Published in Image Processing On Line on 2026–06–00.
Submitted on 2025–10–02, accepted on 2025–05–19.
ISSN 2105–1232 © 2026 IPOL & the authors CC–BY–NC–SA
This article is available online with supplementary materials,
software, datasets and online demo at
https://doi.org/10.5201/ipol.2026.646

# Revisit of JPEG Decompression with Artifact Suppression by Total Variation and its Variants

Yanhao Li[1], Quentin Bammey[2,3], Marina Gardella[1], Rafael Grompone von Gioi[1], Miguel Colom[1], Jean-Michel Morel[4]

[1]Université Paris-Saclay, ENS Paris-Saclay, CNRS, Centre Borelli, Gif-sur-Yvette, France
{yanhao.li, marina.gardella, rafael.grompone, miguel.colom-barco}@ens-paris-saclay.fr
[2]École Polytechnique Fédérale de Lausanne, Image and Visual Representation Lab (IVRL), Lausanne, Switzerland
[3]LTCI, Télécom Paris, Institut Polytechnique de Paris, Institut Mines-Télécom, Palaiseau, France
quentin.bammey@telecom-paris.fr
[4]Lingnan University, Tuen Mun, Hong Kong
jeanmichelmorel@ln.edu.hk

*Communicated by* Mario González and Axel Davy  *Demo edited by* Yanhao Li

## Abstract

The question of JPEG artifacts reduction has existed since the birth of JPEG compression, and gained importance due to the popularity of JPEG compression. Suppression approaches are called idempotent when they guarantee the preservation of quantized DCT coefficients both before and after recompression using the same quantization matrix. We briefly revisit the idempotent JPEG decompression method based on total variation (TV) minimization using three variant costs: classical TV, adapted TV and Dirichlet integral. We review the constrained optimization problem of the JPEG decompression and its solution by gradient descent with projection onto a convex set, followed by a detailed description of its implementation. Both quantitative and qualitative experiments demonstrate the efficacy of this decompression method in eliminating artifacts and the over-optimization issue that smooths out original textures. Additionally, we investigate the performance differences among the three variant cost functions.

## Source Code

The reviewed source code and documentation for this algorithm are available from the web page of this article[1]. Usage instructions are included in the `README.md` file of the archive.

**Keywords:** JPEG decompression; total variation; inverse problem

[1]https://doi.org/10.5201/ipol.2026.646

# 1 Introduction

JPEG has been and still is the most widely used image compression format. Yet, it suffers from strong artifacts caused by the lossy quantization of DCT coefficients and severe chroma subsampling. The most widely used standard decompression scheme takes the center of the quantization interval of each DCT coefficient to reconstruct the image. This leaves noticeable distortion including ringing and blocking artifacts as the relationship between neighboring blocks are ignored, as is shown in Figure 1.

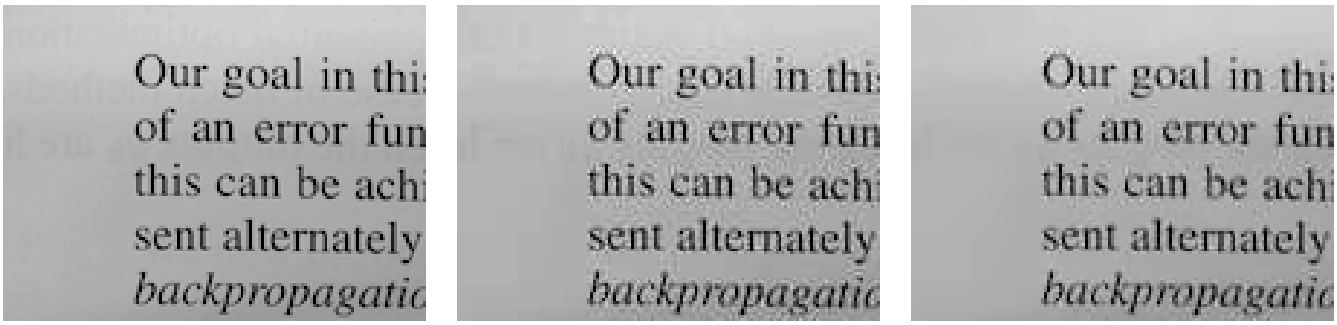



Figure 1: Left: original image; middle: decompressed image using the standard decoding; right: decompressed image with the minimization of TV. Noticeable distortion is caused by the lossy JPEG compression when standard decoding is used to decompress the image, including blocking artifacts in the flat area and ringing artifacts near the contrasted contents (e.g. the text). The blocking artifacts are due to the independent compression and decompression on each $8 \times 8$ block that neglects the continuity between adjacent blocks. The ringing artifacts are caused by the strong quantization of the high-frequency DCT coefficients that leads to the loss of precision of high frequencies. The decompressed image using the decompression with artifact suppression by total variation contains almost no blocking artifacts and less ringing artifacts.

In order to remove these undesirable artifacts, one type of method focuses on restoring the original image by considering the relationships between adjacent pixels while imposing the compression consistency with respect to the quantization limits, so that the decompressed image after recompression with the same quantization table has the same quantized coefficients as before. Such methods therefore are idempotent. Besides the standard decompression that is idempotent, one can minimize the total variation (TV) of the decoded image by choosing adequately DCT values that are still bounded by the coefficient intervals, to satisfy the idempotence. This method has been studied in previous works [2, 5, 6]. In this article, we briefly revisit the JPEG decompression method with TV-minimization for artifact suppression using the standard TV cost, the adapted TV cost proposed in [2], and the Dirichlet integral cost which has a similar formula to the TV cost.

# 2 Related Work

The suppression of JPEG artifacts has been studied by researchers since the birth of JPEG compression. One main direction of research is deblocking, that aims at restoring the image contents and removing the blocking artifacts of a decompressed JPEG image without directly considering the quantized coefficients in the JPEG file. Jung et al. [20] proposed a deblocking method based on the sparse representation of image contents modeled by the K-SVD algorithm. Xiong et al. [27] introduced a deblocking method using wavelet representations. Fu et al. [16] introduced a deep convolutional sparse coding network architecture for blocking artifact reduction. Fu et al.[15] invented an interpretable deep network to learn both pixel-level regressive prior and semantic-level discriminative prior for JPEG deblocking. Sun et al. [26] proposed a convolutional neural network on the DCT domain to learn the mapping relationship between JPEG images and original images to reduce compression artifacts.

Another main axis is the decompression of a JPEG image by considering the constraints imposed by the quantization of DCT coefficients. Bredies and Holler [5] conducted an in-depth study of the general problem of TV-minimization for JPEG decompression, and extended the prominent primal-dual algorithm of Chambolle and Pock [8] supplemented by a primal-dual gap stopping criterion to the decompression problem. The same authors also developed a method for decompression with zooming [6]. Chang et al. [10] combined the K-SVD dictionary learning method and TV-minimization method for suppressing blocky artifacts while preserving textures. Alter et al. [2] presented a weighted TV model to penalize pixels differently in the center and on the edges of a block. Ehrlich et al. [13] presented a neural network with parameterization by the quantization matrix to correct JPEG artifacts in the DCT domain.

