance. Interestingly, the black-box modules utilized do not have to exactly match the formulation of the subproblems they replace, as long as they address the same fundamental task.

The main concept of *modular optimization strategies* described above was preceded by a line of optimization-based iterative algorithms including stages of *explicitly* solving regularized inverse problems, often associated with denoising or maximum a posteriori estimation tasks (see, e.g., Afonso et al. 2010; Zoran and Weiss 2011). Yet, actual employment of image denoisers as black boxes was explicitly proposed only later in the Plug-and-Play Priors framework (Venkatakrishnan et al. 2013; Sreehari et al. 2016), where the alternating direction method of multipliers (ADMM) (Boyd et al. 2011) was used to form iterative structures based on denoising modules to solve inverse imaging problems (specifically, demonstrated by Venkatakrishnan et al. (2013) and Sreehari et al. (2016) for tomographic reconstruction based on the BM3D denoiser (Dabov et al. 2007)). The Plug-and-Play Priors framework (based on ADMM and denoisers) proved very useful to a variety of practical inverse problems (Dar et al. 2016b; Rond et al. 2016; Brifman et al. 2016; Chan et al. 2017; Buzzard et al. 2018; Kwan et al. 2018; Yazaki et al. 2019; Brifman et al. 2019; Ahmad et al. 2019), and its convergence was analyzed for several particular cases (Chan et al. 2017; Chan 2019). Another prominent approach based on denoising modules is the Regularization-by-Denoising (RED) framework (Romano et al. 2017; Hong et al. 2019; Brifman et al. 2019), proposing to regularize the basic problem using the black-box denoising function. Then an efficient sequential procedure based on iterative optimization techniques of ADMM or a fixed-point strategy is called upon, thereby clarifying that modular optimizations can be constructed not only based on ADMM. Other non-ADMM methods using denoisers for restoration or reconstruction problems were proposed based on FISTA for addressing nonlinear problems (Kamilov et al. 2017; Ahmad et al. 2019), primal-dual splitting (Ono 2017), backward projections (Tirer and Giryes 2018a,b, 2019), and ISTA for online updates (Sun et al. 2019a,b). All of these firmly established the wide applicability of denoising-based modular approaches for inverse problems addressing restoration and reconstruction of visual data.

We here consider the modular optimization strategy as a general concept beyond the extensively studied aspect of using denoisers for solving inverse problems. The deviation from the denoising-based modular optimizations started by Dar et al. (2016a, 2018c), and also the related work of Beygi et al. (2017a,b), where inverse problems were addressed based on compression techniques, essentially functioning as complexity regularizers. Specifically, image deblurring and inpainting problems were addressed by Dar et al. (2018c) using JPEG2000 and the state-of-the-art image coding method of the High Efficiency Video Coding (HEVC) standard. Moreover, a shift-invariant regularizer was proposed by Dar et al. (2018c) to amend the limitations of the regular compression-based prior. All these complex problem

structures were treated using the ADMM optimization tool in a Plug-and-Play manner.

Another important generalization is due to a recent research line (Dar et al. 2016a, 2018a,b,c,d), branching out from the original Plug-and-Plug Priors framework, suggesting to address intricate compression and restoration problems based on image and video compression modules applied as black boxes. Importantly, this framework shows that modularity is possible not only for priors and that the basic modules employed can be other than denoisers. Furthermore, using standard compression techniques in modular optimization frameworks extends the range of imaging problems addressed far outside the area of inverse problems. This extension is pursued in (Dar et al. 2018a,b,c,d) where systems involving acquisition, compression, and rendering processes are optimized based on ADMM and standard compression techniques. This established the ability to optimize complex systems while being compatible to prevalent compression standards and without using post-processing – thereby emphasizing on the usefulness of modular optimization strategies to much more than using denoisers as black-box priors or in ready-to-use modules. Specifically, the ADMM-based framework in (Dar et al. 2018a,b,c,d) also exhibits how to address intricate rate-distortion optimizations (a fundamental concept in modern compression techniques (Shoham and Gersho 1988; Ortega and Ramchandran 1998; Sullivan and Wiegand 1998)) by decoupling the challenging distortion metric from the actual compression task, consequently enabling the use of standard techniques as modules. Indeed, this idea inspired the nice work reported by Rott Shaham and Michaeli (2018) where an alternating minimization process is used to decouple a perceptual distortion metric from a standard compression technique – thus externally adding desired perceptual aspects into a standardized compression method.

We further point out here on a new direction of developing modular optimizations for distributed representations. In general, ADMM is a technique for distributed optimization, and, therefore, it is natural to utilize its valuable decoupling ability also for optimizations aimed at distributed representations. Specifically, we suggest to employ black-box modules for creating multiple descriptions of a given signal. Therefore, we overview our recent work (Dar and Bruckstein 2021) on holographic compression of images, where standard image compression techniques are adjusted to settings of duplication-based storage systems. The idea is to create a set of standard-compatible representations, all of them being equally important in refining the data reconstruction. We conclude by discussing the general implications of modular optimizations to distributed tasks.

Modular ADMM-Based Optimization: General Construction and Guidelines

Unconstrained Lagrangian Optimizations via ADMM

Consider an arbitrary optimization problem of an unconstrained Lagrangian form, namely,

$$\hat{\mathbf{v}} = \underset{\mathbf{v}\in\mathcal{M}}{\operatorname{argmin}} \; R(\mathbf{v}) + \lambda D(\mathbf{x}, \mathbf{v}) \tag{1}$$

to be solved for the optimization variable $\mathbf{v} \in \mathcal{M} \subset \mathbb{R}^N$, where $\mathcal{M}$ is a (continuous or discrete) subset of the N-dimensional real space. Moreover, the optimization (1) is defined for a given column vector $\mathbf{x} \in \mathbb{R}^M$. In this section, we refer to general scalar-valued functions satisfying $R : \mathcal{M} \to \mathbb{R}$ and $D : \mathbb{R}^M \times \mathbb{R}^N \to \mathbb{R}$. In the sequel discussing the applications for restoration and compression tasks, the general definitions given here take the following form. For restoration tasks, posed as inverse problems, $\mathcal{M}$ is set to be $\mathbb{R}^N$, and the functions R and D implement regularization and fidelity terms, respectively. In the case of compression, $\mathcal{M}$ is a discrete set of decompressed signals supported by the compression architecture, and the functions R and D measure bit-cost and distortion, respectively. While restoration and compression problems introduce various mathematical forms to the general optimization (1), we here address this general structure.

Particular instances of the problem (1) often take challenging forms that require significant engineering and/or computational resources. Addressing a new problem may require the design and implementation of a complete algorithm from scratch, ignoring existing knowledge and tools from potentially related problems. Then, computational difficulties may arise due to high dimensionality of specific instances of (1) such that direct solutions become very costly or even impractical. Such reasons motivate the translation of (1) into a tractable procedure addressing the original task, sometimes in an approximated manner, while avoiding the complications mentioned above. A prominent approach for such designs is described next.

The alternating direction method of multipliers (ADMM) technique (Boyd et al. 2011) is a popular tool for addressing the potentially challenging problem (1). For this we start by splitting the optimization variable such that (1) becomes

$$\begin{aligned} \hat{\mathbf{v}} = \; & \underset{\mathbf{v}\in\mathcal{M},\mathbf{z}\in\mathbb{R}^N}{\operatorname{argmin}} \quad R(\mathbf{v}) + \lambda D(\mathbf{x}, \mathbf{z}) \qquad\qquad f \\ & \text{subject to} \quad \mathbf{v} = \mathbf{z} \end{aligned} \tag{2}$$

where $\mathbf{z} \in \mathbb{R}^N$ is an auxiliary variable that is not directly constrained to the domain $\mathcal{M}$. Next, we apply the scaled form of the augmented Lagrangian and the method of multipliers (Boyd et al. 2011, Ch. 2) on (2) and obtain the iterative procedure

$$\left(\hat{\mathbf{v}}^{(t)}, \hat{\mathbf{z}}^{(t)}\right) = \underset{\mathbf{v}\in\mathcal{M},\mathbf{z}\in\mathbb{R}^N}{\operatorname{argmin}} \; R(\mathbf{v}) + \lambda D(\mathbf{x}, \mathbf{z}) + \frac{\beta}{2}\left\|\mathbf{v} - \mathbf{z} + \mathbf{u}^{(t)}\right\|_2^2 \tag{3}$$

$$\mathbf{u}^{(t+1)} = \mathbf{u}^{(t)} + \left(\hat{\mathbf{v}}^{(t)} - \hat{\mathbf{z}}^{(t)}\right), \tag{4}$$

where t denotes the iteration index, $\mathbf{u}^{(t)} \in \mathbb{R}^N$ is the scaled dual variable, and β is an auxiliary parameter introduced by the augmented Lagrangian. Then, the ADMM form of the problem is derived by applying one iteration of alternating minimization

on (3), yielding a series of simpler optimizations:

$$\hat{\mathbf{v}}^{(t)} = \underset{\mathbf{v}\in\mathcal{M}}{\operatorname{argmin}}\, R\left(\mathbf{v}\right) + \frac{\beta}{2}\left\|\mathbf{v} - \tilde{\mathbf{z}}^{(t)}\right\|_2^2 \tag{5}$$

$$\hat{\mathbf{z}}^{(t)} = \underset{\mathbf{z}\in\mathbb{R}^N}{\operatorname{argmin}}\, \lambda D\left(\mathbf{x}, \mathbf{z}\right) + \frac{\beta}{2}\left\|\mathbf{z} - \tilde{\mathbf{v}}^{(t)}\right\|_2^2 \tag{6}$$

$$\mathbf{u}^{(t+1)} = \mathbf{u}^{(t)} + \left(\hat{\mathbf{v}}^{(t)} - \hat{\mathbf{z}}^{(t)}\right) \tag{7}$$

where $\tilde{\mathbf{z}}^{(t)} = \hat{\mathbf{z}}^{(t-1)} - \mathbf{u}^{(t)}$ and $\tilde{\mathbf{v}}^{(t)} = \hat{\mathbf{v}}^{(t)} + \mathbf{u}^{(t)}$. Importantly, in the last ADMM-based structure, the possibly nontrivial domain $\mathcal{M}$ and the related function R are decoupled from the second, perhaps intricate, function D. Accordingly, the new subtasks in the process are much simpler. Specifically, note that (6) is a continuous optimization problem over $\mathbb{R}^N$, regardless of the original domain of problem (1) that may be even discrete. Note that in the general case, where R, D, and $\mathcal{M}$ can induce non-convexity and discreteness to the problem, there are no convergence guarantees corresponding to the ADMM process formulated above, and its usefulness should be evaluated empirically. However, this common practice has already provided many useful methods for various applications, and selected examples of those are presented in sections "Image Restoration Based on Denoising Modules" and "Modular Optimizations Based on Standard Compression Techniques".

Employing Black-Box Modules

While the ADMM form in (5), (6), and (7) indeed seems easier to carry out than a complex instance of (1), the explicit definition and deployment of $\mathcal{M}$ and/or R in the optimization stage (5) may still require some engineering efforts (such as design, implementation, etc.). In the case of restoration tasks, this means detailed definitions and implementations of regularization functions. For compression architectures, one should establish binary compressed representations matching signal-domain reconstructions. As explained next, the fundamental idea of using black-box modules is to avoid explicit treatment of such details and still achieve excellent, or even state-of-the-art, results with respect to the actual goal.

The main guideline when addressing a problem based on modular optimization strategies is to *formulate the initial optimization problem* (in our case, an instance of (1)) and *choose an iterative optimization technique* (here, ADMM) *such that* the *resulting sequential process* includes:

- A stage corresponding to a basic problem, having well-established solutions readily available to use. In the developments presented here, we ask to replace the optimization stage (5) with a module applied as a black box and, by that, encapsulating the various aspects of the original problem domain $\mathcal{M}$ and function

R. Now, if (5) can be identified as a prototype formulation corresponding to a fundamental problem (e.g., denoising, compression), then one can replace the direct treatments of (5) with application of a module addressing the same basic problem – possibly based on another formulation or even an algorithm that does not correspond to an explicit mathematical expression. Such module is applied as a black box and denoted here as

$$\hat{\mathbf{v}}^{(t)} = BlackBoxModule\left(\tilde{\mathbf{z}}^{(t)}; \theta\left(\beta\right)\right) \tag{8}$$

where $\theta\left(\beta\right)$ (which will be denoted from now on as θ) is a parameter generalizing the role of the Lagrange multiplier β in determining the *implicit* trade-off between the components appeared in (5) before the replacement with the module. The generic method is summarized in Algorithm 1, where the number of parameters is reduced based on the relation $\tilde{\beta} \triangleq \frac{\beta}{2\lambda}$ such that only the parameters θ and $\tilde{\beta}$ are required as inputs for the method (for simplicity, we do not use the fact that both θ and $\tilde{\beta}$ originally depend on β).

