Image Restoration via Successive Compression

Yehuda Dar, Alfred M. Bruckstein, and Michael Elad
Computer Science Department
Technion – Israel Institute of Technology
{ydar,freddy,elad}@cs.technion.ac.il

***Abstract*—In this paper we propose a method for solving various imaging inverse problems via complexity regularization that leverages existing image compression techniques. Lossy compression has already been proposed in the past for Gaussian denoising – the simplest inverse problem. However, extending this approach to more complicated inverse problems (e.g., deblurring, inpainting, etc.) seemed to result in intractable optimization tasks. In this work we address this difficulty by decomposing the complicated optimization problem via the Half Quadratic Splitting approach, resulting in a sequential solution of a simpler ℓ_2-regularized inverse problem followed by a rate-distortion optimization, replaced by an efficient compression technique. In addition, we suggest an improved complexity regularizer that quantifies the average block-complexity in the restored signal, which in turn, extends our algorithm to rely on averaging multiple decompressed images obtained from compression of shifted images. We demonstrate the proposed scheme for inpainting of corrupted images, using leading image compression techniques such as JPEG2000 and HEVC.**

I. Introduction

Signal compression and image restoration are two fundamental and interconnected topics in signal processing. Image restoration relies on solving inverse problems and often requires regularization when the Bayesian approach is practiced. Past work has considered complexity, measured as the compression bit-cost of candidate solutions, as regularization for image restoration. Several studies (e.g., [1], [2]) suggested complexity-regularized solutions to the Gaussian denoising task, which is the simplest inverse problem, by applying lossy compression on the noisy signal.

The extension of the complexity-regularized approach to more complicated inverse problems (e.g., deblurring, inpainting, super-resolution, etc.) was studied in [3], where a thorough theoretical treatment of the problem was provided. However, the practical employment of the approach reached a dead-end in the form of highly intractable optimization tasks, that rarely reduce to reasonable structures. For example, this approach was demonstrated in [3] only for Poisson denoising and with a particularly designed compression architecture.

In this paper, we propose a methodology that enables practical solution of various complexity-regularized inverse problems, removing the limitations on the image deterioration model and the utilized compression method, hence, establishing a generic complexity-regularized approach for image restoration. We suggest to decouple the two intricate parts of the optimization problem via the useful Half Quadratic Splitting approach (for another use of it see [4]). Our approach results in an iterative solution involving simpler ℓ_2-regularized inverse problems followed by standard rate-distortion optimizations that can be replaced by any existing compression technique. The proposed approach can be viewed as the compression counterpart of the recent Plug-and-Play Priors framework [5], where restoration problems were solved by an iterative process that relies on an arbitrary Gaussian denoiser, allowing the solution of complicated problems (e.g., see [6]).

As many compression methods operate on non-overlapping blocks in the image, the corresponding complexity measure is shift-sensitive. In order to alleviate this shortcoming, we further extend the complexity regularization to measure the average complexity of all the overlapping blocks in the recovered image. This can be interpreted as a variant of the Expected Patch Log-Likelihood (EPLL) concept [4]. In our case, this approach enhances the proposed procedure, leading to an average of multiple decompressed images following compression of shifted images. Interestingly, this procedure recalls the cycle-spinning approach [7], originally proposed for wavelet-based denoising. Moreover, whereas the compression artifact-reduction techniques in [8], [9] suggest to enhance decompressed images by repeated compressions of their shifted versions that are averaged, here we generalize this approach to the solution of various image restoration tasks.

We demonstrate our approach for image inpainting, considering the noisy and the noiseless corruption models. The proposed technique is evaluated with the JPEG2000 and the HEVC still-image compression methods, showing impressive restoration results.

II. Complexity Regularization

Our starting point is the degradation model that defines the relation between the measurements and the desired signal: A signal $\mathbf{x} \in \mathbb{R}^N$ is corrupted via

$$\mathbf{y} = \mathbf{H}\mathbf{x} + \mathbf{n} \tag{1}$$

where $\mathbf{H}$ is a $M \times N$ matrix representing a deteriorating operation (such as blur, pixel erasure, decimation, etc.) and $\mathbf{n}$ is a M-length vector of white Gaussian noise[1] (zero mean and variance σ_n^2). The considered task is to restore $\mathbf{x}$ from its corrupted version $\mathbf{y}$. A popular approach for addressing this problem from a statistical perspective is the Maximum A-Posteriori (MAP) estimation, $\hat{\mathbf{x}} = \underset{\mathbf{x}}{\text{argmax}}\, p(\mathbf{x}|\mathbf{y})$, where $p(\mathbf{x}|\mathbf{y})$ is the posterior probability. Then, via Bayes rule,