# 3 JPEG Compression and its Standard Decompression

Throughout this paper, we use the following notations:

| Notation | Description |
|---|---|
| $u_0$ | Original uncompressed image |
| $\tilde{u}$ | Decompressed image |
| $u$ | Variable for the decompressed image to be optimized |
| $\tilde{v}$ | Quantized DCT coefficients from the JPEG file |
| $q \in \mathbb{R}^{8\times 8}$ | JPEG quantization table |
| $\mathbf{D}$ | Block-wise $8 \times 8$ DCT operator |
| $\mathbf{D}^{-1}$ | Block-wise $8 \times 8$ IDCT operator |
| $\mathcal{Q}$ | Quantization operator |
| $\mathcal{D}$ | Dequantization operator |
| $J(u)$ | Regularization cost function |

Table 1: Summary of notations.

## 3.1 JPEG Compression

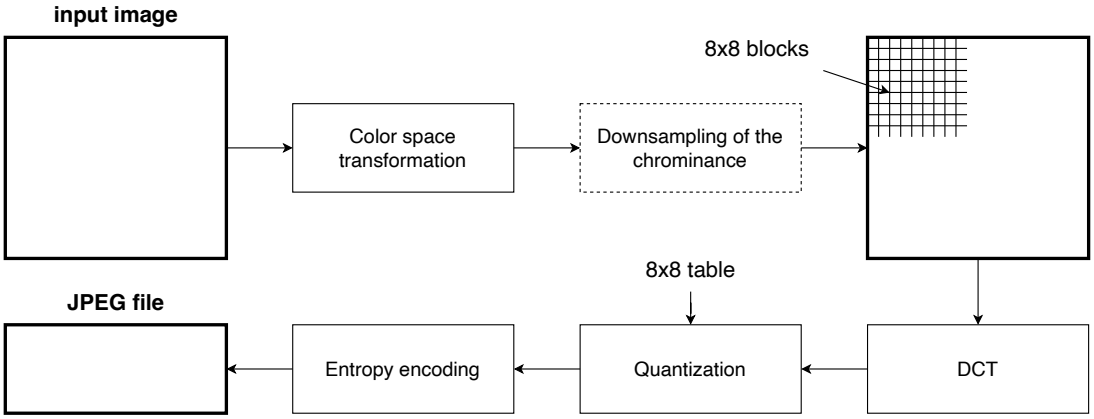



Figure 2: Workflow of the JPEG compression algorithm.

The JPEG compression algorithm consists of the following steps, as illustrated in Figure 2:

1. **Color space transformation**: The input image is first converted from RGB to $\mathrm{YC_bC_r}$ color space.

2. **Chroma downsampling**: Downsampling is applied to the chrominance components $C_b$ and $C_r$ to reduce their resolution. Common downsampling ratios include 4:4:4 (no downsampling), 4:2:2 (horizontal reduction by a factor of 2), and most commonly 4:2:0 (reduction by a factor of 2 in both horizontal and vertical directions).

3. **Block-wise DCT**: Each component layer is divided into $8 \times 8$ blocks. The DCT transform $\mathbf{D} : \mathbb{R}^{8\times8} \to \mathbb{R}^{8\times8}$ is applied to each block $u^{8\times8}$ to obtain the DCT coefficients

$$\mathbf{D}(u^{8\times8})_{i,j} = \frac{1}{4} T_i T_j \sum_{x=0}^{7}\sum_{y=0}^{7} u_{x,y}^{8\times8} \cos\left[\frac{(2x+1)i\pi}{16}\right] \cos\left[\frac{(2y+1)j\pi}{16}\right], \tag{1}$$

where the normalization factors are $T_k = \begin{cases} \frac{1}{\sqrt{2}}, & \text{if } k = 0, \\ 1, & \text{if } k \neq 0. \end{cases}$

4. **Quantization**: Each DCT block $x^{8\times8} = \mathbf{D}u^{8\times8}$ is quantized by dividing each coefficient by the corresponding element of a quantization table $q \in \mathbb{R}_+^{8\times8}$ and rounding to the nearest integer

$$\mathcal{Q}(x^{8\times8})_{i,j} = \text{round}\big(\frac{x_{i,j}^{8\times8}}{q_{ij}}\big), \tag{2}$$

where $\mathcal{Q}$ denotes the quantization operation.

5. **Entropy encoding**: The quantized coefficients are entropy encoded using lossless Huffman coding or arithmetic coding to further reduce the file size.

## 3.2 Standard JPEG Decompression

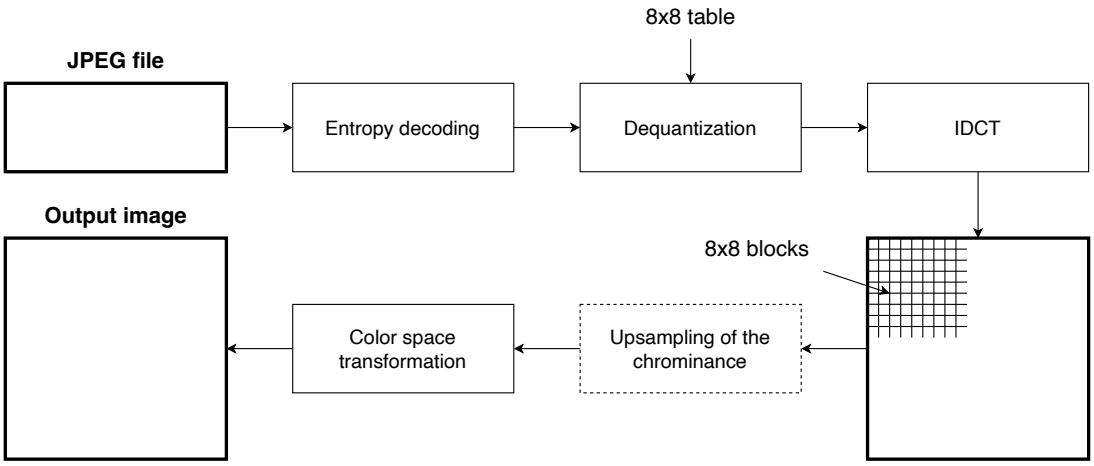



Figure 3: Workflow of the standard JPEG decompression algorithm.

The JPEG standard decompression converts a JPEG file to an image by applying the inverse steps of the compression in the reverse order, as is illustrated in Figure 3:

1. **Entropy decoding**: The compressed JPEG file is first entropy decoded to recover the quantized DCT coefficients for each $8 \times 8$ block.

2. **Dequantization**: Each block of quantized DCT coefficients $\tilde{v}^{8\times 8}$ is multiplied by its corresponding value in the quantization table $q \in \mathbb{R}_+^{8\times 8}$ to obtain the dequantized DCT coefficients

$$\mathcal{D}(\tilde{v}^{8\times 8})_{i,j} = \tilde{v}_{i,j}^{8\times 8} \cdot q_{i,j}, \tag{3}$$

   where $\mathcal{D}$ is the dequantization operation.