- A subproblem considering the distance function D while having a form that can be practically solved. This refers here to subproblem (6). In many interesting applications, the distance function is a particular case of

$$D\left(\mathbf{x}, \mathbf{z}\right) = \sum_{j=1}^{K} \alpha_j \left\|\mathbf{A}_j\mathbf{x} - \mathbf{B}_j\mathbf{z}\right\|_2^2 \tag{9}$$

for some positive integer K, positive real values $\left\{\alpha_j\right\}_{j=1}^{K}$, and matrices $\left\{\mathbf{A}_j\right\}_{j=1}^{K} \in \mathbb{R}^{\tilde{N}\times M}$, $\left\{\mathbf{B}_j\right\}_{j=1}^{K} \in \mathbb{R}^{\tilde{N}\times N}$. Then, for the form (9), the optimization step is a least squares problem that can be easily addressed for many structures of matrices $\left\{\mathbf{A}_j\right\}_{j=1}^{K} \in \mathbb{R}^{\tilde{N}\times M}$ and $\left\{\mathbf{B}_j\right\}_{j=1}^{K} \in \mathbb{R}^{\tilde{N}\times N}$.

One should note that the modular optimization process in Algorithm 1 provides a result that is an output of the black-box module applied in the last iteration. This eventual output can be the signal $\hat{\mathbf{v}}^{(t)} \in \mathcal{M}$ produced by the module at the last iteration and/or other relevant data possibly outputted by the module. This structure is useful, for example, in the case of compression where the important output is a binary compressed representation (i.e., a direct output of the module which is coupled with the signal $\hat{\mathbf{v}}^{(t)} \in \mathcal{M}$). Various applications may benefit from an alternative application that is described next.

Another Splitting Structure

We now turn to describe the construction of a process mirroring Algorithm 1 and utilized often for restoration and reconstruction problems. For the developments

Algorithm 1 General Modular Optimization – Type I: Overall Results Are Module Outputs

Inputs: $\mathbf{x}$, θ, $\tilde{\beta}$.
Initialize $t = 0$, $\hat{\mathbf{z}}^{(0)} = \mathbf{x}$, $\mathbf{u}^{(1)} = \mathbf{0}$.
repeat
  $t \leftarrow t + 1$
  $\tilde{\mathbf{z}}^{(t)} = \hat{\mathbf{z}}^{(t-1)} - \mathbf{u}^{(t)}$
  $\hat{\mathbf{v}}^{(t)} = BlackBoxModule\left(\tilde{\mathbf{z}}^{(t)}; \theta\right)$
  $\tilde{\mathbf{v}}^{(t)} = \hat{\mathbf{v}}^{(t)} + \mathbf{u}^{(t)}$
  $\hat{\mathbf{z}}^{(t)} = \underset{\mathbf{z}\in\mathbb{R}^N}{\text{argmin}}\, D(\mathbf{x}, \mathbf{z}) + \tilde{\beta}\left\|\mathbf{z} - \tilde{\mathbf{v}}^{(t)}\right\|_2^2$
  $\mathbf{u}^{(t+1)} = \mathbf{u}^{(t)} + \left(\hat{\mathbf{v}}^{(t)} - \hat{\mathbf{z}}^{(t)}\right)$
until stopping criterion is satisfied
Output: $\hat{\mathbf{v}}^{(t)}$ and/or other application-specific outputs of $BlackBoxModule$.

overviewed, here, we assume that the output domain of the basic optimization problem (1) satisfies $\mathcal{M} = \mathbb{R}^N$.

The alternative process stems from a delicate difference in the variable splitting applied on the basic problem, namely, instead of (2), we write

$$\begin{aligned} \hat{\mathbf{v}} = \quad & \underset{\mathbf{v}\in\mathbb{R}^N, \mathbf{z}\in\mathbb{R}^N}{\text{argmin}} \quad R(\mathbf{z}) + \lambda D(\mathbf{x}, \mathbf{v}) \\ & \text{subject to} \quad \mathbf{v} = \mathbf{z} \end{aligned} \tag{10}$$

where $\mathbf{z} \in \mathbb{R}^N$ is an auxiliary variable used here to replace the occurrence of $\mathbf{v}$ as the argument of R, whereas the function D still refers to $\mathbf{v}$ (note the difference from the variable splitting described in (2)). Then, similarly to section "Unconstrained Lagrangian Optimizations via ADMM", further developing (10) using the scaled form of the augmented Lagrangian, the method of multipliers, and alternating minimization gives

$$\hat{\mathbf{v}}^{(t)} = \underset{\mathbf{v}\in\mathbb{R}^N}{\text{argmin}}\, \lambda D(\mathbf{x}, \mathbf{v}) + \frac{\beta}{2}\left\|\mathbf{v} - \tilde{\mathbf{z}}^{(t)}\right\|_2^2 \tag{11}$$

$$\hat{\mathbf{z}}^{(t)} = \underset{\mathbf{z}\in\mathbb{R}^N}{\text{argmin}}\, R(\mathbf{z}) + \frac{\beta}{2}\left\|\mathbf{z} - \tilde{\mathbf{v}}^{(t)}\right\|_2^2 \tag{12}$$

$$\mathbf{u}^{(t+1)} = \mathbf{u}^{(t)} + \left(\hat{\mathbf{v}}^{(t)} - \hat{\mathbf{z}}^{(t)}\right) \tag{13}$$

where $\tilde{\mathbf{z}}^{(t)} = \hat{\mathbf{z}}^{(t-1)} - \mathbf{u}^{(t)}$ and $\tilde{\mathbf{v}}^{(t)} = \hat{\mathbf{v}}^{(t)} + \mathbf{u}^{(t)}$. Note that the current procedure in (11), (12), and (13) includes the same subproblems as in (5), (6), and (7) but in a different order (and also up to the setting $\mathcal{M} = \mathbb{R}^N$ used in this subsection).

Like in section "Employing Black-Box Modules", we identify the stage considering the function R, here in (12), as a solution to a fundamental problem that

can be replaced by an available black-box implementation. This yields the process described in Algorithm 2. Note that the result of the procedure is not a direct output of the black-box module. This delicate change with respect to Algorithm 1 may lead to improved results in various applications such as image restoration (where the black-box module is utilized for regularization purposes and, in practice, it is often better not to use its output directly as the result of the entire procedure).

Algorithm 2 General Modular Optimization – Type II: Overall Results Are Not Module Outputs

Inputs: $\mathbf{x}$, θ, $\tilde{\beta}$.
Initialize $t = 0$, $\hat{\mathbf{z}}^{(0)} = \mathbf{x}$, $\mathbf{u}^{(1)} = \mathbf{0}$.
repeat
  $t \leftarrow t + 1$
  $\tilde{\mathbf{z}}^{(t)} = \hat{\mathbf{z}}^{(t-1)} - \mathbf{u}^{(t)}$
  $\hat{\mathbf{v}}^{(t)} = \underset{\mathbf{v}\in\mathbb{R}^N}{\operatorname{argmin}} D(\mathbf{x}, \mathbf{v}) + \tilde{\beta}\left\|\mathbf{v} - \tilde{\mathbf{z}}^{(t)}\right\|_2^2$
  $\tilde{\mathbf{v}}^{(t)} = \hat{\mathbf{v}}^{(t)} + \mathbf{u}^{(t)}$
  $\hat{\mathbf{z}}^{(t)} = BlackBoxModule\left(\tilde{\mathbf{v}}^{(t)}; \theta\right)$
  $\mathbf{u}^{(t+1)} = \mathbf{u}^{(t)} + \left(\hat{\mathbf{v}}^{(t)} - \hat{\mathbf{z}}^{(t)}\right)$
until stopping criterion is satisfied
Output: $\hat{\mathbf{v}}^{(t)}$.

Image Restoration Based on Denoising Modules

In the previous section, we presented the modular optimization approach in its general form, independent of the type of tasks addressed and modules utilized. In this section, we focus on the prevalent application of denoising-based modular optimizations to image restoration problems.

The problem setting considered in this section is defined as follows. A signal $\mathbf{v}_0 \in \mathbb{R}^N$ is going through a degradation process, resulting in the observation $\mathbf{x} \in \mathbb{R}^M$ satisfying

$$\mathbf{x} = \mathbf{H}\mathbf{v}_0 + \mathbf{n}, \tag{14}$$

where $\mathbf{H}$ is a $M \times N$ real matrix and $\mathbf{n}$ is a white Gaussian noise column-vector of length M (the noise components are zero mean and have variance σ_n^2). The restoration task is to estimate the unknown $\mathbf{v}_0$, given $\mathbf{x}$ and the knowledge of the degradation operator $\mathbf{H}$ and the noise variance σ_n^2. For the purpose of restoration, we define the function D from (1) as the fidelity term of the respective inverse problem, namely,

$$D(\mathbf{x}, \mathbf{v}) = \left\|\mathbf{x} - \mathbf{H}\mathbf{v}\right\|_2^2. \tag{15}$$

The ADMM optimization structure based on black-box denoisers, first proposed in the Plug-and-Play Priors design (Venkatakrishnan et al. 2013; Sreehari et al. 2016), mainly stems from associating the function $R : \mathbb{R}^N \to \mathbb{R}$ with a regularizer implemented (explicitly or implicitly) in a ready-to-use denoising process. Then, the optimization for the $\hat{\mathbf{z}}^{(t)}$, appearing in (12), can be interpreted as an inverse problem for denoising $\tilde{\mathbf{v}}^{(t)}$ using the regularizer R. One can also perceive (12) as a maximum a posteriori (MAP) estimation of a signal from its noisy version $\tilde{\mathbf{v}}^{(t)}$, i.e.,

$$\hat{\mathbf{z}}^{(t)} = \underset{\mathbf{z}\in\mathbb{R}^N}{\operatorname{argmax}} \log p_R(\mathbf{z}) + \log p_\eta\left(\tilde{\mathbf{v}}^{(t)} - \mathbf{z}\right) \tag{16}$$

where $p_R(\mathbf{z}) \triangleq exp\left(-R(\mathbf{z})\right)$ is the prior probability function assumed for the clean signal and p_η is the probability density function of an additive Gaussian noise vector η with i.i.d. components having zero mean and $1/\beta$ variance. Accordingly, the correspondence of (12) to denoising problems motivates the usage of black-box denoisers as the modules applied at stage 8 of Algorithm 2. These denoisers should be set to remove noise having variance of $1/\beta$ from the signal $\tilde{\mathbf{v}}^{(t)}$. Importantly, the substitution of (12) with applications of Gaussian denoisers was experimentally shown beneficial also for denoisers that do not follow the MAP estimation form or the regularized inverse problem approach. Specifically, one can even employ algorithmic denoisers that were designed based on completely different mindsets. The denoising-based restoration procedure for an arbitrary degradation operator $\mathbf{H}$ is summarized in Algorithm 3.

The decoupling induced by the ADMM structure leads to an additional conceptual simplification: stage 6 of Algorithm 3 can be interpreted as a ℓ_2-constrained deconvolution problem (or ℓ_2-regularized least squares computation) with respect to the degradation operator $\mathbf{H}$. Note that this is one of the simplest restoration formulations addressing the degradation process (14) from the regularized inverse-problem perspective. The corresponding analytic solution is

$$\hat{\mathbf{v}}^{(t)} = \left(\mathbf{H}^T\mathbf{H} + \tilde{\beta}\mathbf{I}\right)^{-1}\left(\mathbf{H}^T\mathbf{x} + \tilde{\beta}\tilde{\mathbf{z}}^{(t)}\right). \tag{17}$$

Alternatively, this computation can be numerically applied in various tractable ways (depending on the specific structure of $\mathbf{H}$). In summary, the overall modular restoration process relies on sequential application of conceptually simple tasks: Gaussian denoising and ℓ_2-constrained deconvolution.