[1]The proposed scheme could be easily generalized to handle other noise statistics, but we omit this here for the lack of space.

applying logarithm on the minimized function, and using the AWGN property in (1), we get

$$\hat{\mathbf{x}} = \underset{\mathbf{x}}{\operatorname{argmin}} \frac{1}{2\sigma_n^2} \left\| \mathbf{H}\mathbf{x} - \mathbf{y} \right\|_2^2 - \log p\left(\mathbf{x}\right) \tag{2}$$

where $p(\mathbf{x})$ is the prior probability, used here for evaluating the probability of the candidate solution. State-of-the-art methods often replace the term $-\log p\left(\mathbf{x}\right)$ with a general regularization function $s(\mathbf{x})$, which does not necessarily directly embody a prior probability function, providing the optimization form

$$\hat{\mathbf{x}} = \underset{\mathbf{x}}{\operatorname{argmin}} \left\| \mathbf{H}\mathbf{x} - \mathbf{y} \right\|_2^2 + \mu s(\mathbf{x}) \tag{3}$$

where $s(\mathbf{x})$ returns a lower value for a more likely candidate solution and $\mu \geq 0$ is a parameter, generalizing the role of σ_n^2, weighting the regularization effect.

The complexity-regularized approach follows the form (3) with a regularization function, $s(\cdot)$, that quantifies the complexity of the candidate solution. This can be achieved by letting $s(\mathbf{x})$ be the number of bits needed to digitally represent the signal $\mathbf{x}$. One can leverage existing compression designs that effectively compress signals based on established models and, as such, get an increased suitability to the underlying signal-model.

For the mathematical development of our method we consider a compression procedure operating on non-overlapping fixed-size blocks of the image by independently encoding them. Each block contains N_b samples. The block-compression is based on a codebook, $\mathcal{C}$, containing a finite set of block reproduction options, each coupled with a binary codeword for its compressed-domain representation. The assumed architecture allocates shorter codewords for more likely patterns, and we denote the length of the binary codeword of $\mathbf{z} \in \mathcal{C}$ as $r(\mathbf{z})$. Thus, a signal $\mathbf{x}$, comprising the blocks $\{\mathbf{x}_i\}_{i\in\mathcal{B}}$ (where $\mathcal{B}$ denotes the index set of blocks in the non-overlapping partitioning of the signal), is compressed via the following rate-distortion optimization (for the commonly used Mean-Squared-Error distortion metric):

$$\{\tilde{\mathbf{x}}_i\}_{i\in\mathcal{B}} = \underset{\{\mathbf{v}_i\}_{i\in\mathcal{B}}\in\mathcal{C}}{\operatorname{argmin}} \sum_{i\in\mathcal{B}} \left\| \mathbf{x}_i - \mathbf{v}_i \right\|_2^2 + \lambda \sum_{i\in\mathcal{B}} r(\mathbf{v}_i), \tag{4}$$

where $\lambda \geq 0$ is a Lagrange multiplier controlling the compression bit-cost. Also note that the optimization in (4) can be solved independently for each block [10], [11].

Let us reformulate (4) as follows. First, let $\mathbf{P}_i$ be a matrix that extracts the i^{th} block from a full signal by standard multiplication, i.e., $\mathbf{P}_i\mathbf{x} = \mathbf{x}_i$, where this operation can extract any block of the signal, not necessarily only from the non-overlapping decomposition $\mathcal{B}$ (note that similar matrices are widely used, e.g., see [4]). Then, we note that a full signal can be expressed as a combination of its non-overlapping blocks that are located in their respective positions by multiplication by $\mathbf{P}_i^T$, namely, $\mathbf{x} = \sum_{i\in\mathcal{B}} \mathbf{P}_i^T \mathbf{x}_i$. Accordingly, the above algebraic tools for block handling are utilized to reformulate the rate-distortion optimization in (4) to

$$\min_{\{\mathbf{v}_i\}_{i\in\mathcal{B}}\in\mathcal{C}} \left\| \mathbf{x} - \sum_{i\in\mathcal{B}} \mathbf{P}_i^T \mathbf{v}_i \right\|_2^2 + \lambda \sum_{i\in\mathcal{B}} r(\mathbf{v}_i). \tag{5}$$