3. **Block-wise IDCT**: The dequantized $8 \times 8$ block of DCT coefficients $\tilde{x}^{8\times 8} = \mathcal{D}\tilde{v}^{8\times 8}$ is transformed back to the spatial domain using the inverse DCT (IDCT)

$$\tilde{u}^{8\times 8} = \mathbf{D}^{-1}(\tilde{x}^{8\times 8}). \tag{4}$$

4. **Block merging and upsampling**: The reconstructed blocks are merged to form the full image. If chroma subsampling was used during compression, the chroma channels are upsampled to match the luminance resolution.

5. **Color space conversion**: The image is converted from the $\mathrm{YC_bC_r}$ color space back to RGB.

For simplicity, we will focus on the luminance channel where no downsampling is applied. Since the entropy coding and the color space conversion are both reversible, the resulting effect of JPEG compression and standard decompression steps on an underlying uncompressed image $u_0$ can be modeled by

$$\tilde{u} = \mathbf{D}^{-1}\,\mathcal{D}\,\mathcal{Q}\,\mathbf{D}(u_0), \tag{5}$$

where $\mathbf{D}$ and $\mathbf{D}^{-1}$ are the DCT and IDCT applied on $8 \times 8$ block of the whole image, $\mathcal{Q}$ is the quantization, $\mathcal{D}$ is the dequantization and $\tilde{u}$ is the reconstructed image.

It is important to note that the quantization step $\mathcal{Q}$ is not invertible, and the subsequent dequantization step $\mathcal{D}$ simply takes the center value of each quantization interval, resulting in DCT coefficients that contain quantization errors. These errors manifest as blocking and ringing artifacts in the reconstructed image, particularly at low JPEG quality settings where quantization is more aggressive. In the following section, we discuss how these artifacts can be mitigated by incorporating the quantization constraints into a regularized optimization framework that improves consistency between neighboring blocks.

# 4 JPEG Decompression with Artifact Suppression

The total variation (TV) functional, defined as

$$TV(u) := \int_\Omega |\nabla u|\, dx, \tag{6}$$

has been widely used as a regularization term in various image restoration tasks prior to the deep learning era, as it promotes piecewise smooth regions while preserving sharp edges. Notable applications of the TV regularizer include inpainting [22, 19, 9, 23, 14], denoising [7, 3, 28, 17, 12], deconvolution [24, 18], super-resolution [7, 25], and cartoon+texture decomposition [21]. Its use for suppressing blocking artifacts in JPEG decompression was first introduced in [2], where JPEG decompression is formulated as a constrained optimization problem.

## 4.1 Decompression with TV- and Dirichlet-minimization

In the simplified setting of compression and decompression in Equation (5), a JPEG file provides quantized DCT coefficients $\tilde{v} = \mathcal{Q}\,\mathbf{D}(u_0)$ for an unknown original image $u_0$. Instead of using the neutral dequantization $\mathcal{D}$ for standard decompression, we can formulate the decompression as an inverse problem that aims at recovering an image $u$ with a regularization term $J(u)$ while satisfying the quantization constraints imposed by the quantized DCT coefficients $\tilde{v}$

$$\tilde{u} = \arg\min_{u \in U} J(u), \tag{7}$$

where $U$ is the image space constrained by the quantization consistency

$$\begin{aligned} U = \{u \in \mathbb{R}^{H\times W} : \mathcal{Q}\,\mathbf{D}(u) = \tilde{v}\} = \{u \in \mathbb{R}^{H\times W} : \mathbf{D}(u)_{i,j} \in [Q^-_{i,j}, Q^+_{i,j}]\}, \\ Q^-_{i,j} = \tilde{v}_{i,j} - \frac{1}{2} q_{i\%8, j\%8}, \\ Q^+_{i,j} = \tilde{v}_{i,j} + \frac{1}{2} q_{i\%8, j\%8}. \end{aligned} \tag{8}$$

$Q^-_{i,j}$ and $Q^+_{i,j}$ are the end points of the quantization interval of each DCT coefficient of the compressed image $\tilde{u}$. $q \in \mathbb{R}^{8\times 8}$ is the quantization table. This ensures the decompressed image to have the same quantized DCT coefficients if it is recompressed using the same quantization table.

In this paper, we mainly study three variants of the cost $J(u)$:

- Total variation: $J_{TV}(u) = \sum_{i,j\in\Omega} |\nabla u_{i,j}|$
- Adapted total variation in [2]: $J_{aTV}(u) = \sum_{i,j\in\Omega} |\alpha_{i,j} \nabla u_{i,j}|$ with $\alpha$ the adapted weight
- Dirichlet integral: $J_D(u) = \sum_{i,j\in\Omega} |\nabla u_{i,j}|^2$

where $\Omega$ is the image domain.

The norm of the gradient is realized in two different ways. In the total variation $J_{TV}$ and Dirichlet integral $J_D$, $|\nabla u_{i,j}|$ is defined as

$$|\nabla u_{i,j}| = \sqrt{(u_{i+1,j} - u_{i,j})^2 + (u_{i,j+1} - u_{i,j})^2 + \epsilon}, \tag{9}$$

where $\epsilon = 10^{-8}$ is a small positive number for the sake of numerical stability. The adapted total variation $J_{aTV}(u)$ is defined to be coherent to the description in [2] as

$$J_{aTV}(u) = \sum_{i,j\in\Omega} \sqrt{a^2_{i,j} + b^2_{i,j} + c^2_{i,j} + d^2_{i,j} + \epsilon}, \tag{10}$$

with

$$\begin{aligned} a_{i,j} &= \alpha_{i\%8}\,(u_{i+1,j} - u_{i,j}), \\ b_{i,j} &= \alpha_{j\%8}\,(u_{i,j+1} - u_{i,j}), \\ c_{i,j} &= \alpha_{(i-1)\%8}\,(u_{i,j} - u_{i-1,j}), \\ d_{i,j} &= \alpha_{(j-1)\%8}\,(u_{i,j} - u_{i,j-1}). \end{aligned} \tag{11}$$

In this way, the adapted weights $\alpha_n$ set larger coefficients for the borders of a $8\times 8$ block and smaller coefficients for the middle of the block to impose larger penalties on blocking artifacts near the block borders. The values of $\alpha_n$ are given in Table 2.

| n | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|---|---|
| $\alpha_n$ | 5.0 | 2.0 | 1.0 | 1.0 | 1.0 | 2.0 | 5.0 | 7.0 |

Table 2: Weights $\alpha_n$ for the adapted TV.