Figures 1 and 2 show typical results obtained using the Plug-and-Play method, implemented in the code published with Chan et al. (2017), based on the BM3D denoiser (Dabov et al. 2007). The deblurring settings (Fig. 1) include a blur operator corresponding to a 9×9 pixels convolution kernel (Gaussian with standard deviation 1.75) and additive white Gaussian noise of standard deviation 10. The inpainting experiment (Fig. 2) considers 80% missing pixels and additive white Gaussian noise of standard deviation 10. Specifically, note the improvement in the PSNR of the intermediate estimates, $\hat{\mathbf{v}}^{(t)}$, evolving throughout the process iterations until

Algorithm 3 Restoration Based on Denoising Modules

Inputs: $\mathbf{x}$, θ, $\tilde{\beta}$.
Initialize $t = 0$, $\hat{\mathbf{z}}^{(0)} = \mathbf{x}$, $\mathbf{u}^{(1)} = \mathbf{0}$.
repeat
  $t \leftarrow t + 1$
  $\tilde{\mathbf{z}}^{(t)} = \hat{\mathbf{z}}^{(t-1)} - \mathbf{u}^{(t)}$
  $\hat{\mathbf{v}}^{(t)} = \underset{\mathbf{v}\in\mathbb{R}^N}{\operatorname{argmin}} \left\|\mathbf{x} - \mathbf{H}\mathbf{v}\right\|_2^2 + \tilde{\beta}\left\|\mathbf{v} - \tilde{\mathbf{z}}^{(t)}\right\|_2^2$
  $\tilde{\mathbf{v}}^{(t)} = \hat{\mathbf{v}}^{(t)} + \mathbf{u}^{(t)}$
  $\hat{\mathbf{z}}^{(t)} = Denoiser\left(\tilde{\mathbf{v}}^{(t)}; \theta\right)$
  $\mathbf{u}^{(t+1)} = \mathbf{u}^{(t)} + \left(\hat{\mathbf{v}}^{(t)} - \hat{\mathbf{z}}^{(t)}\right)$
until stopping criterion is satisfied
Output: $\hat{\mathbf{v}}^{(t)}$.

practical convergence (Figs. 1d and 2d). See Chan et al. (2017) for details and analysis of the convergence appearing here.

Modular Optimizations Based on Standard Compression Techniques

In this section, we overview the utilization of compression modules for restoration and challenging compression purposes. The use of modules beyond denoisers further establishes the modularity property as a general idea, relevant to various tasks.

Preliminaries: Lossy Compression via Operational Rate-Distortion Optimization

Consider a signal, $\mathbf{x} \in \mathbb{R}^N$, to be compressed and represented as a sequence of bits. We describe a lossy compression procedure as the function

$$C : \mathbb{R}^N \rightarrow \mathcal{B}, \tag{18}$$

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

$$\mathbf{b} = C\left(\mathbf{x}\right), \tag{19}$$

where $\mathbf{b} \in \mathcal{B}$ is the binary compressed data useful for storage or transmission. Then, a matching decompression process gets the compressed data $\mathbf{b}$ as its input and reconstructs a signal-domain representation via

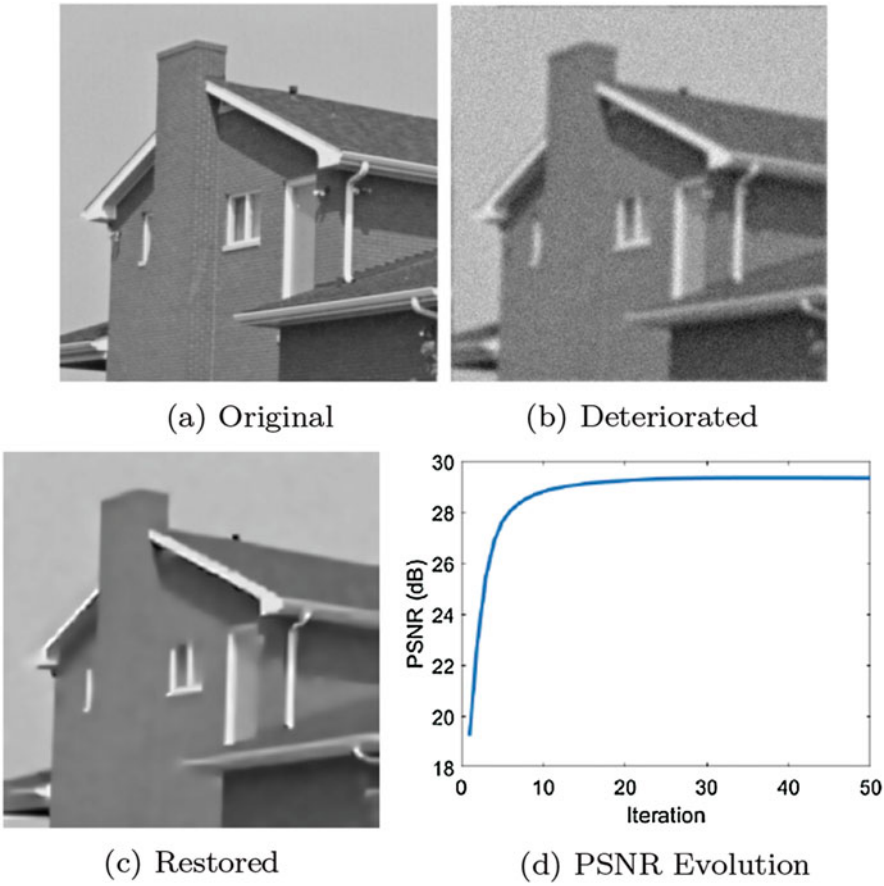

Fig. 1 Deblurring using denoising-based Plug-and-Play method (Chan et al. 2017). The utilized denoiser is BM3D (Dabov et al. 2007). The degradation includes a Gaussian blur (of 9×9 pixels kernel and 1.75 standard deviation), followed by additive noise with $\sigma_n = 10$, applied on the House image (256×256 pixels). (**a**) The original image. (**b**) Deteriorated image. (**c**) Restored image using the method by Chan et al. (2017) (29.33 dB). (**d**) The PSNR evolution of the intermediate estimate $\hat{\mathbf{v}}^{(t)}$ along the restoration-process iterations

$$\mathbf{v} = F\left(\mathbf{b}\right), \tag{20}$$

where

$$F : \mathcal{B} \to \mathcal{S} \tag{21}$$

maps the binary compressed representations in $\mathcal{B}$ to their corresponding decompressed signals from the discrete set $\mathcal{S} \subset \mathbb{R}^N$. The decompressed signal $\mathbf{v}$ can be

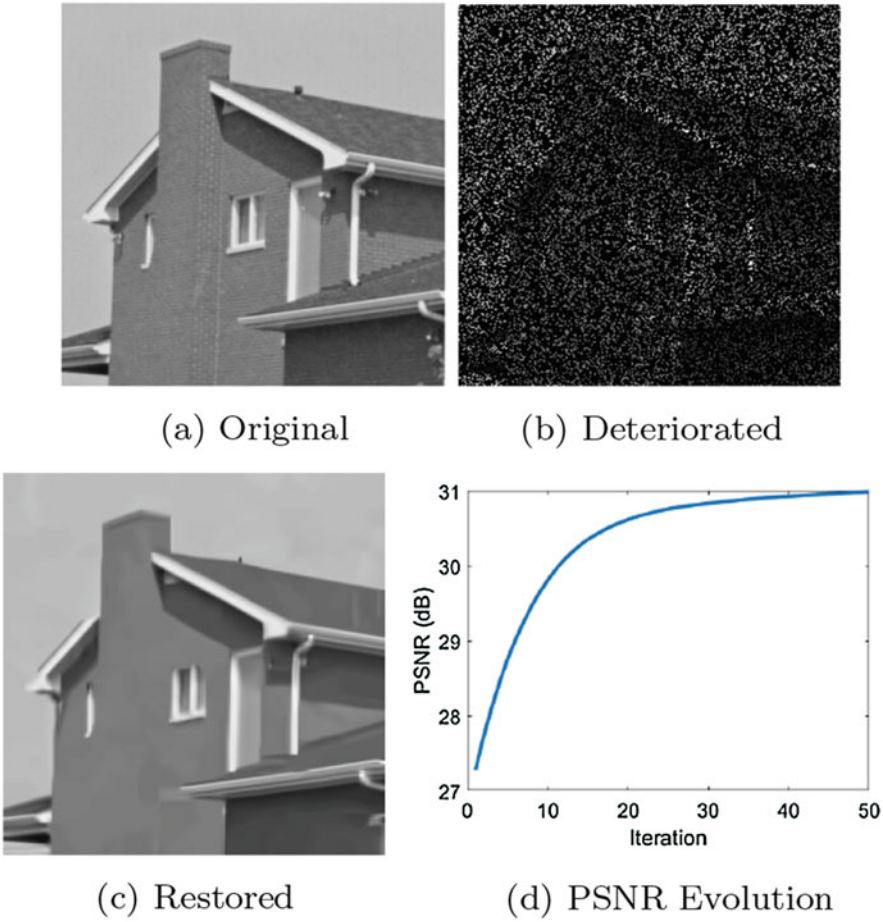

Fig. 2 Inpainting using denoising-based Plug-and-Play method (Chan et al. 2017). The employed denoiser is BM3D (Dabov et al. 2007). The degradation includes 80% missing pixels and additive noise with $\sigma_n = 10$, applied on the House image (256×256 pixels). (**a**) The original image. (**b**) Deteriorated image. (**c**) Restored image using the method by Chan et al. (2017) (30.98 dB). (**d**) The PSNR evolution of the intermediate estimate $\hat{\mathbf{v}}^{(t)}$ along the iterations

further processed or outputted to a user. For example, in the case of visual signals, $\mathbf{v}$ is usually displayed.

Modern compression architectures (see, e.g., Ortega and Ramchandran 1998; Sullivan and Wiegand 1998; Shukla et al. 2005; Sullivan et al. 2012) implement the compression function C using operational rate-distortion optimizations, a tool established by Shoham and Gersho (1988), Chou et al. (1989), and Ortega and Ramchandran (1998), and can be explained using our notions as follows. A given deterministic signal $\mathbf{x}$ is compressed based on an optimization process searching for its best compressed representation $\mathbf{b} \in \mathcal{B}$, coupled with the decompressed signal

$\mathbf{v} \in \mathcal{S}$. The optimization trades off two opposing aspects of the representation: bit-cost and reconstruction quality. The bit-cost of the binary representation $\mathbf{b} \in \mathcal{B}$ is its length. Since, by (20), each $\mathbf{b} \in \mathcal{B}$ corresponds to one decompressed signal $\mathbf{v} \in \mathcal{S}$, we define the bit-cost of a decompressed signal $\mathbf{v} \in \mathcal{S}$ as the length of its binary representation $\mathbf{b} = F^{-1}(\mathbf{v})$. We also define the function $R_{\mathcal{S}}(\mathbf{v})$ to evaluate the bit-cost of the compressed binary representation associated with $\mathbf{v}$. Specifically, for $\mathbf{v} \in \mathcal{S}$ that satisfies $\mathbf{v} = F\left(\mathbf{b}\right)$, the bit-cost is

$$R_{\mathcal{S}}(\mathbf{v}) \triangleq \text{length}\left\{\mathbf{b}\right\}, \tag{22}$$

where length $\{\cdot\}$ counts the length of a binary description. The second part of the trade-off is the reconstruction distortion, $D(\mathbf{x}, \mathbf{v})$, evaluating the distance between the compression input $\mathbf{x}$ and its decompressed form $\mathbf{v}$. Note that the distortion value is real and nonnegative. Then, the optimization task including bit-cost constraints, corresponding to storage space or transmission bandwidth limitations, is

$$\begin{aligned} \hat{\mathbf{v}} = &\underset{\mathbf{v}\in\mathcal{S}}{\text{argmin}} \;\; D(\mathbf{x}, \mathbf{v}) \\ &\text{subject to} \quad R_{\mathcal{S}}(\mathbf{v}) \leq r \end{aligned} \tag{23}$$

where $r \geq 0$ is the maximal representation bit-cost allowed. Another relevant optimization problem, mirroring (23), is defined to minimize the compression bit-cost under a limited distortion amount, i.e.,

$$\begin{aligned} \hat{\mathbf{v}} = &\underset{\mathbf{v}\in\mathcal{S}}{\text{argmin}} \;\; R_{\mathcal{S}}(\mathbf{v}) \\ &\text{subject to} \quad D(\mathbf{x}, \mathbf{v}) \leq d \end{aligned} \tag{24}$$

where $d \geq 0$ is the tolerated distortion level. Without loss of generality, we consider here the optimization form in (24). The constrained optimization (24) is usually cast (see, e.g., Shoham and Gersho 1988, Chou et al. 1989, Ortega and Ramchandran 1998, Sullivan and Wiegand 1998, Shukla et al. 2005, and Sullivan et al. 2012) to its unconstrained Lagrangian form

$$\hat{\mathbf{v}} = \underset{\mathbf{v}\in\mathcal{S}}{\text{argmin}} \;\; R_{\mathcal{S}}(\mathbf{v}) + \lambda D(\mathbf{x}, \mathbf{v}) \tag{25}$$

where $\lambda \geq 0$ is a Lagrange multiplier corresponding to a distortion constraint $d_\lambda \geq 0$. Such compression without a prespecified distortion level is common, e.g., in video coding (Sullivan et al. 2012).