We further define the group of candidate solutions, corresponding to the full signal, based on non-overlapping blocks as

$$\mathcal{C}_{\mathcal{B}} = \left\{ \mathbf{v} \;\middle|\; \mathbf{v} = \sum_{i\in\mathcal{B}} \mathbf{P}_i^T \mathbf{v}_i, \quad \{\mathbf{v}_i\}_{i\in\mathcal{B}} \in \mathcal{C} \right\}. \tag{6}$$

In addition, we define the total bit cost for $\mathbf{v} \in \mathcal{C}_{\mathcal{B}}$ as $r_{tot}(\mathbf{v}) \triangleq \sum_{i\in\mathcal{B}} r(\mathbf{P}_i\mathbf{v})$. Then, we can reformulate the rate-distortion optimization in (5) to the form

$$\tilde{\mathbf{x}} = \underset{\mathbf{v}\in\mathcal{C}_{\mathcal{B}}}{\operatorname{argmin}} \left\| \mathbf{x} - \mathbf{v} \right\|_2^2 + \lambda r_{tot}(\mathbf{v}). \tag{7}$$

We now return to the inverse problem formulation in (3) and update it by setting the complexity regularizer $s(\mathbf{x}) = \bar{r}_{tot}(\mathbf{x}) \triangleq \sum_{i\in\mathcal{B}} \bar{r}(\mathbf{P}_i\mathbf{x})$, where $\bar{r}(\mathbf{z})$ is defined for any $\mathbf{z} \in \mathbb{R}^{N_b}$ as

$$\bar{r}(\mathbf{z}) = \begin{cases} r\left(\mathbf{z}\right) & ,\mathbf{z} \in \mathcal{C} \\ \infty & ,\mathbf{z} \notin \mathcal{C} \end{cases}, \tag{8}$$

i.e., blocks that are not in the codebook have infinite complexity. Thus, the complexity-regularized restoration task is formulated as

$$\hat{\mathbf{x}} = \underset{\mathbf{x}}{\operatorname{argmin}} \left\| \mathbf{H}\mathbf{x} - \mathbf{y} \right\|_2^2 + \mu \bar{r}_{tot}(\mathbf{x}), \tag{9}$$

noting that due to (8) the solution candidates are only within $\mathcal{C}_{\mathcal{B}}$ as defined in (6).

When the corruption is only an additive white Gaussian noise (i.e., $\mathbf{H} = \mathbf{I}$), the problem in (9) takes the form of the rate-distortion optimization in (7), here compressing the signal $\mathbf{y}$. However, for more intricate models involving arbitrary $\mathbf{H}$, the optimization in general does not lend itself to a standard rate-distortion optimization form nor having a practically convenient procedure to its solution. In the next section we explain how this optimization structure can be accommodated to provide a constructive procedure to its treatment.

III. The Proposed Method

A. Total Complexity of Non-Overlapping Blocks

Our goal is to provide a method that can practically solve the optimization problem in (9). We start by rewriting (9) as

$$\hat{\mathbf{x}} = \underset{\mathbf{x}}{\operatorname{argmin}} \left\| \mathbf{H}\mathbf{x} - \mathbf{y} \right\|_2^2 + \mu \sum_{i\in\mathcal{B}} \bar{r}(\mathbf{P}_i\mathbf{x}), \tag{10}$$

noting that for a general $\mathbf{H}$ the above problem cannot be decomposed to independent treatment of the blocks. We suggest to remedy this via the Half Quadratic Splitting approach, applied here by introducing the auxiliary variables $\{\mathbf{z}_i\}_{i\in\mathcal{B}}$, where $\mathbf{z}_i$ is associated with the i^{th} non-overlapping block. More explicitly, we update the optimization (10) to

$$\begin{aligned} \left(\hat{\mathbf{x}}, \{\hat{\mathbf{z}}_i\}_{i\in\mathcal{B}}\right) &= \underset{\mathbf{x},\{\mathbf{z}_i\}_{i\in\mathcal{B}}}{\operatorname{argmin}} \left\| \mathbf{H}\mathbf{x} - \mathbf{y} \right\|_2^2 + \mu \sum_{i\in\mathcal{B}} \bar{r}(\mathbf{z}_i) \\ \text{s.t.} \quad & \mathbf{z}_i = \mathbf{P}_i\mathbf{x} \quad ,\text{for } i \in \mathcal{B}. \end{aligned} \tag{11}$$