## 4.2 Solving the Constrained Minimization Problem

The minimization problem in Equation (7) is about a convex objective $J$ on the convex set $U$. In [2] this is solved by the classic subgradient descent algorithm with a non-expansive projection

$$u^{k+1} = P(u^k - t_k\, g(u^k)), \tag{12}$$

where $P : \mathbb{R}^{H\times W} \to U$ is a non-expansive[2] projection on the convex closed set $U$, $t_k$ is the step size, and $g = \nabla J$ is the subgradient of $J$. It can be proved [4] that if the step size satisfies $\sum_{k=0}^{\infty} t_k^2 < \infty$ and $\sum_{k=0}^{\infty} t_k = \infty$, then the lowest cost $J_{best}^{(k)} \triangleq \min_{i=0,\dots,k} J(u^i)$ converges to the optimal value $J^* = \min_{u\in U} J(u)$. In practice, the step size is set to $t_k = \beta/(k+1)$ with $\beta$ a positive parameter.

*Proof.* Assume the norm of subgradients is bounded, *i.e.* there is a $G$ such that $||g(u^k)||_2 \le G$. Let $\tilde{u} \in U$ be an optimal point that minimizes $J$ on $U$. Assume there is a $R$ such that $||u^0 - \tilde{u}||_2 \le R$. We have

$$\begin{aligned}
||u^{k+1} - \tilde{u}||_2^2 &= ||P(u^k - t_k\, g(u^k)) - \tilde{u}||_2^2 \\
&\le ||u^k - t_k\, g(u^k) - \tilde{u}||_2^2 \\
&\text{(because projection } P \text{ satisfies } \forall(x,y),\ ||P(x) - P(y)||_2 \le ||x-y||_2) \\
&= ||u^k - \tilde{u}||_2^2 + t_k^2||g(u^k)||_2^2 - 2t_k g(u^k)^{\mathrm{T}}(u^k - \tilde{u}) \\
&\le ||u^k - \tilde{u}||_2^2 + t_k^2||g(u^k)||_2^2 - 2t_k(J(u^k) - J(\tilde{u})) \\
&\text{(because the subgradient } g \text{ of the convex function } J \text{ satisfies } J(\tilde{u}) \ge J(u^k) + g(u^k)^{\mathrm{T}}(\tilde{u} - u^k)).
\end{aligned} \tag{13}$$

By recursively applying the above inequality, we have

$$||u^{k+1} - \tilde{u}||_2^2 \le ||u^0 - \tilde{u}||_2^2 + \sum_{i=0}^{k} t_i^2||g(u^i)||_2^2 - 2\sum_{i=0}^{k} t_i(J(u^i) - J(\tilde{u})). \tag{14}$$

Using $||u^0 - \tilde{u}||_2 \le R$ and $||g(u^i)||_2 \le G$, then we have

$$\begin{aligned}
2\sum_{i=0}^{k} t_i(J(u^i) - J(\tilde{u})) &\le ||u^0 - \tilde{u}||_2^2 + \sum_{i=0}^{k} t_i^2||g(u^i)||_2^2 - ||u^{i+1} - \tilde{u}||_2^2 \\
&\le R^2 + \sum_{i=0}^{k} t_i^2||g(u^i)||_2^2 \\
&\le R^2 + G^2\sum_{i=0}^{k} t_i^2.
\end{aligned} \tag{15}$$

Combining this with

$$\sum_{i=0}^{k} t_i(J(u^i) - J(\tilde{u})) \ge \left(\sum_{i=0}^{k} t_i\right)\left(\min_{i=0,\dots,k} J(u^i) - J(\tilde{u})\right) = \left(\sum_{i=0}^{k} t_i\right)\left(J_{best}^{(k)} - J(\tilde{u})\right), \tag{16}$$

[2] A *non-expansive* mapping $P$ satisfies $\forall(x,y) \in \mathbb{R}^{H\times W} \times \mathbb{R}^{H\times W}, ||P(x) - P(y)|| \le ||x - y||$.

we obtain the inequality

$$J_{best}^{(k)} - J(\tilde{u}) \leq \frac{R^2 + G^2 \sum_{i=0}^{k} t_k^2}{2 \sum_{i=0}^{k} t_k}, \tag{17}$$

which converges to 0 when the step size is set to satisfy $\sum_{k=0}^{\infty} t_k^2 < \infty$ and $\sum_{k=0}^{\infty} t_k = \infty$. □

**Non-expansive projection.** The above algorithm also requires a non-expansive projection $P$ on the convex set $U$ defined as $U = \mathbf{D}^{-1}(V)$ where $V = \{v \in \mathbb{R}^{H \times W} : v_{i,j} \in [Q_{i,j}^-, Q_{i,j}^+]\}$ is a hypercube in the DCT coefficient domain. Since $V$ is a convex closed set and $\mathbf{D}$ is a linear transform, the set $U$ is also a convex closed set. The non-expansive projection $P : \mathbb{R}^{H \times W} \to U$ is defined by

$$P(u) := \mathbf{D}^{-1} \circ \pi_V \circ \mathbf{D}(u), \tag{18}$$

where $\pi_V$ is an orthogonal projection function on $V$ that clips each value to its corresponding interval

$$\forall v \in \mathbb{R}^{H \times W},\ \pi_V(v)_{i,j} = \min\left(\max\left(v_{i,j}, Q_{i,j}^-\right), Q_{i,j}^+\right). \tag{19}$$

It can be proved that $P$ is non-expansive: $\forall x, y \in \mathbb{R}^{H \times W}$,

$$\begin{aligned} ||P(x) - P(y)||_2^2 &= ||\mathbf{D}^{-1} \circ \pi_V \circ \mathbf{D}(x) - \mathbf{D}^{-1} \circ \pi_V \circ \mathbf{D}(y)||_2^2 \\ &= ||\mathbf{D}^{-1} \left(\pi_V \circ \mathbf{D}(x) - \pi_V \circ \mathbf{D}(y)\right)||_2^2 \\ &= ||\pi_V \circ \mathbf{D}(x) - \pi_V \circ \mathbf{D}(y)||_2^2 \\ &\leq ||\mathbf{D}(x) - \mathbf{D}(y)||_2^2 = ||x - y||_2^2. \end{aligned} \tag{20}$$

The second equality is based on the linearity of the inverse DCT. The third and the last equalities hold as the DCT and inverse DCT are both orthonormal transforms. The inequality is based on the non-expansiveness of $\pi_V$.

## 4.3 Implementation

It is worth noting that in [2] the authors take the last iterative point $u^k$ as the optimal restored image, yet one should take the iterative point at the lowest cost $u_{best}^{(k)} \triangleq \arg\min_{u^i \in \{u^0, \ldots, u^k\}} J(u^i)$ according to the previous proof of the convergence of the constrained minimization.

In addition, the decreasing step size is set to $t_k = \beta/k$ with an adjustable parameter $\beta$, in contrast to $t_k = 1/k$ in [2] which might end up in over-optimization. This is because the optimal restored image does not correspond to the minimum TV cost but to an intermediate cost during the iterations. Indeed, the TV only provides an approximate prior that preserves edges whilst smoothing away artifacts in flat regions, yet it can also smooth away oscillating textures when the minimization problem is over-optimized. For this reason, here we allow the control of the step size with a parameter named $\beta$ to avoid the over-optimization from the first iteration. The used default values of $\beta$ are shown in Table 3.

| cost function | $J_{TV}$ | $J_{aTV}$ | $J_D$ |
|---|---|---|---|
| $\beta$ | 0.8 | 0.1 | 0.01 |

Table 3: Default values of the step size $\beta$ for different cost functions.