When working with high-dimensional signals (large N values), the discrete set $\mathcal{S}$ tends to be huge. Then, for arbitrarily structured distortion metrics $D(\mathbf{x}, \mathbf{v})$, one cannot directly solve the Lagrangian form in (25) via iterating over the elements in $\mathcal{S}$ and evaluating their corresponding costs (recall that (25) is a discrete optimization problem). Accordingly, compression methods are designed such that

the combination of $D(\mathbf{x}, \mathbf{v})$, $\mathcal{S}$, and $\mathcal{B}$ leads to a computationally tractable task. This is often obtained using architectures where nonoverlapping signal segments are independently compressed with respect to the squared-error distortion measure (see details in the Appendix). However, while such computationally efficient architectures that rely on squared-error metrics are prevalent (we also refer them as *standard* compression techniques), they are often too simple and limit the compression performance one could wish for in various settings of interest. This will be further demonstrated in section "Modular Strategies for Intricate Compression Problems".

Restoration by Compression

Regularization of inverse problems based on complexity measures is a well-established approach for estimation tasks (see, e.g., Rissanen 2000). In a subclass of these methods, complexity is defined based on the number of bits required for the compressed representation of the candidate estimate. This motivated various studies of signal and image denoising using lossy compression techniques (see, for example, Natarajan 1995 and Liu and Moulin 2001). The extension of this idea to image restoration problems beyond Gaussian denoising was studied from a theoretical perspective by Moulin and Liu (2000), also including a limited experimental demonstration for Poisson denoising based on a particularly designed compression process. Implementing the compression-based approach for other image restoration problems (such as deblurring, inpainting, super resolution, etc.) was considered as impractical for a long while until the *Restoration by Compression* architecture (Dar et al. 2016a, 2018c) resolved the computational difficulties via ADMM-based modularity. Next, we overview the main construction and applicative aspects of the *Restoration by Compression* idea as a special case of the generic modular optimization designs presented above.

The core idea in the *Restoration by Compression* approach (Dar et al. 2016a, 2018c) is to exploit existing compression techniques such that their underlying signal models will be indirectly used for desired restoration purposes. For this, we define the function R in (1) as a complexity regularizer, measuring the likelihood of a signal based on its compression bit-cost (assuming that more probable signals receive shorter compressed representations). Specifically, the regularizer extends the bit-cost evaluation function (22) such that for any $\mathbf{z} \in \mathbb{R}^N$ it returns

$$R(\mathbf{z}) = \begin{cases} R_{\mathcal{S}}(\mathbf{z}) & \text{for } \mathbf{z} \in \mathcal{S} \\ \infty & \text{for } \mathbf{z} \notin \mathcal{S} \end{cases}, \tag{26}$$

where $\mathcal{S}$ and $R_{\mathcal{S}}$ are conceptually associated with an existing compression technique. Then, considering the complexity regularizer (26), the optimization for the $\hat{\mathbf{z}}^{(t)}$ in (12) becomes equivalent to an operational rate-distortion optimization (25) with respect to the implicit architecture of the ready-to-use compression

method. This motivates the replacement of (12) with an application of a black-box compression module, followed by its respective decompression process, i.e.,

$$\mathbf{b}^{(t)} = StandardCompression\left(\tilde{\mathbf{v}}^{(t)};\theta\right) \tag{27}$$

$$\hat{\mathbf{z}}^{(t)} = StandardDecompression\left(\mathbf{b}^{(t)}\right). \tag{28}$$

Note that the application of both compression and decompression is in accordance with the optimization form in (25) that looks for the optimal decompressed signal corresponding to the given signal to compress. Interestingly, the utilized compression modules do not have to rely on rate-distortion optimizations (25), as in the case of transform coding architectures (such as the JPEG2000 method included in the following demonstrations). The Restoration by Compression procedure is summarized in Algorithm 4. See Dar et al. (2018c) for further detail on the parameters given to the compression modules in the iterative process. Moreover, the main concepts of the proposed algorithm are explained by Dar et al. (2018c) using rate-distortion theory for cyclo-stationary Gaussian signals.

Algorithm 4 Restoration by Compression: Basic Complexity Regularization

Inputs: $\mathbf{x}$, θ, $\tilde{\beta}$.
Initialize $t = 0$, $\hat{\mathbf{z}}^{(0)} = \mathbf{x}$, $\mathbf{u}^{(1)} = \mathbf{0}$.
repeat
  $t \leftarrow t + 1$
  $\tilde{\mathbf{z}}^{(t)} = \hat{\mathbf{z}}^{(t-1)} - \mathbf{u}^{(t)}$
  $\hat{\mathbf{v}}^{(t)} = \underset{\mathbf{v}\in\mathbb{R}^N}{\operatorname{argmin}} \left\|\mathbf{x} - \mathbf{H}\mathbf{v}\right\|_2^2 + \tilde{\beta}\left\|\mathbf{v} - \tilde{\mathbf{z}}^{(t)}\right\|_2^2$
  $\tilde{\mathbf{v}}^{(t)} = \hat{\mathbf{v}}^{(t)} + \mathbf{u}^{(t)}$
  $\mathbf{b}^{(t)} = StandardCompression\left(\tilde{\mathbf{v}}^{(t)};\theta\right)$
  $\hat{\mathbf{z}}^{(t)} = StandardDecompression\left(\mathbf{b}^{(t)}\right)$
  $\mathbf{u}^{(t+1)} = \mathbf{u}^{(t)} + \left(\hat{\mathbf{v}}^{(t)} - \hat{\mathbf{z}}^{(t)}\right)$
until stopping criterion is satisfied
Output: $\hat{\mathbf{v}}^{(t)}$.

Clearly, the artifacts introduced by the compression module participating in restoration process affect the produced estimate. Many compression artifacts originate in the common approach of independently coding nonoverlapping segments of the image. This block-based design also influences the complexity measure defining the regularizer in (26), essentially equivalent to summing the compression bit-costs of all the nonoverlapping blocks. This aspect was identified by Dar et al. (2018c) as introducing shift sensitivity into the regularizer (26). Accordingly, a shift-invariant complexity regularizer was proposed by Dar et al. (2018c), measuring the total bit-cost of *all the shifted versions* of the estimate evaluated. The shift operator $shift_j\{\cdot\}$ can be defined as a two-dimensional cyclical shift on an image or, alternatively, as

returning the rectangular portion of the image that starts at a shifted coordinate from the upper-left corner pixel of the full image (see Dar et al. 2018c for details). For each $j \in \{1, \ldots, N_b\}$, the two-dimensional offset applied by $shift_j\{\cdot\}$ is different. This leads to an extended Restoration by Compression process, described in Algorithm 5, including ADMM-based decoupling of the compressions of various shifts of the processed signals. Further details on the shift-invariant regularizer and the algorithm development are provided in Dar et al. (2018c). The applications of Algorithm 5 for deblurring and inpainting of images are presented in Figs. 3 and 4, respectively. The compression modules employed are JPEG2000 and HEVC (in its BPG implementation for image coding (Bellard)). Since HEVC provides significantly better compression performance than JPEG2000, a corresponding gap in their restoration abilities is also evident.

Algorithm 5 Restoration by Compression: Shift-Invariant Complexity Regularization

Inputs: $\mathbf{y}$, θ, $\tilde{\beta}$, and the number of shifts N_b.
Initialize $\left\{\hat{\mathbf{z}}^{j,(0)}\right\}_{j=1}^{N_b}$ (depending on the deterioration type).
$t = 1$ and $\mathbf{u}^{j,(1)} = \mathbf{0}$ for $j = 1, \ldots, N_b$.
repeat
  $\tilde{\mathbf{z}}^{j,(t)} = \hat{\mathbf{z}}^{j,(t-1)} - \mathbf{u}^{j,(t)}$ for $j = 1, \ldots, N_b$
  Solve the ℓ_2-constrained deconvolution:

$$\hat{\mathbf{v}}^{(t)} = \underset{\mathbf{v}\in\mathbb{R}^N}{\operatorname{argmin}} \left\|\mathbf{x} - \mathbf{H}\mathbf{v}\right\|_2^2 + \tilde{\beta} \sum_{j=1}^{N_b} \left\|\mathbf{v} - \tilde{\mathbf{z}}^{j,(t)}\right\|_2^2$$

  for $j = 1, \ldots, N_b$ **do**
    $\tilde{\mathbf{v}}_{shifted}^{j,(t)} = shift_j\left\{\hat{\mathbf{v}}^{(t)} + \mathbf{u}^{j,(t)}\right\}$
    $\mathbf{b}^{j,(t)} = StandardCompression\left(\tilde{\mathbf{v}}_{shifted}^{j,(t)}; \theta\right)$
    $\hat{\mathbf{z}}_{shifted}^{j,(t)} = StandardDecompression\left(\mathbf{b}^{j,(t)}\right)$
    $\hat{\mathbf{z}}^{j,(t)} = shift_j^{-1}\left\{\hat{\mathbf{z}}_{shifted}^{j,(t)}\right\}$
    $\mathbf{u}^{j,(t+1)} = \mathbf{u}^{j,(t)} + \left(\hat{\mathbf{v}}^{(t)} - \hat{\mathbf{z}}^{j,(t)}\right)$
  end for
  $t \leftarrow t + 1$
until stopping criterion is satisfied
Output: $\hat{\mathbf{v}}^{(t)}$.

Modular Strategies for Intricate Compression Problems

The utilization of available compression methods in modular restoration processes (Algorithms 4 and 5) naturally raises the question whether modular optimization strategies are relevant also to intricate compression problems. This is indeed the case, as established by the *System-Aware Compression* framework (Dar et al. 2018a,b,d), where ADMM-based modular strategies are employed for

Fig. 3 The deblurring experiment (settings #2 in Dar et al. 2018c) for the Cameraman image (256 × 256 pixels). (**a**) The underlying image. (**b**) Degraded image (20.76 dB). (**c**) Restored image using Algorithm 5 with JPEG2000 compression (28.10 dB). (**d**) Restored image using Algorithm 5 with HEVC compression (30.14 dB)

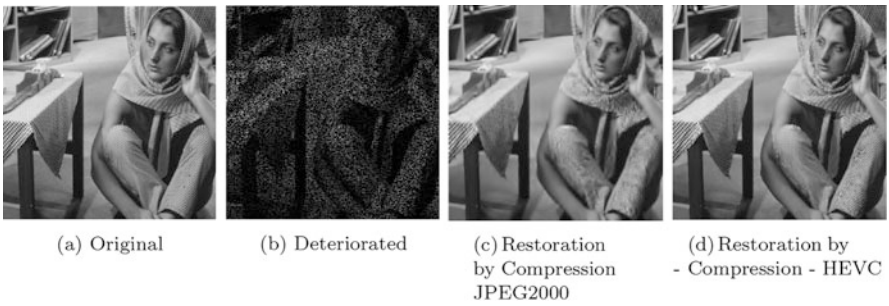

Fig. 4 The inpainting experiment (80% missing pixels) for the Barbara image (512 × 512 pixels). (**a**) The original image. (**b**) Deteriorated image. (**c**) Restored image using Algorithm 5 with JPEG2000 compression (24.83 dB). (**d**) Restored image using Algorithm 5 with HEVC compression (28.80 dB)

optimizing end-to-end performance of systems involving acquisition, compression, and rendering stages. The main idea is to decouple unusual distortion metrics from the actual compression tasks that, in turn, can be applied using black-box compression modules (which are operated with respect to the elementary squared-error metric). Hence, this methodology opens a new research path for addressing complex compression problems including, for example, optimizations for nonlocal processing/prediction architectures, enhancement filters or degradation processes, and perceptual metrics assessing subjective quality of audio/visual signals. Indeed, a successful implementation of this approach for perceptually oriented image compression (using an alternating minimization procedure) was proposed by Rott Shaham and Michaeli (2018).

In this section, we overview the *System-Aware Compression* concept (Dar et al. 2018a,b,d), demonstrating the main aspects of using modular optimizations for intricate compression problems. The motivation for the *System-Aware Compression* framework stems from a structure common to many imaging systems (see

Fig. 5), where a source signal is first acquired, then compressed for its storage or transmission, and eventually decompressed and rendered back into a signal that can be displayed or further processed. Obviously, in such systems, the quality of the eventual output depends on the entire acquisition-rendering chain and not solely on the lossy compression component. Yet, the employed compression technique is often system independent, hence inducing suboptimal rate-distortion performance for the entire system. The *System-Aware Compression* architecture is a practical and modular way for optimizing the end-to-end performance (in its rate-distortion trade-off sense) of such acquisition-rendering systems.