Then, an unconstrained problem is formed by extending the cost function to include quadratic penalizing terms for the above equality constraints:

$$\left(\hat{\mathbf{x}}, \{\hat{\mathbf{z}}_i\}_{i\in\mathcal{B}}\right) = \underset{\mathbf{x},\{\mathbf{z}_i\}_{i\in\mathcal{B}}}{\operatorname{argmin}} \; \|\mathbf{H}\mathbf{x}-\mathbf{y}\|_2^2 + \frac{\beta}{2}\sum_{i\in\mathcal{B}}\|\mathbf{P}_i\mathbf{x}-\mathbf{z}_i\|_2^2 + \mu\sum_{i\in\mathcal{B}}\bar{r}(\mathbf{z}_i), \quad (12)$$

where $\beta > 0$ is a parameter enforcing each $\mathbf{z}_i$ to be close to $\mathbf{P}_i\mathbf{x}$, accordingly, the constraints in (11) will be more closely satisfied for a larger β. Applying an alternating minimization approach on the problem in (12), together with increasing the parameter β in each iteration, provides an effective iterative solution (e.g., see [4]), where its t^{th} iteration is formed of the following three steps:

$$\hat{\mathbf{x}}^{(t)} = \underset{\mathbf{x}}{\operatorname{argmin}} \, \|\mathbf{H}\mathbf{x}-\mathbf{y}\|_2^2 + \frac{\beta^{(t)}}{2}\sum_{i\in\mathcal{B}}\left\|\mathbf{P}_i\mathbf{x}-\hat{\mathbf{z}}_i^{(t-1)}\right\|_2^2 \quad (13)$$

$$\hat{\mathbf{z}}_i^{(t)} = \underset{\mathbf{z}_i}{\operatorname{argmin}} \, \frac{\beta^{(t)}}{2}\left\|\mathbf{P}_i\hat{\mathbf{x}}^{(t)}-\mathbf{z}_i\right\|_2^2 + \mu\bar{r}(\mathbf{z}_i), \; i\in\mathcal{B} \quad (14)$$

$$\text{Set } \beta^{(t+1)} \text{ as an increment of } \beta^{(t)}. \quad (15)$$

Importantly, we can identify the set of optimization problems in (14) as a full-image rate-distortion optimization based on a partitioning of non-overlapping blocks, operating at a Lagrange multiplier value of $\lambda = \frac{2\mu}{\beta^{(t)}}$. We denote the full image compression-decompression procedure in (14) as $\hat{\mathbf{z}}^{(t)} = CompressDecompress_\lambda\left(\hat{\mathbf{x}}^{(t)}\right)$, where $\hat{\mathbf{z}}^{(t)}$ is the decompressed image composed of the non-overlapping decompressed blocks.

In addition, the analytic solution of the first optimization problem in (13) is a weighted averaging of the corrupted image with the decompressed blocks, explicitly formulated as

$$\hat{\mathbf{x}}^{(t)} = \left(\mathbf{H}^T\mathbf{H} + \frac{\beta^{(t)}}{2}\mathbf{I}\right)^{-1}\left(\mathbf{H}^T\mathbf{y} + \frac{\beta^{(t)}}{2}\sum_{i\in\mathcal{B}}\mathbf{P}_i^T\hat{\mathbf{z}}_i^{(t-1)}\right)$$

where, in the last calculation, we used the fact that $\sum_{i\in\mathcal{B}}\mathbf{P}_i^T\mathbf{P}_i = \mathbf{I}$ as the blocks do not overlap.

B. Average Complexity of All Overlapping Blocks

We return to the basic formulation of the inverse problem given in (10) and suggest to extend the regularization to measure the average complexity (bit-cost) of all the overlapping blocks in the signal, then the optimization becomes

$$\hat{\mathbf{x}} = \underset{\mathbf{x}}{\operatorname{argmin}} \, \|\mathbf{H}\mathbf{x}-\mathbf{y}\|_2^2 + \mu\sum_{i\in\mathcal{B}^*}\bar{r}(\mathbf{P}_i\mathbf{x}), \quad (16)$$

where $\mathcal{B}^*$ is the set containing the indices of all the overlapping blocks in the image. In addition, note that the normalization by the size of $\mathcal{B}^*$, needed for the averaging, is included here in the value of μ. It is important to note that the solution of (16) is no longer restricted to $\mathcal{C}_\mathcal{B}$ defined in (6).