The pseudo-code of our improved algorithm is described in Algorithm 1.

**Algorithm 1:** JPEG decompression with TV-minimization for artifact suppression

**Input** $\tilde{v} \in \mathbb{R}^{H \times W}$**:** quantized DCT coefficients
**Input** $q \in \mathbb{R}^{8 \times 8}$**:** quantization table
**Param** $J$**:** cost function
**Param** $K$**:** number of iterations
**Param** $\beta \in \mathbb{R}_*^+$**:** step size control
**Output** $\tilde{u} \in \mathbb{R}^{H \times W}$**:** decompressed image

$u = \mathbf{DCT}_{8\times 8}(\tilde{v})$
$\tilde{u} = u$
$J^* = J(u)$
**for** $k$ **from** *1* **to** $K$ **do**
  $t_k = \beta / k$
  $u = u - t_k \cdot \nabla_u J(u)$
  $v = \mathbf{DCT}_{8\times 8}(u)$
  **for** $(i, j) \in [0, \ldots, H-1] \times [0, \ldots, W-1]$ **do**
    $v_{i,j} = \min\left(\max\left(v_{i,j}, \tilde{v}_{i,j} - \frac{1}{2} q_{i\%8, j\%8}\right), \tilde{v}_{i,j} + \frac{1}{2} q_{i\%8, j\%8}\right)$
  $u = \mathbf{IDCT}_{8\times 8}(v)$
  **if** $J(u) < J^*$ **then**
    $\tilde{u} = u$
    $J^* = J(u)$
**return** $\tilde{u}$

# 5 Experiments

In this section, we evaluate the JPEG decompression method based on the minimization of the three cost functions: the standard total variation cost $J_{TV}$, the adapted total variation cost $J_{aTV}$, and the Dirichlet integral cost $J_D$. Both quantitative and qualitative evaluations are conducted.

## 5.1 Quantitative Evaluation on SIDD and RAISE Datasets

The three variants of decompression algorithm were quantitatively evaluated on two datasets: SIDD [1] and RAISE [11]. Specifically, 160 noiseless sRGB images in PNG format from the SIDD dataset, and 200 natural sRGB images in TIFF format from the RAISE dataset were selected for the evaluation. Such setting took in consideration that the hypothesis of piecewise smoothness is closer to the real modality of contents on noiseless images but could be violated by the presence of noise on natural images.

The average image quality improvements on the SIDD dataset measured by PSNR and SSIM are shown in Table 4. Different JPEG qualities from 50 to 90 were considered, and the number of iterations $K$ was set to 5. We can see at high quality compressions (e.g. $Q = 90$ or 80) that the Dirichlet cost function $J_D$ works better than the other variants, and the TV regularizer $J_{TV}$ performs best on severely compressed images and consistently outperforms $J_{aTV}$. It is worth noting that at $Q = 90$ the standard decoding is better than $J_{TV}$ and $J_{aTV}$, and the improvement brought by $J_D$ is small. This suggests that employing the minimization-based decompression method may be unnecessary as it might over-smooth an image.

The same evaluation was carried out on the RAISE dataset, with the average PSNRs and SSIMs shown in Table 5. It is seen that the TV cost function $J_{TV}$ achieves the best PSNR scores for all the compression qualities, suggesting that $J_{TV}$ is better at reducing the JPEG artifacts at pixel level.

Nevertheless, the Dirichlet integral cost function $J_D$ achieves the best SSIM scores for compression qualities $Q$ from 60 to 90, which hints that $J_D$ is suitable for preserving structural contents.

| PSNR ↑ / SSIM ↑ | Q=90 | Q=80 | Q=70 | Q=60 | Q=50 |
|---|---|---|---|---|---|
| standard | 47.6 / 0.989 | 46.0 / **0.984** | 45.0 / 0.979 | 44.1 / 0.975 | 43.4 / 0.972 |
| $J_{TV}$ | 47.5 / 0.988 | 46.1 / **0.984** | **45.3** / **.980** | **44.6** / **.978** | **44.1** / **.975** |
| $J_{aTV}$ | 47.3 / 0.988 | 46.0 / 0.983 | 45.2 / **.980** | **44.6** / 0.977 | 43.9 / **.975** |
| $J_D$ | **47.8** / **.990** | **46.2** / **.984** | 45.2 / **.980** | 44.3 / 0.976 | 43.6 / 0.973 |

Table 4: The average PSNR and SSIM values of the three cost functions evaluated on images of the SIDD dataset compressed by JPEG at different qualities. The number of iteration steps was set to 5. The row *standard* shows the quality of the image decompressed by the standard JPEG coding.

| PSNR ↑ / SSIM ↑ | Q=90 | Q=80 | Q=70 | Q=60 | Q=50 |
|---|---|---|---|---|---|
| standard | 40.0 / 0.973 | 36.8 / 0.953 | 35.3 / 0.939 | 34.3 / 0.927 | 33.5 / 0.917 |
| $J_{TV}$ | **40.3** / 0.973 | **37.3** / 0.953 | **35.8** / 0.940 | **34.7** / **.929** | **34.0** / **.921** |
| $J_{aTV}$ | 40.2 / 0.973 | 37.2 / 0.953 | 35.6 / 0.939 | 34.6 / **.929** | 33.9 / 0.920 |
| $J_D$ | 40.2 / **.975** | 37.1 / **.955** | 35.6 / **.941** | 34.6 / **.929** | 33.8 / 0.920 |

Table 5: The average PSNR and SSIM values of the three cost functions evaluated on images of the RAISE dataset compressed by JPEG at different qualities. The number of iteration steps was set to 5. The row *standard* shows the quality of the image decompressed by the standard JPEG coding.

When comparing the results on noiseless images of the SIDD dataset and on noisy images of the RAISE dataset, it is interesting to notice that the inconsistency between the PSNR and SSIM is present only on the natural images of the RAISE dataset, but not on the noiseless images of the SIDD dataset. This could be explained by the possibility that the TV cost function degrades the image structures when being applied to images with noise. As the hypothesis of piecewise smoothness is no longer valid on noisy images but the TV-minimization tends to reconstruct piecewise smooth areas, the final decompressed image may contain undesirable smooth structures in which noise accumulates into blobs. On the other hand, the Dirichlet integral cost function $J_D$ imposes weaker artifact suppression and does not create piecewise smooth areas. Thus there are fewer unwanted smooth structures decreasing the SSIM score on decompressed images using $J_D$.