Let us describe the system structure considered for the mathematical development of the method (Fig. 6). A source signal, an N-length column vector $\mathbf{x} \in \mathbb{R}^N$,

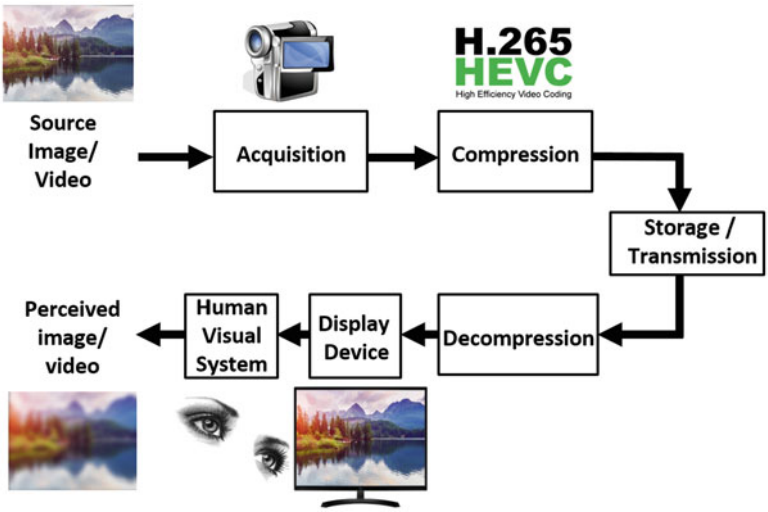

Fig. 5 The general imaging system structure motivating the System-Aware Compression approach

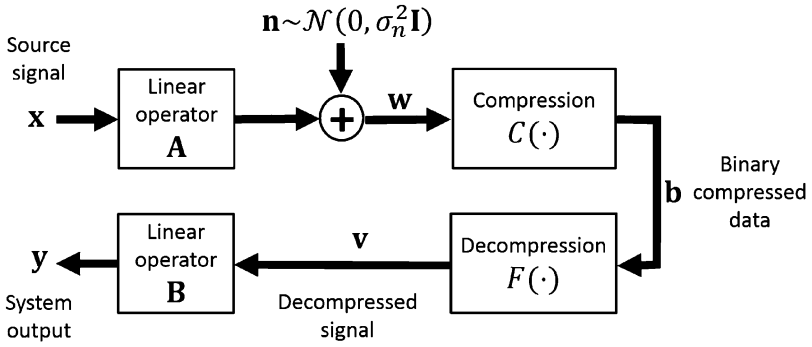

Fig. 6 The system model addressed by the System-Aware Compression framework

undergoes a linear processing represented by the $M \times N$ matrix $\mathbf{A}$ and, then, deteriorated by an additive white Gaussian noise vector $\mathbf{n} \sim \mathcal{N}\left(0, \sigma_n^2 \mathbf{I}\right)$, resulting in the signal

$$\mathbf{w} = \mathbf{A}\mathbf{x} + \mathbf{n} \tag{29}$$

where $\mathbf{w}$ and $\mathbf{n}$ are M-length column vectors. We represent the lossy compression procedure via the mapping $C : \mathbb{R}^M \rightarrow \mathcal{B}$ from the M-dimensional signal domain to a discrete set $\mathcal{B}$ of binary compressed representations (which may have different lengths). The signal $\mathbf{w}$ is the input to the compression component of the system, producing the compressed binary data $\mathbf{b} = C(\mathbf{w})$ 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 $\mathbf{b} \in \mathcal{B}$ is decompressed to provide the signal $\mathbf{v} = F\left(\mathbf{b}\right)$ where $F : \mathcal{B} \rightarrow \mathcal{S}$ represents the decompression mapping between the binary compressed representations in $\mathcal{B}$ to the corresponding decompressed signals in the discrete set $\mathcal{S} \subset \mathbb{R}^M$. The decompressed signal $\mathbf{v}$ is further processed by the linear operator denoted as the $N \times M$ matrix $\mathbf{B}$, resulting in the system output signal

$$\mathbf{y} = \mathbf{B}\mathbf{v}, \tag{30}$$

which is an N-length real-valued column vector.

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

We assume here that the operators $\mathbf{A}$ and $\mathbf{B}$, as well as the noise variance σ_n^2, 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 $\mathbf{y}$ to be the best approximation of the source signal $\mathbf{x}$ under the bit-budget constraint. However, at the compression stage, we do not accurately know $\mathbf{x}$ but rather its degraded form $\mathbf{w}$ formulated in (29). This motivates us to suggest the following distortion metric with respect to the system output $\mathbf{y}$

$$D_s\left(\mathbf{w}, \mathbf{y}\right) = \frac{1}{M}\left\|\mathbf{w} - \mathbf{A}\mathbf{y}\right\|_2^2. \tag{31}$$

This metric conforms with the fact that if $\mathbf{y}$ is close to $\mathbf{x}$, then, by (29), $\mathbf{w}$ will be close to $\mathbf{A}\mathbf{y}$ up to the noise $\mathbf{n}$. Indeed, for the ideal case of $\mathbf{y} = \mathbf{x}$, the metric (31) becomes

$$D_s\left(\mathbf{w}, \mathbf{x}\right) = \frac{1}{M}\left\|\mathbf{n}\right\|_2^2 \approx \sigma_n^2 \tag{32}$$

where the last approximate equality is under the assumption of a sufficiently large M (the length of $\mathbf{n}$). Since $\mathbf{y} = \mathbf{B}\mathbf{v}$, we can rewrite the distortion $D_s\left(\mathbf{w}, \mathbf{y}\right)$ in (31) as a function of the decompressed signal $\mathbf{v}$, namely,

$$D_c(\mathbf{w}, \mathbf{v}) = \frac{1}{M} \left\| \mathbf{w} - \mathbf{AB}\mathbf{v} \right\|_2^2. \tag{33}$$

Since the operator $\mathbf{B}$ produces the output signal $\mathbf{y}$, an ideal result will be $\mathbf{y} = \mathbf{P}_B\mathbf{x}$, where $\mathbf{P}_B$ is the matrix projecting onto $\mathbf{B}$'s range. The corresponding ideal distortion is

$$d_0 \triangleq D_s\left(\mathbf{w}, \mathbf{P}_B\mathbf{x}\right) = \frac{1}{M} \left\| \mathbf{A}\left(\mathbf{I} - \mathbf{P}_B\right)\mathbf{x} + \mathbf{n} \right\|_2^2. \tag{34}$$

We use the distortion metric (33) to constrain the bit-cost minimization in the following rate-distortion optimization

$$\begin{aligned} \hat{\mathbf{v}} = & \underset{\mathbf{v}\in\mathcal{S}}{\operatorname{argmin}} \quad R(\mathbf{v}) \\ & \text{subject to} \quad d_0 \le \frac{1}{M} \left\| \mathbf{w} - \mathbf{AB}\mathbf{v} \right\|_2^2 \le d_0 + d \end{aligned} \tag{35}$$

where $R(\mathbf{v})$ evaluates the length of the binary compressed description of the decompressed signal $\mathbf{v}$ and $d \ge 0$ determines the allowed distortion. By (34), the value d_0 depends on the operator $\mathbf{A}$, the null space of $\mathbf{B}$, the source signal $\mathbf{x}$, and the noise realization $\mathbf{n}$. Since $\mathbf{x}$ and $\mathbf{n}$ are unknown, d_0 cannot be accurately calculated in the operational case (in Dar et al. (2018d) we formulate the expected value of d_0 for the case of a cyclo-stationary Gaussian source signal). We address the optimization (35) using its unconstrained Lagrangian form

$$\hat{\mathbf{v}} = \underset{\mathbf{v}\in\mathcal{S}}{\operatorname{argmin}} \; R(\mathbf{v}) + \lambda \frac{1}{M} \left\| \mathbf{w} - \mathbf{AB}\mathbf{v} \right\|_2^2 \tag{36}$$

where $\lambda \ge 0$ is a Lagrange multiplier corresponding to some distortion constraint $d_\lambda \ge d_0$ (such optimization strategy with respect to some Lagrange multiplier is common, e.g., in video coding (Sullivan et al. 2012)). In the case of high-dimensional signals, the discrete set $\mathcal{S}$ is extremely large, and, therefore, it is impractical to directly solve the Lagrangian form in (36) for generally structured matrices $\mathbf{A}$ and $\mathbf{B}$. This difficulty vanishes, for example, when $\mathbf{A} = \mathbf{B} = \mathbf{I}$, reducing the Lagrangian optimization in (36) to the standard (system independent) compression form (see, e.g., Shoham and Gersho 1988 and Ortega and Ramchandran 1998).

The optimization (36) matches the generic template presented in section "Modular ADMM-Based Optimization: General Construction and Guidelines", and, therefore, we can formulate an ADMM-based modular procedure to address it. This modular optimization process is a special case of the generic procedure described in Algorithm 1, taking here the form of Algorithm 6. Note that we use the form of Algorithm 1 where the eventual output is the output of the module applied in the last iteration, which in our case corresponds to the output of the compression module in the last iteration (and this is the desired output because in this section we consider compression application, unlike the Restoration by

Compression method in Algorithm 4 that its purpose is restoration by means of compression-based regularization). The interested reader is referred to Dar et al. (2018d) for a rate-distortion theoretic analysis for cyclo-stationary Gaussian signals and linear shift-invariant operators, explaining various aspects of the proposed procedure.

Algorithm 6 System-Aware Compression

Inputs: $\mathbf{w}$, θ, $\tilde{\beta}$.
Initialize $t = 0$, $\hat{\mathbf{z}}^{(0)} = \mathbf{w}$, $\mathbf{u}^{(1)} = \mathbf{0}$.
repeat
  $t \leftarrow t + 1$
  $\tilde{\mathbf{z}}^{(t)} = \hat{\mathbf{z}}^{(t-1)} - \mathbf{u}^{(t)}$
  $\mathbf{b}^{(t)} = StandardCompression\left(\tilde{\mathbf{z}}^{(t)}, \theta\right)$
  $\hat{\mathbf{v}}^{(t)} = StandardDecompression\left(\mathbf{b}^{(t)}\right)$
  $\tilde{\mathbf{v}}^{(t)} = \hat{\mathbf{v}}^{(t)} + \mathbf{u}^{(t)}$
  $\hat{\mathbf{z}}^{(t)} = \underset{\mathbf{z}\in\mathbb{R}^N}{\operatorname{argmin}} \left\|\mathbf{w} - \mathbf{ABz}\right\|_2^2 + \tilde{\beta}\left\|\mathbf{z} - \tilde{\mathbf{v}}^{(t)}\right\|_2^2$
  $\mathbf{u}^{(t+1)} = \mathbf{u}^{(t)} + \left(\hat{\mathbf{v}}^{(t)} - \hat{\mathbf{z}}^{(t)}\right)$
until stopping criterion is satisfied
Output: $\mathbf{b}^{(t)}$, which is the binary compressed data obtained in the last iteration.

To demonstrate the essence of the System-Aware Compression approach, we provide here a representative example taken from Dar et al. (2018b), excluding the acquisition stage (i.e., $\mathbf{A} = \mathbf{I}$ and $\sigma_n^2 = 0$) while considering a post-decompression operator $\mathbf{B}$ implementing a shift-invariant Gaussian blur degradation. One can perceive this setting as optimizing image compression with respect to degradation occurring later (after decompression) at the display device, where no additional processing is done after decompression in order to counterbalance the degradation. To observe the gains achieved by the modular optimization approach, let us first examine the unoptimized (regular) compression process where the input image (Fig. 7a) is compressed using the state-of-the-art HEVC standard at a bit-rate of 3.75 bits per pixel (bpp), yielding the decompressed image in Fig. 7b (this is the image before blur degradation). Then, the decompressed image after degradation (Fig. 7c) is very blurry, as also reflected in the respective PSNR value (measured with respect to the image before compression). In the modular optimization approach, the input image is processed such that the compression in the last iteration is applied on a sharpened version (see Fig. 7d) adjusted to the known blur operator; then, the compressed image at bit-rate 2.21 bpp eventually results in a degraded decompression with moderate blur effects (Fig. 7f) and a PSNR gain of 7.52 dB with respect to the regular compression (which used even a higher bit-rate). See Dar et al. (2018b) for extensive experimental evaluation including PSNR-bitrate performance curves and comparison to additional alternatives. Furthermore, LCD display degradations associated with motion blur are also examined by Dar et al. (2018b).