The suggested problem in (16) is even more intricate than the previous formulation in (10) as it includes all the overlapping blocks. Nevertheless, the approach of Half Quadratic Splitting helps us again to establish a practical solution. Here, the auxiliary variables are defined as $\{\mathbf{z}_i\}_{i\in\mathcal{B}^*}$, where $\mathbf{z}_i$ corresponds to the i^{th} overlapping block. Accordingly, the optimization problem (16) is updated to

$$\left(\hat{\mathbf{x}}, \{\hat{\mathbf{z}}_i\}_{i\in\mathcal{B}^*}\right) = \underset{\mathbf{x},\{\mathbf{z}_i\}_{i\in\mathcal{B}^*}}{\operatorname{argmin}} \; \|\mathbf{H}\mathbf{x}-\mathbf{y}\|_2^2 + \mu\sum_{i\in\mathcal{B}^*}\bar{r}(\mathbf{z}_i)$$
$$\text{s.t.} \qquad \mathbf{z}_i = \mathbf{P}_i\mathbf{x} \qquad ,\text{for } i\in\mathcal{B}^*. \quad (17)$$

As in the former subsection we insert the constraints as quadratic penalizing terms and apply alternating minimization, and the achieved iterative solution has the following three steps in each iteration:

$$\hat{\mathbf{x}}^{(t)} = \underset{\mathbf{x}}{\operatorname{argmin}} \, \|\mathbf{H}\mathbf{x}-\mathbf{y}\|_2^2 + \frac{\beta^{(t)}}{2}\sum_{i\in\mathcal{B}^*}\left\|\mathbf{P}_i\mathbf{x}-\hat{\mathbf{z}}_i^{(t-1)}\right\|_2^2 \quad (18)$$

$$\hat{\mathbf{z}}_i^{(t)} = \underset{\mathbf{z}_i}{\operatorname{argmin}} \, \frac{\beta^{(t)}}{2}\left\|\mathbf{P}_i\hat{\mathbf{x}}^{(t)}-\mathbf{z}_i\right\|_2^2 + \mu\bar{r}(\mathbf{z}_i), \; i\in\mathcal{B}^* \quad (19)$$

$$\text{Set } \beta^{(t+1)} \text{ as an increment of } \beta^{(t)}. \quad (20)$$

However, the meaning of the optimizations in (18) and (19) is different than before due to the overlapping blocks.

The step (19) consists of block-level rate-distortion optimizations for all the overlapping blocks in the image. We can identify this stage as multiple applications of a full image compression-decompression procedure, each operates on a different set of non-overlapping blocks that forms a corresponding shifted version of the image. For an image $\mathbf{x}$, also having a 2D form, and a compression block-size of $\sqrt{N_b}\times\sqrt{N_b}$ samples, there are N_b shifted block-grids. The j^{th} shift ($j = 1, ..., N_b$) corresponds to a rectangular image defined by taking the j^{th} pixel of $\mathbf{x}$'s first block as its upper left corner and defining its entire domain by containing an integer number of blocks. We denote the j^{th} shifted image as $shift_j\{\mathbf{x}\}$. Additionally, we denote the set of block indices that participate in the j^{th} shifted image as $\mathcal{B}^j$ (noting that $\mathcal{B}^1 = \mathcal{B}$), accordingly, $\mathcal{B}^* = \cup_{j=1}^{N^b}\mathcal{B}^j$. This lets us produce the decompressed blocks $\left\{\hat{\mathbf{z}}_i^{(t)}\right\}_{i\in\mathcal{B}^*}$ by multiple full-image compression decompression applications, i.e., $\hat{\mathbf{z}}^{j,(t)} = CompressDecompress_\lambda\left(shift_j\left\{\hat{\mathbf{x}}^{(t)}\right\}\right)$ for $j = 1, ..., N_b$, where the j^{th} decompressed image[2], $\hat{\mathbf{z}}^{j,(t)}$, consists of the blocks $\left\{\hat{\mathbf{z}}_i^{(t)}\right\}_{i\in\mathcal{B}^j}$ and $\lambda = \frac{2\mu}{\beta^{(t)}}$.

The optimization in (18) is again a weighted averaging, but now it considers all the overlapping decompressed blocks, the analytic solution is formulated as

$$\hat{\mathbf{x}}^{(t)} = \left(\mathbf{H}^T\mathbf{H} + \frac{\beta^{(t)}}{2}\sum_{i\in\mathcal{B}^*}\mathbf{P}_i^T\mathbf{P}_i\right)^{-1} \times \left(\mathbf{H}^T\mathbf{y} + \frac{\beta^{(t)}}{2}\sum_{i\in\mathcal{B}^*}\mathbf{P}_i^T\hat{\mathbf{z}}_i^{(t-1)}\right). \quad (21)$$

[2]For mathematical convenience, $\hat{\mathbf{z}}^{j,(t)}$ is considered to have the size of $\mathbf{x}$, as can be achieved by appropriately locating and padding the decompressed j^{th} shift of $\mathbf{x}$.