We further evaluated the quality of the decompressed images in the RAISE dataset as a function of the number of iterations. Figure 4 shows the variation of the average PSNR and SSIM with respect to the number of iterations for different JPEG compression qualities from 50 to 90. The TV cost function $J_{TV}$ generally performs better than the other two cost functions when evaluated using PSNR. However, we observe the over-optimization problem that leads to the degraded image qualities as the number of iteration increases. For very weak JPEG compression (e.g. $Q = 90$) the decompression quality of the minimization-based algorithms could drop to a level below the decompression quality of the standard JPEG decoding. The same over-optimization problem occurs for the SSIM metric. It can be seen that the SSIM scores obtained using $J_{TV}$ and $J_{aTV}$ improve during the first few iterations and then drop dramatically. On the other hand, the decompression algorithm using $J_D$ suffers less from over-optimization when evaluated using SSIM and tends toward a steady SSIM value as the number of iterations increases, suggesting that $J_D$ better preserves structural content.

## 5.2 Case Study

We conducted qualitative experiments on three representative images, `traffic`, `text` and `eye`, to compare the performance of the three decompression algorithms in different cases. `traffic` is a noiseless image, and `text` and `eye` are images with camera noise.

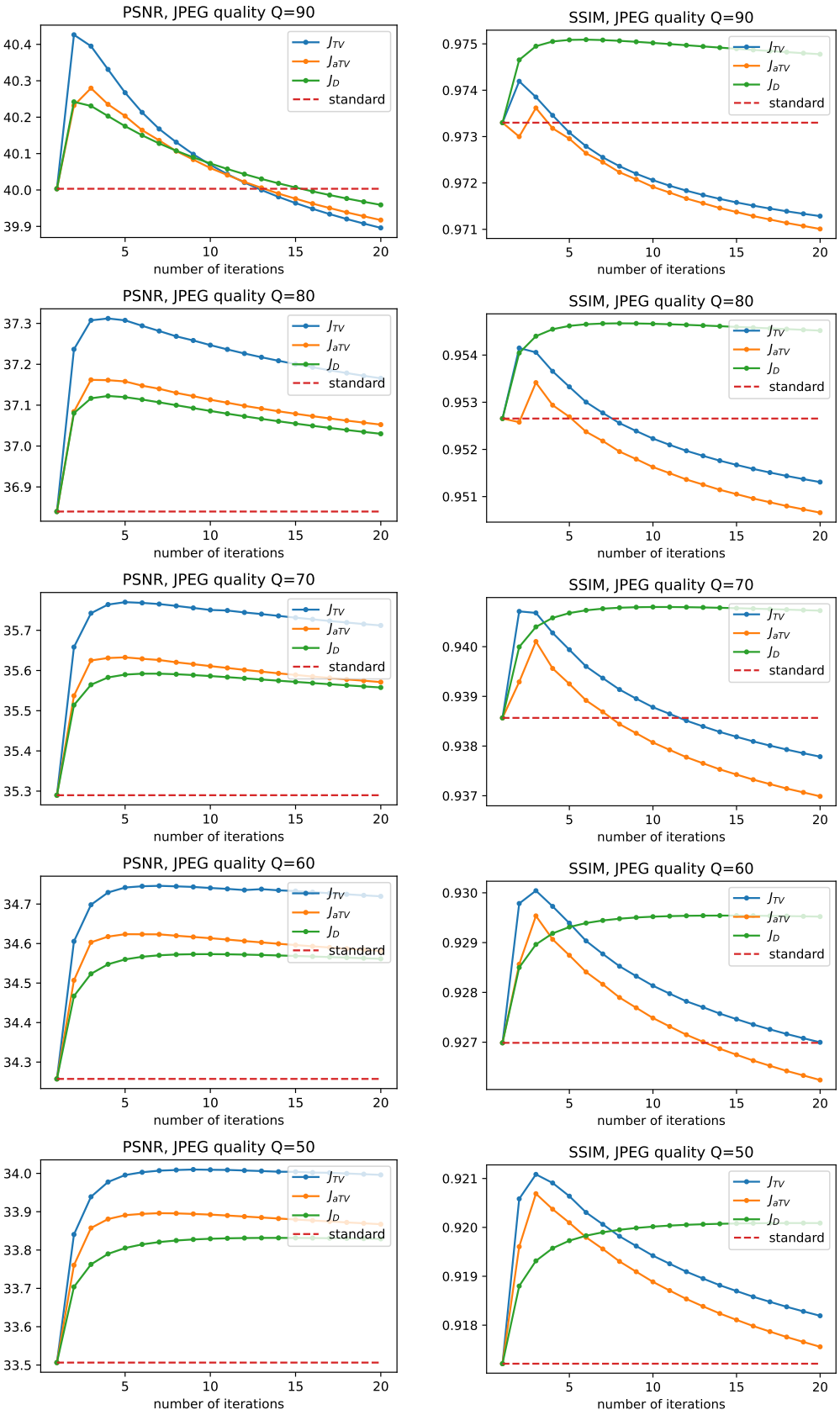
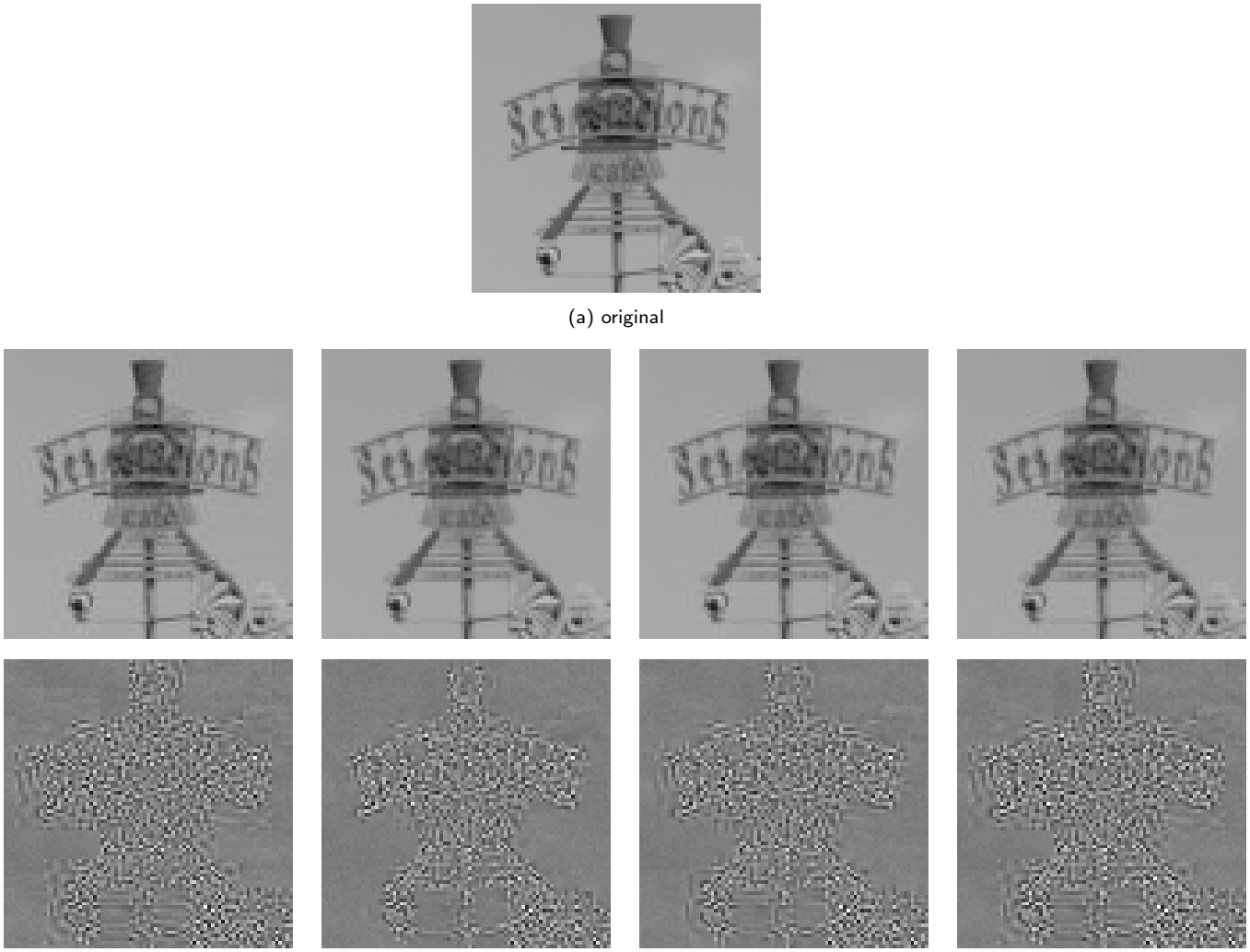