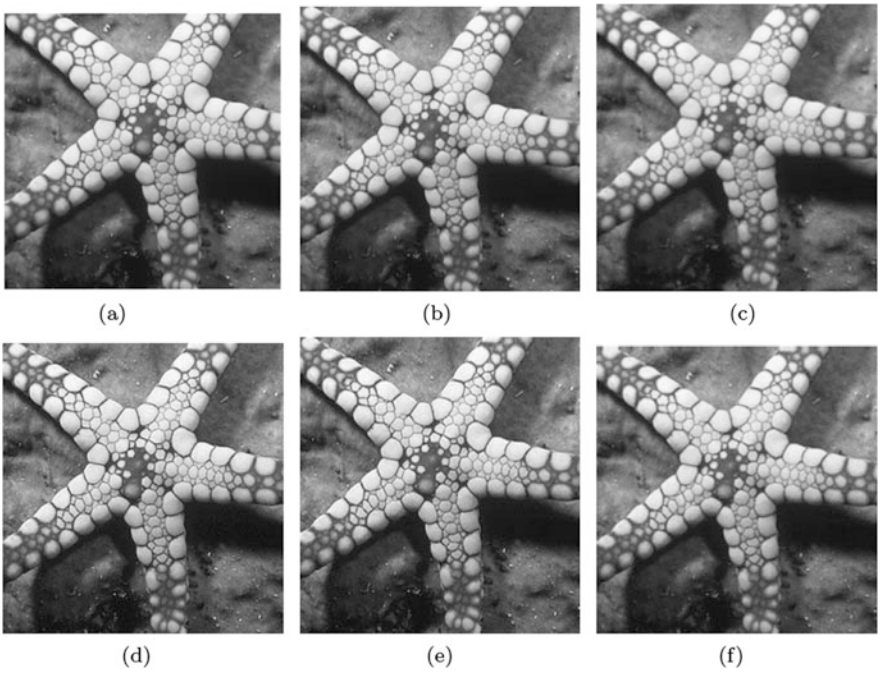

Fig. 7 Comparison of the System-Aware Compression approach and regular compression. The settings consider a Gaussian blur operator degrading the decompressed image. The intermediate and eventual images of the regular and the modular optimization process are presented. (**a**) Input. (**b**) Regular Decompression (3.75 bpp). (**c**) Regular Degraded Decompression (34.32 dB). (**d**) System-Aware Compression: Input to Last Iteration Compression. (**e**) System-Aware Compression: Decompression (2.21 bpp). (**f**) System-Aware Compression: Degraded Decompression (41.84 dB)

Additional evaluations of the System-Aware Compression approach are provided by Dar et al. (2018d) for video compression settings including acquisition degradation of low-pass filtering and subsampling and post-decompression nearest-neighbor upsampling. In Dar et al. (2018a), the idea is demonstrated for a simplified model of multimedia distribution networks, where a set of possible degradation operators and their probabilities are considered by the optimized compression process.

Distributed Representations Using Black-Box Modules

All the above problems conduct optimizations for finding one signal (or compressed representation) that minimizes a Lagrangian cost of interest. As shown, these tasks are addressed very well by modular optimizations, relying on sequential black-box module applications. In this section, we demonstrate that the modular optimization

approach is useful also to problems seeking for a set of signals (or representations) that collaboratively minimize a joint Lagrangian cost.

The General Framework

The following extends the settings and developments given in section "Unconstrained Lagrangian Optimizations via ADMM". The general optimization form for distributed representations broadens the single-representation problem in (1) to an unconstrained Lagrangian form optimizing several signals, i.e.,

$$\left(\hat{\mathbf{v}}_1, \ldots, \hat{\mathbf{v}}_K\right) = \underset{\mathbf{v}_1,\ldots,\mathbf{v}_K \in \mathcal{M}}{\operatorname{argmin}} \sum_{i=1}^{K} R\left(\mathbf{v}_i\right) + \lambda D\left(\mathbf{x}; \mathbf{v}_1, \ldots, \mathbf{v}_K\right) \tag{37}$$

to be solved for the K representations $\mathbf{v}_1, \ldots, \mathbf{v}_K \in \mathcal{M} \subset \mathbb{R}^N$, where $\mathcal{M}$ is a (continuous or discrete) subset of the N-dimensional real space. While there are several optimization variables, they all intend to (possibly differently) represent the single given signal $\mathbf{x} \in \mathbb{R}^M$. The general scalar-valued function $R : \mathcal{M} \to \mathbb{R}$ is defined for individual representation as inputs, and D is a scalar-valued function receiving $\mathbf{x}$ and all the representations together as inputs.

The computational challenge of solving (37) is clear, as it is hard even in the case of optimizing one signal (as discussed in section "Modular Optimizations Based on Standard Compression Techniques"). Nevertheless, we can utilize variable splitting and ADMM techniques to develop a sequential optimization process addressing (37). Essentially, this is an extension of the ADMM constructions presented in section "Unconstrained Lagrangian Optimizations via ADMM". Here the developments originate in the variable splitting applied on (37) via

$$\begin{aligned} &\left(\left\{\hat{\mathbf{v}}_i\right\}_{i=1}^{K}, \left\{\hat{\mathbf{z}}_i\right\}_{i=1}^{K}\right) = \underset{\substack{\{\mathbf{v}_i\}_{i=1}^{K} \in \mathcal{M} \\ \{\mathbf{z}_i\}_{i=1}^{K} \in \mathbb{R}^N}}{\operatorname{argmin}} \sum_{i=1}^{K} R\left(\mathbf{v}_i\right) + \lambda D\left(\mathbf{x}; \mathbf{z}_1, \ldots, \mathbf{z}_K\right) \\ &\text{subject to} \quad \mathbf{v}_i = \mathbf{z}_i \quad \text{for } i = 1, \ldots, K \end{aligned} \tag{38}$$

where $\mathbf{z}_1, \ldots, \mathbf{z}_K \in \mathbb{R}^N$ are auxiliary variables that are not directly constrained to the eventual output domain $\mathcal{M}$ (similar to the developments in section "Unconstrained Lagrangian Optimizations via ADMM").

Then, the scaled form of the augmented Lagrangian and the method of multipliers (Boyd et al. 2011, Ch. 2) renders (38) into the sequential process

$$\left(\left\{\hat{\mathbf{v}}_i^{(t)}\right\}_{i=1}^{K}, \left\{\hat{\mathbf{z}}_i^{(t)}\right\}_{i=1}^{K}\right) = \tag{39}$$

$$\underset{\substack{\{\mathbf{v}_i\}_{i=1}^{K}\in\mathcal{M}\\ \{\mathbf{z}_i\}_{i=1}^{K}\in\mathbb{R}^N}}{\operatorname{argmin}} \sum_{i=1}^{K} R\left(\mathbf{v}_i\right) + \lambda D\left(\mathbf{x}; \mathbf{z}_1, \ldots, \mathbf{z}_K\right) + \frac{\beta}{2}\sum_{i=1}^{K}\left\|\mathbf{v}_i - \mathbf{z}_i + \mathbf{u}_i^{(t)}\right\|_2^2$$

$$\mathbf{u}_i^{(t+1)} = \mathbf{u}_i^{(t)} + \left(\hat{\mathbf{v}}_i^{(t)} - \hat{\mathbf{z}}_i^{(t)}\right) \quad \text{for } i = 1, \ldots, K \tag{40}$$

where t denotes the iteration index, $\left\{\mathbf{u}_i^{(t)}\right\}_{i=1}^{K} \in \mathbb{R}^N$ are the scaled dual variables, and β is an auxiliary parameter originating at the augmented Lagrangian (note that β is an intentionally joined parameter for the purpose of easing the parameter tuning process). The corresponding ADMM process is obtained by applying one iteration of alternating minimization on (39), leading to

$$\hat{\mathbf{v}}_i^{(t)} = \underset{\mathbf{v}_i\in\mathcal{M}}{\operatorname{argmin}}\, R\left(\mathbf{v}_i\right) + \frac{\beta}{2}\left\|\mathbf{v}_i - \tilde{\mathbf{z}}_i^{(t)}\right\|_2^2 \quad \text{for } i = 1, \ldots, K \tag{41}$$

$$\hat{\mathbf{z}}_i^{(t)} = \underset{\mathbf{z}_i\in\mathbb{R}^N}{\operatorname{argmin}}\, \lambda D\left(\mathbf{x}; \left\{\hat{\mathbf{z}}_i^{(t)}\right\}_{j=1}^{i-1}, \mathbf{z}_i, \left\{\hat{\mathbf{z}}_i^{(t-1)}\right\}_{j=i+1}^{K}\right) + \frac{\beta}{2}\left\|\mathbf{z}_i - \tilde{\mathbf{v}}_i^{(t)}\right\|_2^2$$

$$\text{for } i = 1, \ldots, K \tag{42}$$

$$\mathbf{u}_i^{(t+1)} = \mathbf{u}_i^{(t)} + \left(\hat{\mathbf{v}}_i^{(t)} - \hat{\mathbf{z}}_i^{(t)}\right) \quad \text{for } i = 1, \ldots, K \tag{43}$$

where $\tilde{\mathbf{z}}_i^{(t)} \triangleq \hat{\mathbf{z}}_i^{(t-1)} - \mathbf{u}_i^{(t)}$ and $\tilde{\mathbf{v}}_i^{(t)} \triangleq \hat{\mathbf{v}}_i^{(t)} + \mathbf{u}_i^{(t)}$. Nicely, the obtained process does not only decouple the treatment of $\left\{\mathcal{M}, R\right\}$ from D as before (see section "Unconstrained Lagrangian Optimizations via ADMM") but also separates the treatment of the various representations. Thus, (41), (42), and (43) simplify the challenging structure in (37). Moreover, the subproblems in (41) have the same form associated with black-box modules applied on individual signals (see section "Employing Black-Box Modules"). This casting leads us to the process summarized in Algorithm 7. Also note that in each iteration t the treatment of the K representations is sequential (this reordered procedure is equivalent to the form in (41), (42), and (43)).

Modular Optimizations for Holographic Compression of Images

We now turn to exemplify the generic approach in Algorithm 7 for the purpose of optimizing distributed representations in compressed, standard-compatible, forms. Our recent framework for holographic compression (Dar and Bruckstein 2021) represents a given signal using a set of distinct compressed descriptions, that any subset of them enables reconstruction of the signal at a quality determined solely

Algorithm 7 General Modular Optimization of Multiple Representations

Inputs: $\mathbf{x}$, θ, $\tilde{\beta}$.
Initialize $t = 0$, $\hat{\mathbf{z}}^{(0)} = \mathbf{x}$, $\mathbf{u}^{(1)} = \mathbf{0}$.
Initialize $t = 0$.
Initialize (for $i = 1, \ldots, K$) $\mathbf{u}_i^{(1)} = \mathbf{0}$ and $\hat{\mathbf{z}}_i^{(0)}$ according to the specific application.
repeat
  $t \leftarrow t + 1$
  for $i = 1, \ldots, K$ **do**
    $\tilde{\mathbf{z}}_i^{(t)} = \hat{\mathbf{z}}_i^{(t-1)} - \mathbf{u}_i^{(t)}$
    $\hat{\mathbf{v}}_i^{(t)} = BlackBoxModule\left(\tilde{\mathbf{z}}_i^{(t)}; \theta\right)$
    $\tilde{\mathbf{v}}_i^{(t)} = \hat{\mathbf{v}}_i^{(t)} + \mathbf{u}_i^{(t)}$
    $\hat{\mathbf{z}}_i^{(t)} = \underset{\mathbf{z}_i \in \mathbb{R}^N}{\operatorname{argmin}} \lambda D\left(\mathbf{x}; \left\{\hat{\mathbf{z}}_i^{(t)}\right\}_{j=1}^{i-1}, \mathbf{z}_i, \left\{\hat{\mathbf{z}}_i^{(t-1)}\right\}_{j=i+1}^{K}\right) + \tilde{\beta}\left\|\mathbf{z}_i - \tilde{\mathbf{v}}_i^{(t)}\right\|_2^2$
    $\mathbf{u}_i^{(t+1)} = \mathbf{u}_i^{(t)} + \left(\hat{\mathbf{v}}_i^{(t)} - \hat{\mathbf{z}}_i^{(t)}\right)$
  end for
until stopping criterion is satisfied
Output: $\hat{\mathbf{v}}_1^{(t)}, \ldots, \hat{\mathbf{v}}_K^{(t)}$ and/or other application-specific outputs of $BlackBoxModule$.

by the number of compressed representations utilized. This property of holographic representations is useful for designing progressive refinement mechanisms independent of the order the representations are accessible (Bruckstein et al. 1998, 2000, 2018).