Here the term $\sum_{i\in\mathcal{B}^*} \mathbf{P}_i^T\mathbf{P}_i$ is a diagonal matrix, where each diagonal element corresponds to a signal sample and counts the number of decompressed blocks that it participates in. Furthermore, the averaging in (21) can be directly applied on the full decompressed images $\hat{\mathbf{z}}^{j,(t)}$ ($j = 1, ..., N_b$) as also appear in the summary in Algorithm 1, where we refer to the sum of decompressed images via the definition $\hat{\mathbf{z}}_{sum}^{(t)} \triangleq \sum_{j=1}^{N_b} \left(\sum_{i\in\mathcal{B}^j} \mathbf{P}_i^T\mathbf{P}_i \right) \hat{\mathbf{z}}^{j,(t)}$.

As we suggest to apply compression-decompression by utilizing a standardized image compression method it is important to note the following issues. First, we argue that our technique can leverage also compression methods that do not accurately follow the above defined rate-distortion optimization. In addition, many compression softwares do not necessarily take the Lagrange multiplier λ as the parameter determining their working point, and may use instead various replacements such as quality parameter, compression ratio, or output bit-rate. Therefore, our method should practically adapt itself to the chosen compression method and its specific controlling parameter, denoted as θ, such that in each iteration the quality (or output bit-rate) of the compression is increased. Without loss of generality, we consider here θ that its reduction increases the output quality and, therefore, our iteration update rule follows $\theta^{(t+1)} < \theta^{(t)}$. Furthermore, the value of this parameter can be set independently of the value of $\beta^{(t)}$. These generalizations are also included in Algorithm 1.

Algorithm 1 Proposed Method: Solution Based on Average Complexity

1: Initialize $\hat{\mathbf{z}}_{sum}^{(0)}$ (depending on the deterioration type).
2: $t = 1$, $\beta^{(1)} = \beta_1$, $\theta^{(1)} = \theta_1$
3: **repeat**
4: $\quad \hat{\mathbf{x}}^{(t)} = \left(\mathbf{H}^T\mathbf{H} + \frac{\beta^{(t)}}{2} \sum_{i\in\mathcal{B}^*} \mathbf{P}_i^T\mathbf{P}_i \right)^{-1} \times \left(\mathbf{H}^T\mathbf{y} + \frac{\beta^{(t)}}{2} \hat{\mathbf{z}}_{sum}^{(t-1)} \right)$
5: $\quad \hat{\mathbf{z}}^{j,(t)} = CompressDecompress_{\theta^{(t)}} \left(shift_j \left\{ \hat{\mathbf{x}}^{(t)} \right\} \right)$, $(j = 1, ..., N_b)$.
6: $\quad \hat{\mathbf{z}}_{sum}^{(t)} = \sum_{j=1}^{N_b} \left(\sum_{i\in\mathcal{B}^j} \mathbf{P}_i^T\mathbf{P}_i \right) \hat{\mathbf{z}}^{j,(t)}$
7: $\quad$ Set $\beta^{(t+1)}$ as an increment of $\beta^{(t)}$
8: $\quad$ Set $\theta^{(t+1)}$ as a decrease of $\theta^{(t)}$
9: $\quad t \leftarrow t + 1$
10: **until** stopping criterion is satisfied

IV. Application to Image Inpainting

In this section we demonstrate a specific utilization of the proposed technique for solving the inpainting problem.

A. Noisy Corruption Model

The corruption model of this problem is defined in the form $\mathbf{y} = \mathbf{H}\mathbf{x} + \mathbf{n}$, where the matrix $\mathbf{H}$ is of $N \times N$ size and diagonal form, with main diagonal entries that can be zero or one. Accordingly, the product $\mathbf{H}\mathbf{x}$ results in an N-length vector where its k^{th} sample is determined by $\mathbf{H}$: if $\mathbf{H}[k,k] = 0$ then it is zero, and for $\mathbf{H}[k,k] = 1$ it equals to the corresponding sample of $\mathbf{x}$.