Figure 4: The variation of the average PSNR and SSIM of the decompressed images in the RAISE dataset with respect to the number of iterations using different decompression algorithms and for different JPEG compression qualities from 50 to 90. $J_{TV}$ outperforms the other two costs when evaluated using PSNR, but is prone to over-optimizing the decompressed image, decreasing both PSNR and SSIM values after a certain number of iterations. $J_D$ also suffers from over-optimization when evaluated using PSNR but is robust to the number of iterations when evaluated using SSIM, suggesting the advantage of using $J_D$ for preserving structural content during the optimization.

The first experiment was performed on the noiseless image `traffic` compressed by JPEG at quality $Q = 80$. The number of iterations was set to 5. The comparison results are shown in Figure 5. We can see the TV cost function $J_{TV}$ achieves the best PSNR and SSIM score, and the ringing artifacts near the signboard are effectively eliminated by the $J_{TV}$-decompression algorithm compared to the decompression algorithms using $J_{aTV}$ or $J_D$ costs.



(a) original

(b) standard decompression PSNR=35.3, SSIM=0.968

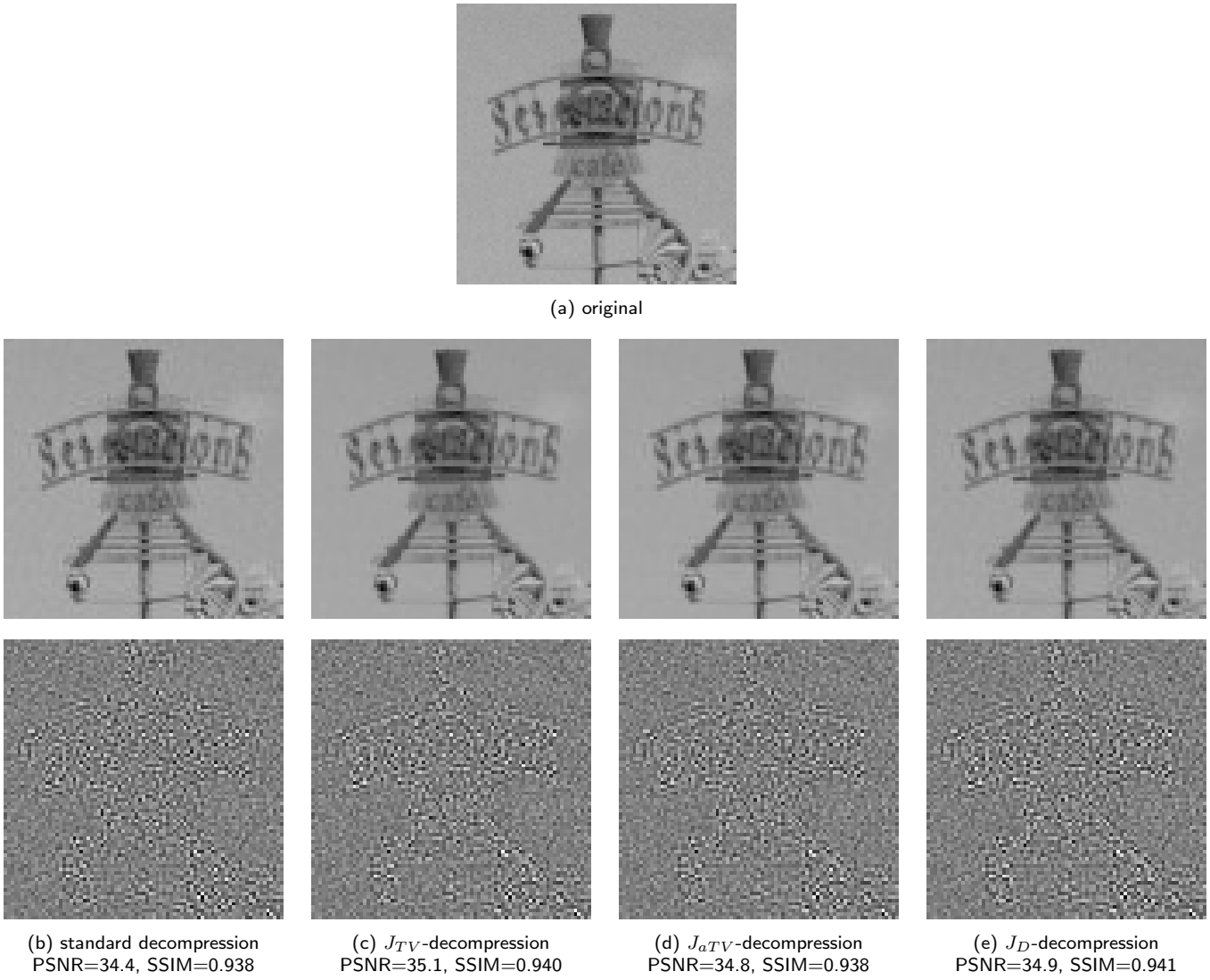
(c) $J_{TV}$-decompression PSNR=36.1, SSIM=0.970

(d) $J_{aTV}$-decompression PSNR=35.9, SSIM=0.968

(e) $J_D$-decompression PSNR=35.8, SSIM=0.967

Figure 5: Comparison of different decompression methods on the noiseless `traffic` image compressed by JPEG at quality $Q = 80$. Top: original image before compression; middle: decompressed image using different methods; bottom: the difference between the decompressed image and the original image. The number of iterations is set to 5. The TV cost function $J_{TV}$ achieves the best PSNR and SSIM score. The ringing artifacts near the signboard are effectively eliminated by the $J_{TV}$-decompression algorithm compared to the decompression algorithms using $J_{aTV}$ or $J_D$ costs.