In Dar and Bruckstein (2021) we identified the shift sensitivity of standard compression techniques as a property useful for constructing holographic representations in binary compressed forms. Specifically, compressions of shifted versions of an image provide a set of distinct decompressed images of similar individual qualities, but combining subsets of them (by back-shifts and averaging) achieves remarkable quality gains (see details in Dar and Bruckstein 2021). While this architecture is new and interesting, it does not include optimization aspects. This led us to suggest an optimization procedure unleashing the potential benefits of the shift-based holographic compression settings. Here we can consider this optimization framework as a special case of the generic process described in Algorithm 7, described as follows. First, the general $\{\mathcal{M}, R\}$ notions are set to the respective components $\{\mathcal{S}, R_{\mathcal{S}}\}$ of a standard compression method (as defined in section "Preliminaries: Lossy Compression via Operational Rate-Distortion Optimization"). This makes the first component in (37) the accumulated bit-cost of all the compressed representations. In Dar and Bruckstein (2021) we set the function D to evaluate the average MSE of reconstructions formed using subsets of m out of the K representations, where $m \in \{2, \ldots, K\}$ and assuming $K > 1$. This improves the reconstruction quality for subsets of m representations, at the inevitable expense of reducing their individual qualities. Therefore, we also include in D a regularization term computing the average MSE of the single-representation reconstructions. We denote the sequence of integers from 1 to K as $[[K]] \triangleq \{1, \ldots, K\}$. For $m \in [[K]]$, an m-combination of the set $[[K]]$ is a subset of m distinct numbers from $[[K]]$. We

denote the set of all m-combinations of $[[K]]$ as $\binom{[[K]]}{m}$, where the latter contains $\binom{K}{m}$ elements. The corresponding formulation of D is

$$D\left(\mathbf{x};\mathbf{v}_1,\ldots,\mathbf{v}_K\right)=\frac{1}{\binom{K}{m}}\sum_{(i_1,\ldots,i_m)\in\binom{[[K]]}{m}}D^{(m)}\left(\mathbf{x};\mathbf{v}_{i_1},\ldots,\mathbf{v}_{i_m}\right)\\ +\eta\frac{1}{K}\sum_{i=1}^{K}D^{(1)}\left(\mathbf{x};\mathbf{v}_i\right) \tag{44}$$

The parameter η determines the regularization level of the individual representation quality. Moreover,

$$D^{(1)}\left(\mathbf{x};\mathbf{v}_i\right)\triangleq\frac{1}{N}\left\|\mathbf{x}-\mathbf{S}_i^T\mathbf{v}_i\right\|_2^2 \tag{45}$$

is the reconstruction MSE corresponding to the single representation $\mathbf{v}_i$, and

$$D^{(m)}\left(\mathbf{x};\mathbf{v}_{i_1},\ldots,\mathbf{v}_{i_m}\right)\triangleq\frac{1}{N}\left\|\mathbf{x}-\frac{1}{m}\sum_{j=1}^{m}\mathbf{S}_{i_j}^T\mathbf{v}_{i_j}\right\|_2^2 \tag{46}$$

is the MSE of reconstruction using the m representations $\mathbf{v}_{i_1},\ldots,\mathbf{v}_{i_m}$. The matrices $\mathbf{S}_i^T$ and $\mathbf{S}_{i_j}^T$ correspond to back shift operators matching the shift forms used to create the compressed representations (further details are available in Dar and Bruckstein 2021). Then, by the settings of $\mathcal{M}$, R, and D, Algorithm 7 is specified for optimizing shift-based holographic compressed representations – this process is described in Algorithm 8.

In Figs. 8 and 9, we provide representative results taken from Dar and Bruckstein (2021). First, Fig. 8 presents reconstructions obtained from JPEG2000-compatible holographic compressions optimized for using sets of four representations. Specifically note the similar quality obtained using the individual representations and how they collaboratively achieve progressive refinement. This behavior is also clearly demonstrated in Fig. 9 by the curves of PSNR versus number of representations (packets) used for reconstructions. In particular, Fig. 9 shows the curves obtained for all the subset combinations in each of the examined methods. This exhibits the ability of the proposed method for optimizing reconstructions that rely on a specified number of representations (independent of the actual participating signals).The interested reader is referred to Dar and Bruckstein (2021) for additional details and experimental demonstrations.

Algorithm 8 Modular Holographic Compression: Optimized for Reconstructions Using m Representations

Inputs: $\mathbf{x}$, $\tilde{\beta}$, λ, η, θ, m, K.
Initialize $t = 0$.
Initialize (for $i = 1, \ldots, K$) $\hat{\mathbf{z}}_i^{(0)} = \mathbf{S}_i\mathbf{x}$ and $\mathbf{u}_i^{(1)} = \mathbf{0}$.
repeat
  $t \leftarrow t + 1$
  for $i = 1, \ldots, K$ **do**
    $\tilde{\mathbf{z}}_i^{(t)} = \hat{\mathbf{z}}_i^{(t-1)} - \mathbf{u}_i^{(t)}$
    $\mathbf{b}_i^{(t)} = StandardCompression\left(\tilde{\mathbf{z}}_i^{(t)}, \theta\right)$
    $\hat{\mathbf{v}}_i^{(t)} = StandardDecompression\left(\mathbf{b}_i^{(t)}\right)$
    $\tilde{\mathbf{v}}_i^{(t)} = \hat{\mathbf{v}}_i^{(t)} + \mathbf{u}_i^{(t)}$
    $\hat{\mathbf{z}}_i^{(t)} = \frac{N\tilde{\beta}\tilde{\mathbf{v}}_i^{(t)} + \frac{\eta}{\lambda K}\mathbf{S}_i\mathbf{x} + \frac{\lambda}{m^2\cdot\binom{K}{m}}\mathbf{S}_i\mathbf{w}_i^{(m)}}{N\tilde{\beta} + \frac{\eta}{\lambda K} + \frac{\lambda}{m^2\cdot\binom{K}{m}}\cdot|\mathcal{I}_i^{(m)}|}$
    where
    $$\mathbf{w}_i^{(m)} \triangleq \sum_{(i_1,\ldots,i_m)\in\mathcal{I}_i^{(m)}} \left(m\mathbf{x} - \sum_{\substack{i_j<i \\ j\in\{1,\ldots,m\}}} \mathbf{S}_{i_j}^T\hat{\mathbf{z}}_{i_j}^{(t)} - \sum_{\substack{i_j>i \\ j\in\{1,\ldots,m\}}} \mathbf{S}_{i_j}^T\hat{\mathbf{z}}_{i_j}^{(t-1)} \right)$$
    and $\mathcal{I}_i^{(m)}$ contains all the m-combinations including the i^{th} representation.
    $\mathbf{u}_i^{[t+1]} = \mathbf{u}_i^{(t)} + \left(\hat{\mathbf{v}}_i^{(t)} - \hat{\mathbf{z}}_i^{(t)}\right)$
  end for
until stopping criterion is satisfied
Output: The binary compressed packets $\mathbf{b}_1^{(t)}, \ldots, \mathbf{b}_K^{(t)}$.

Conclusion

In this chapter, we presented the recent methodology of modular optimizations, employing black-box modules in procedures addressing various imaging problems. The main idea is that fundamental tasks, such as denoising and compression, have excellent ready-to-use techniques that can be utilized for solving more intricate problems. We presented the developments of ADMM-based algorithms for modular optimizations, starting in general settings exhibiting the essence and prominent guidelines of the approach. Then, we overviewed settings where denoising and compression techniques are operated as stages in sequential procedures for restoration and intricate compression problems. We also outlined the extension of modular optimizations to formation of distributed representations and particularly exemplified it for the case of holographic compression of images. The perspectives emphasized in this chapter should motivate new ideas and settings extending the current class of module types used and problems addressed via modular optimization strategies.

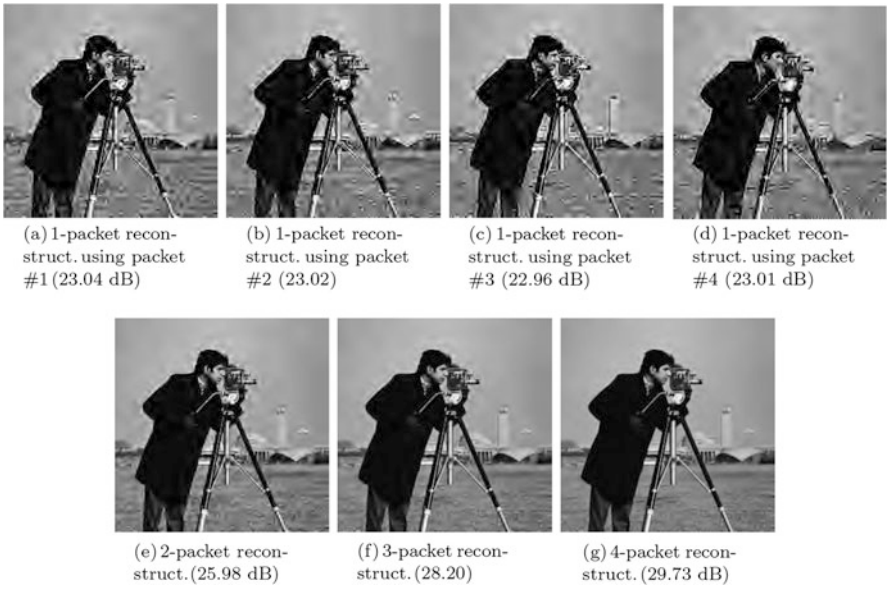

Fig. 8 Examples (taken from Dar and Bruckstein 2021) for reconstructions of the "Cameraman" image using several representations out of a set of four holographic compressed descriptions. The compression employed is JPEG2000 at a compression ratio of 1:50. (**a**)–(**d**) the 1-packet reconstructions using each of the individual packets. (**e**)–(**g**) examples for the m-packet reconstructions for $m = 2, 3, 4$

Appendix: Operational Rate-Distortion Optimizations in Block-Based Architectures

The computational challenge of operational rate-distortion optimizations (see section "Preliminaries: Lossy Compression via Operational Rate-Distortion Optimization") is often addressed via the squared-error metric

$$D(\mathbf{x}, \mathbf{v}) = \|\mathbf{x} - \mathbf{v}\|_2^2 , \tag{47}$$

leading to practical forms of the Lagrangian rate-distortion optimization (25). These useful structures also process the signal $\mathbf{x}$ based on its segmentation into a set of nonoverlapping blocks $\{\mathbf{x}_i\}_{i\in\mathcal{I}}$; here, each of them is a column vector of N_b samples, and $\mathcal{I}$ is the set of indices corresponding to the nonoverlapping blocks of the signal. Correspondingly, the decompressed signal $\mathbf{v}$ is decomposed into a set of nonoverlapping blocks $\{\mathbf{v}_i\}_{i\in\mathcal{I}}$. This lets us casting (47) into

$$D(\mathbf{x}, \mathbf{v}) = \sum_{i\in\mathcal{I}} \|\mathbf{x}_i - \mathbf{v}_i\|_2^2 , \tag{48}$$

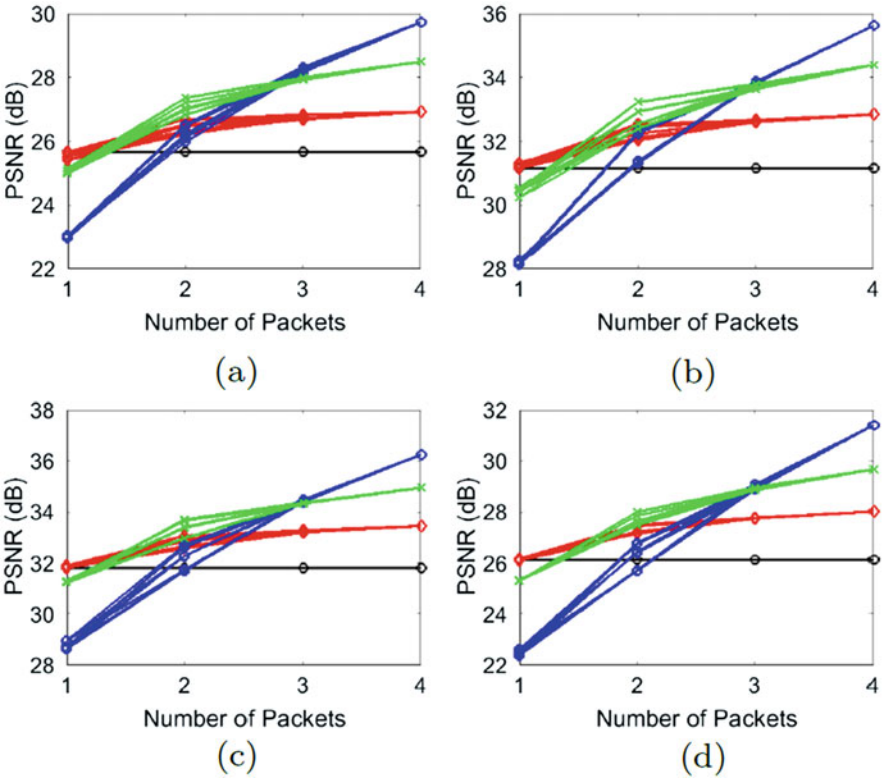

Fig. 9 PSNR versus the number of representations used for the reconstructions. The entire set contains four packets, each formed by JPEG2000 compression at 1:50 compression ratio. The black, red, green, and blue curves, respectively, represent the methods of exact duplications, baseline (unoptimized), optimized for reconstruction from pairs of packets, and optimized for reconstruction from four packets. (**a**) Cameraman. (**b**) House. (**c**) Lena. (**d**) Barbara

exhibiting that, for squared-error measures, the total distortion can be computed as the sum of distortions associated with its nonoverlapping blocks. While this property is satisfied for any segmentation of the signal into nonoverlapping blocks, we will exemplify it here for blocks of equal sizes that allow using one block-level compression procedure for all the blocks.