Now we turn to specifically interpret the optimization in step 4 of Algorithm 1[3]. First, since $\mathbf{H}$ is a square diagonal matrix then $\mathbf{H}^T = \mathbf{H}$. Consequently, the multiplication $\mathbf{H}^T\mathbf{y}$ results in a vector $\mathbf{y}$ with zeroed entries according to $\mathbf{H}$'s structure. Second, we note that $\mathbf{H}^T\mathbf{H} = \mathbf{H}$, so the weight matrix is correctly adapted to the missing pixels in $\mathbf{H}^T\mathbf{y}$. Finally, also recall that $\mathbf{W} \triangleq \sum_{i\in\mathcal{B}^*} \mathbf{P}_i^T\mathbf{P}_i$ is a diagonal matrix. Hence, the computation of step 4 of Algorithm 1 is eased to be componentwise, such that the k^{th} sample of $\hat{\mathbf{x}}^{(t)}$ is:

$$\hat{\mathbf{x}}^{(t)}[k] = \frac{\mathbf{H}[k,k] \cdot \mathbf{y}[k] + \frac{1}{2}\beta^{(t)} \hat{\mathbf{z}}_{sum}^{(t-1)}[k]}{\mathbf{H}[k,k] + \frac{1}{2}\beta^{(t)} \mathbf{W}[k,k]}. \tag{22}$$

B. Noiseless Corruption Model

The noiseless inpainting problem considers the corruption model $\mathbf{y} = \mathbf{H}\mathbf{x}$, i.e., again pixels are zeroed according to $\mathbf{H}$, however, no noise is involved. The fact that there is no noise on the pixels that were not erased lets us update the computation of step 4 of Algorithm 1, from the formula given in (22), to

$$\hat{\mathbf{x}}^{(t)}[k] = \begin{cases} \mathbf{y}[k] & \text{, for } \mathbf{H}[k,k] = 1 \\ \frac{\hat{\mathbf{z}}_{sum}^{(t-1)}[k]}{\mathbf{W}[k,k]} & \text{, for } \mathbf{H}[k,k] = 0 \end{cases} \tag{23}$$

while the remaining procedure is as outlined in Algorithm 1.

C. Experimental Results

We here provide experimental results demonstrating the proposed method for recovery from corruption by the above inpainting models. The proposed method was evaluated for its work in conjunction with JPEG2000 (as provided in Matlab) and HEVC still-image compression[4] techniques. As will be described next, we set our method to run many iterations and, therefore, we suggest to consider only a part of all the possible shifts, i.e., taking a portion of $\mathcal{B}^*$[5]. In addition, the parameters should be set according to the utilized compression method. Moreover, the parameter values should be wisely paired with the number of iterations to conduct, as well as with the number of image shifts. All the following experiments share the settings $\beta^{(t+1)} = 1.1\beta^{(t)}$ and $\beta_1 = 0.001$, and the JPEG2000 applications are at a compression-ratio of $\theta^{(t)} = \max\{1, (2\mu)/\beta^{(t)}\}$.

Let us start from the noisy deterioration model, we applied Algorithm 1 together with the computation stage in (22) to images degraded by noise and pixel erasure at rates of 25%,

[3]As for the initialization step, $\hat{\mathbf{z}}_{sum}^{(0)}$ is set to $\mathbf{y}$ and a graylevel value of 128 replaces the missing pixels.

[4]Using the BPG software (0.9.6) available at http://bellard.org/bpg/.

[5]The shifts are taken by considering the rectangular cropped images which have their upper-left corner pixel within a $a \times a$ block in the upper-left corner of the full image, and their bottom-right corner pixel is the same as of the full image. This extends the mathematical developments in the previous section as practical compression handles arbitrarily sized rectangular images.

(a) Corrupted (b) Exemplar-based [12] (21.60dB) (c) Deterministic-Annealing [13] (26.81dB) (d) Proposed JPEG2000 (25.67dB) (e) Proposed HEVC (25.60dB)

Fig. 2. Restoration of Lena (512×512) from noiseless deterioration of missing square blocks (each of 16×16 pixels). The PSNR values in this experiment are only of the missing regions.

(a) Corrupted (8.87dB) (b) Reconstruction (28.97dB)

Fig. 1. Restoration of Barbara (512×512) from deterioration of 50% missing pixels and noise of $\sigma_n = 5$. The restoration utilized JPEG2000 compression.

TABLE I
NOISY INPAINTING: PSNR RESULTS

Image 512x512	**Missing Pixels**	$\sigma_n = 5$		$\sigma_n = 15$	
		Deteriorated	Recovered	Deteriorated	Recovered
Lena	75%	6.92	31.16	6.86	28.27
	50%	8.67	34.47	8.57	29.12
	25%	11.66	36.54	11.46	29.21
Barbara	75%	7.13	25.29	7.06	24.36
	50%	8.87	28.97	8.77	26.70
	25%	11.87	32.71	11.67	28.26

50% and 75%. We apply our method based on JPEG2000 with the following specific settings: 45 iterations, $\mu = 0.25$, considering shifts in the 8×8 upper-left block. The results in Table I and Fig. 1 show that the images were well restored with respect to the addressed deterioration.

In the noiseless deterioration model we consider a corrupted 512x512 pixels image with repeatedly missing square blocks (Fig. 2a). We applied Algorithm 1 in conjunction with the calculation in (23). Here we show results obtained by utilizing JPEG2000 (Fig. 2d) and HEVC (Fig. 2e). Comparison to other techniques (see Fig. 2) shows that our method achieves good results both objectively and subjectively. Our method based on JPEG2000 is applied here with the following specific settings: 60 iterations, $\mu = 2$, considering shifts in the 20×20 upper-left block. In the application based on HEVC we found it effective to set the initial quality parameter to 51 (its maximal value) and decrease it by one, once in every three iterations.

V. CONCLUSION

In this paper we proposed a method that revives the utilization of complexity regularization for various image restoration tasks. The complicated inverse problem was decomposed to an iterative procedure consisting of solution of a simpler ℓ_2-regularized inverse problem followed by a rate-distortion optimization, which we further suggested to replace by an efficient independent compression technique. Moreover, we revised the regularization to measure the average block-complexity in the restored signal, which, in turn, extended the former algorithm to rely on the average of multiple decompressed images. The proposed approach was demonstrated for the inpainting problem, addressed by utilization of the JPEG2000 and HEVC image compression techniques.

ACKNOWLEDGEMENT

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

REFERENCES

[1] B. K. Natarajan, "Filtering random noise from deterministic signals via data compression," *IEEE Trans. Signal Process.*, vol. 43, no. 11, pp. 2595–2605, 1995.

[2] J. Liu and P. Moulin, "Complexity-regularized image denoising," *IEEE Trans. Image Process.*, vol. 10, no. 6, pp. 841–851, 2001.

[3] P. Moulin and J. Liu, "Statistical imaging and complexity regularization," *IEEE Trans. Inf. Theory*, vol. 46, no. 5, pp. 1762–1777, 2000.

[4] D. Zoran and Y. Weiss, "From learning models of natural image patches to whole image restoration," in *IEEE ICCV*, 2011.

[5] S. V. Venkatakrishnan, C. A. Bouman, and B. Wohlberg, "Plug-and-play priors for model based reconstruction," in *IEEE GlobalSIP*, 2013.

[6] Y. Dar, A. M. Bruckstein, M. Elad, and R. Giryes, "Postprocessing of compressed images via sequential denoising," *IEEE Trans. Image Process.*, vol. 25, no. 7, pp. 3044–3058, 2016.

[7] R. R. Coifman and D. L. Donoho, "Translation-invariant de-noising," in *Wavelets and Statistics*. Springer, 1995, pp. 125–150.

[8] A. Nosratinia, "Enhancement of JPEG-compressed images by re-application of jpeg," *J. VLSI Signal Process. Syst. for Signal, Image and Video Technol.*, vol. 27, no. 1-2, pp. 69–79, 2001.

[9] ——, "Postprocessing of JPEG-2000 images to remove compression artifacts," *IEEE Signal Process. Lett.*, vol. 10, no. 10, 2003.

[10] Y. Shoham and A. Gersho, "Efficient bit allocation for an arbitrary set of quantizers," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 36, no. 9, pp. 1445–1453, 1988.

[11] A. Ortega and K. Ramchandran, "Rate-distortion methods for image and video compression," *IEEE Signal Process. Mag.*, vol. 15, no. 6, pp. 23–50, 1998.

[12] A. Criminisi, P. Pérez, and K. Toyama, "Region filling and object removal by exemplar-based image inpainting," *IEEE Trans. Image Process.*, vol. 13, no. 9, pp. 1200–1212, 2004.

[13] X. Li, "Image recovery via hybrid sparse representations: A deterministic annealing approach," *IEEE J. Sel. Topics Signal Process.*, vol. 5, no. 5, pp. 953–962, 2011.