Mirroring the definitions described in section "Preliminaries: Lossy Compression via Operational Rate-Distortion Optimization" for full-signal compression architectures, the block-level process corresponds to a function $C_b : \mathbb{R}^{N_b} \to \mathcal{B}_b$, mapping the N_b-dimensional signal-block domain to a discrete set $\mathcal{B}_b$ of binary compressed representations of blocks. The associated block decompression process is denoted by the function $F_b : \mathcal{B}_b \to \mathcal{S}_b$, mapping the binary compressed representations in $\mathcal{B}_b$ to their decompressed signal blocks from the discrete set $\mathcal{S}_b \subset \mathbb{R}^{N_b}$. The bit-cost evaluation function $R_b(\mathbf{v}_i)$ is defined to quantify the number of bits needed for

the compressed representation matching the decompressed signal block $\mathbf{v}_i \in \mathbb{R}^{N_b}$. Then, the compression of the nonoverlapping signal blocks $\{\mathbf{x}_i\}_{i\in\mathcal{I}}$ producing the decompressed blocks $\{\mathbf{v}_i\}_{i\in\mathcal{I}}$ requires a total bit budget satisfying

$$R(\mathbf{v}) = \sum_{i\in\mathcal{I}} R_b(\mathbf{v}_i). \tag{49}$$

Plugging the block-based compression design into the Lagrangian form (25) gives

$$\{\hat{\mathbf{v}}_i\}_{i\in\mathcal{I}} = \underset{\{\mathbf{v}_i\}_{i\in\mathcal{I}}\in\mathcal{S}_b}{\operatorname{argmin}} \sum_{i\in\mathcal{I}} R_b(\mathbf{v}_i) + \lambda \sum_{i\in\mathcal{I}} \|\mathbf{x}_i - \mathbf{v}_i\|_2^2 \tag{50}$$

that reduces to a set of block-level rate-distortion Lagrangian optimizations, i.e.,

$$\text{For } i \in \mathcal{I}: \quad \hat{\mathbf{v}}_i = \underset{\mathbf{v}_i\in\mathcal{S}_b}{\operatorname{argmin}} \; R_b(\mathbf{v}_i) + \lambda \|\mathbf{x}_i - \mathbf{v}_i\|_2^2. \tag{51}$$

Note that the block-level optimizations in (51) are independent and refer to the same Lagrangian multiplier λ. Commonly, compression designs are based on processing of low-dimensional blocks, allowing to practically address the block-level optimizations in (51). For example, one can evaluate the Lagrangian cost for all the elements in $\mathcal{S}_b$ (since this set is sufficiently small).

References

Afonso, M.V., Bioucas-Dias, J.M., Figueiredo, M.A.: Fast image recovery using variable splitting and constrained optimization. IEEE Trans. Image Process. **19**(9), 2345–2356 (2010)

Ahmad, R., Bouman, C.A., Buzzard, G.T., Chan, S., Reehorst, E.T., Schniter, P.: Plug and play methods for magnetic resonance imaging. arXiv preprint arXiv:1903.08616 (2019)

Bahat, Y., Efrat, N., Irani, M.: Non-uniform blind deblurring by reblurring. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 3286–3294 (2017)

Ballé, J., Laparra, V., Simoncelli, E.P.: End-to-end optimized image compression. In: Proceedings of ICLR (2017)

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

Beygi, S., Jalali, S., Maleki, A., Mitra, U.: Compressed sensing of compressible signals. In: IEEE International Symposium on Information Theory (ISIT), pp. 2158–2162 (2017a)

Beygi, S., Jalali, S., Maleki, A., Mitra, U.: An efficient algorithm for compression-based compressed sensing. arXiv preprint arXiv:1704.01992 (2017b)

Boyd, S., Parikh, N., Chu, E., Peleato, B., Eckstein, J.: Distributed optimization and statistical learning via the alternating direction method of multipliers. Found. Trends Mach. Learn. **3**(1), 1–122 (2011)

Brifman, A., Romano, Y., Elad, M.: Turning a denoiser into a super-resolver using plug and play priors. In: 2016 IEEE International Conference on Image Processing (ICIP), pp. 1404–1408. IEEE (2016)

Brifman, A., Romano, Y., Elad, M.: Unified single-image and video super-resolution via denoising algorithms. IEEE Trans. Image Process. **28**(12), 6063–6076 (2019)

Bruckstein, A.M., Holt, R.J., Netravali, A.N.: Holographic representations of images. IEEE Trans. Image Process. **7**(11), 1583–1597 (1998)
Bruckstein, A.M., Holt, R.J., Netravali, A.N.: On holographic transform compression of images. Int. J. Imag. Syst. Technol. **11**(5), 292–314 (2000)
Bruckstein, A.M., Ezerman, M.F., Fahreza, A.A., Ling, S.: Holographic sensing. arXiv preprint arXiv:1807.10899 (2018)
Burger, M., Dirks, H., Schonlieb, C.-B.: A variational model for joint motion estimation and image reconstruction. SIAM J. Imag. Sci. **11**(1), 94–128 (2018)
Buzzard, G.T., Chan, S.H., Sreehari, S., Bouman, C.A.: Plug-and-play unplugged: optimization-free reconstruction using consensus equilibrium. SIAM J. Imag. Sci. **11**(3), 2001–2020 (2018)
Chan, S.H.: Performance analysis of plug-and-play ADMM: a graph signal processing perspective. IEEE Trans. Comput. Imag. **5**(2), 274–286 (2019)
Chan, S.H., Wang, X., Elgendy, O.A.: Plug-and-play ADMM for image restoration: fixed-point convergence and applications. IEEE Trans. Comput. Imag. **3**(1), 84–98 (2017)
Chatterjee, P., Milanfar, P.: Is denoising dead? IEEE Trans. Image Process. **19**(4), 895–911 (2009)
Chou, P.A., Lookabaugh, T., Gray, R.M.: Optimal pruning with applications to tree-structured source coding and modeling. IEEE Trans. Inf. Theory **35**(2), 299–315 (1989)
Corona, V., Aviles-Rivero, A.I., Debroux, N., Graves, M., Le Guyader, C., Schönlieb, C.-B., Williams, G.: Multi-tasking to correct: motion-compensated mri via joint reconstruction and registration. In: International Conference on Scale Space and Variational Methods in Computer Vision, pp. 263–274. Springer (2019a)
Corona, V., Benning, M., Ehrhardt, M.J., Gladden, L.F., Mair, R., Reci, A., Sederman, A.J., Reichelt, S., Schönlieb, C.-B.: Enhancing joint reconstruction and segmentation with non-convex bregman iteration. Inverse Probl. **35**(5), 055001 (2019b)
Dabov, K., Foi, A., Katkovnik, V., Egiazarian, K.: Image denoising by sparse 3-D transform-domain collaborative filtering. IEEE Trans. Image Process. **16**(8), 2080–2095 (2007)
Dar, Y., Bruckstein, A.M.: Benefiting from duplicates of compressed data: shift-based holographic compression of images. J. Math. Imag. Vis. 1–14 **63**, 380–393 (2021)
Dar, Y., Bruckstein, A.M., Elad, M.: Image restoration via successive compression. In: Picture Coding Symposium (PCS), pp. 1–5 (2016a)
Dar, Y., Bruckstein, A.M., Elad, M., Giryes, R.: Postprocessing of compressed images via sequential denoising. IEEE Trans. Image Process. **25**(7), 3044–3058 (2016b)
Dar, Y., Elad, M., Bruckstein, A.M.: Compression for multiple reconstructions. In: IEEE International Conference on Image Processing (ICIP), pp. 440–444 (2018a)
Dar, Y., Elad, M., Bruckstein, A.M.: Optimized pre-compensating compression. IEEE Trans. Image Process. **27**(10), 4798–4809 (2018b)
Dar, Y., Elad, M., Bruckstein, A.M.: Restoration by compression. IEEE Trans. Sig. Process. **66**(22), 5833–5847 (2018c)
Dar, Y., Elad, M., Bruckstein, A.M.: System-aware compression. In: IEEE International Symposium on Information Theory (ISIT), pp. 2226–2230 (2018d)
Hong, T., Romano, Y., Elad, M.: Acceleration of red via vector extrapolation. J. Vis. Commun. Image Represent. **63**, 102575 (2019)
Kamilov, U.S., Mansour, H., Wohlberg, B.: A plug-and-play priors approach for solving nonlinear imaging inverse problems. IEEE Sig. Process. Lett. **24**(12), 1872–1876 (2017)
Kwan, C., Choi, J., Chan, S., Zhou, J., Budavari, B.: A super-resolution and fusion approach to enhancing hyperspectral images. Remote Sens. **10**(9), 1416 (2018)
Lai, W.-S., Huang, J.-B., Hu, Z., Ahuja, N., Yang, M.-H.: A comparative study for single image blind deblurring. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1701–1709 (2016)
Laparra, V., Berardino, A., Ballé, J., Simoncelli, E.P.: Perceptually optimized image rendering. J. Opt. Soc. Am. A **34**, 1511 (2017)
Liu, J., Moulin, P.: Complexity-regularized image denoising. IEEE Trans. Image Process. **10**(6), 841–851 (2001)

Moulin, P., Liu, J.: Statistical imaging and complexity regularization. IEEE Trans. Inf. Theory **46**(5), 1762–1777 (2000)

Natarajan, B.K.: Filtering random noise from deterministic signals via data compression. IEEE Trans. Sig. Process. **43**(11), 2595–2605 (1995)

Ono, S.: Primal-dual plug-and-play image restoration. IEEE Sig. Process. Lett. **24**(8), 1108–1112 (2017)

Ortega, A., Ramchandran, K.: Rate-distortion methods for image and video compression. IEEE Sig. Process. Mag. **15**(6), 23–50 (1998)

Rissanen, J.: MDL denoising. IEEE Trans. Inf. Theory **46**(7), 2537–2543 (2000)

Romano, Y., Elad, M., Milanfar, P.: The little engine that could: regularization by denoising (RED). SIAM J. Imag. Sci. **10**(4), 1804–1844 (2017)

Rond, A., Giryes, R., Elad, M.: Poisson inverse problems by the plug-and-play scheme. J. Vis. Commun. Image Represent. **41**, 96–108 (2016)

Rott Shaham, T., Michaeli, T.: Deformation aware image compression. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2453–2462 (2018)

Shoham, Y., Gersho, A.: Efficient bit allocation for an arbitrary set of quantizers. IEEE Trans. Acoust. Speech Sig. Process. **36**(9), 1445–1453 (1988)

Shukla, R., Dragotti, P.L., Do, M.N., Vetterli, M.: Rate-distortion optimized tree-structured compression algorithms for piecewise polynomial images. IEEE Trans. Image Process. **14**(3), 343–359 (2005)

Sreehari, S., Venkatakrishnan, S., Wohlberg, B., Buzzard, G.T., Drummy, L.F., Simmons, J.P., Bouman, C.A.: Plug-and-play priors for bright field electron tomography and sparse interpolation. IEEE Trans. Comput. Imag. **2**(4), 408–423 (2016)

Sullivan, G.J., Wiegand, T.: Rate-distortion optimization for video compression. IEEE Sig. Process. Mag. **15**(6), 74–90 (1998)

Sullivan, G.J., Ohm, J., Han, W.-J., Wiegand, T.: Overview of the high efficiency video coding (HEVC) standard. IEEE Trans. Circuits Syst. Video Technol. **22**(12), 1649–1668 (2012)

Sun, Y., Wohlberg, B., Kamilov, U.S.: An online plug-and-play algorithm for regularized image reconstruction. IEEE Trans. Comput. Imag.**5**, 395–408 (2019a)

Sun, Y., Xu, S., Li, Y., Tian, L., Wohlberg, B., Kamilov, U.S.: Regularized fourier ptychography using an online plug-and-play algorithm. In: ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 7665–7669. IEEE (2019b)

Tirer, T., Giryes, R.: Image restoration by iterative denoising and backward projections. IEEE Trans. Image Process. **28**(3), 1220–1234 (2018a)

Tirer, T., Giryes, R.: An iterative denoising and backwards projections method and its advantages for blind deblurring. In: 2018 25th IEEE International Conference on Image Processing (ICIP), pp. 973–977. IEEE (2018b)

Tirer, T., Giryes, R.: Back-projection based fidelity term for ill-posed linear inverse problems. arXiv preprint arXiv:1906.06794 (2019)

Venkatakrishnan, S.V., Bouman, C.A., Wohlberg, B.: Plug-and-play priors for model based reconstruction. In: IEEE GlobalSIP (2013)

Yazaki, Y., Tanaka, Y., Chan, S.H.: Interpolation and denoising of graph signals using plug-and-play ADMM. In: ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 5431–5435. IEEE (2019)

Zoran, D., Weiss, Y.: From learning models of natural image patches to whole image restoration. In: IEEE International Conference on Computer Vision (ICCV), pp. 479–486 (2011)