The second experiment was performed on the same uncompressed `traffic` image with additive Gaussian noise of $\sigma = 3$. Likewise, the image was JPEG-compressed at quality $Q = 80$, and the number of iterations was set to 5. The comparison results in Figure 6 show that the $J_{TV}$-decompression method achieves the best PSNR score, while the $J_D$-decompression method achieves the best SSIM score. We can visually see that the $J_{TV}$-decompression method not only reduces JPEG artifacts but also the original noise of the image. The background sky is smoother than in the uncompressed image, yet some new weak structures are present in the sky. The reason is that the original image does not follow the hypothesis of piecewise smoothness but the $J_{TV}$-decompression method attempts to find piecewise smooth structures within the quantization constraints, resulting



(a) original

(b) standard decompression PSNR=34.4, SSIM=0.938

(c) $J_{TV}$-decompression PSNR=35.1, SSIM=0.940

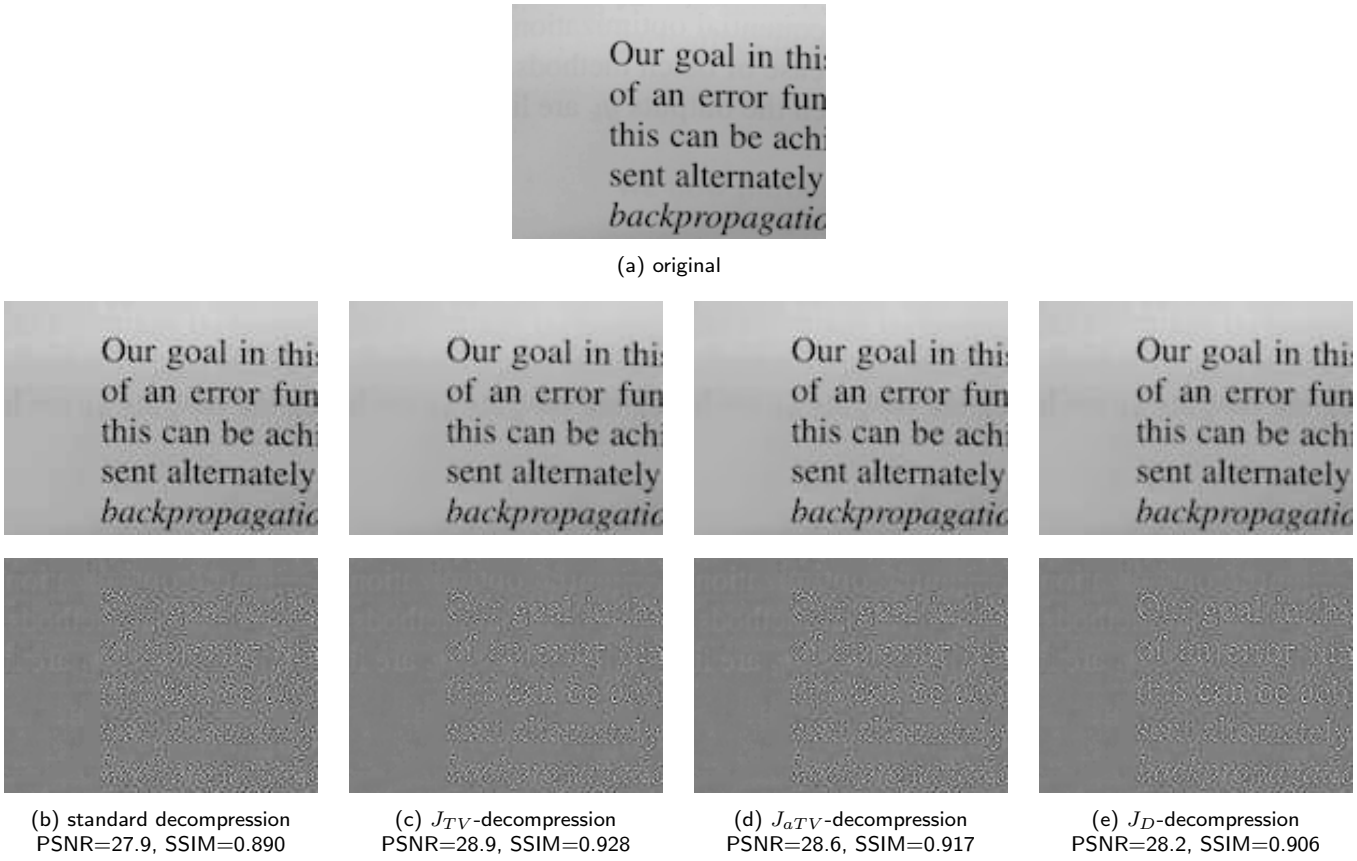
(d) $J_{aTV}$-decompression PSNR=34.8, SSIM=0.938

(e) $J_D$-decompression PSNR=34.9, SSIM=0.941

Figure 6: Comparison of different decompression methods on the `traffic` image with additive Gaussian noise of $\sigma = 3$ compressed by JPEG at quality $Q = 80$. Top: image before compression; middle: decompressed image using different methods; bottom: the difference between the decompressed image and the original image. The number of iterations is set to 5. $J_{TV}$ achieves the best PSNR and $J_D$ achieves the best SSIM score. The $J_{TV}$-decompression method not only reduces the JPEG artifacts but also the original noise of the image. The background sky is smoother, yet new weak structures appear in the sky which do not affect the PSNR score but decreases the SSIM score. By contrast, the $J_D$-decompression method achieves the best SSIM value simply because it performs weaker artifact suppression and does not introduce new structures, but its decompressed image still contains many artifacts.

in structures with weaker noise but distorted content. The reduced noise with altered structure does not harm the PSNR score, but is captured by SSIM which compares the structural similarity of two images. As a result, the PSNR of the decompressed image using $J_{TV}$ is high while its SSIM is low. By contrast, the $J_D$-decompression method achieves the best SSIM value simply because it performs weaker artifact suppression, including both JPEG artifacts and original noise, and does not introduce new structures that degrade the SSIM value. Its decompressed image still contains many artifacts.

The third experiment was performed on the `text` image which was compressed by JPEG at quality $Q = 50$. The number of iterations was set to 15. The comparison results are visualized in Figure 7, where the TV cost $J_{TV}$ achieves the best restoration quality in terms of both PSNR and SSIM. As the compressed image is of very low quality and the text can be approximately considered piecewise smooth, the image prior underlying the TV cost function is suitable for restoring the original content and yields a large quality improvement. We can see that the blocking artifacts in the background are



(a) original

(b) standard decompression
PSNR=27.9, SSIM=0.890

(c) $J_{TV}$-decompression
PSNR=28.9, SSIM=0.928

(d) $J_{aTV}$-decompression
PSNR=28.6, SSIM=0.917

(e) $J_D$-decompression
PSNR=28.2, SSIM=0.906

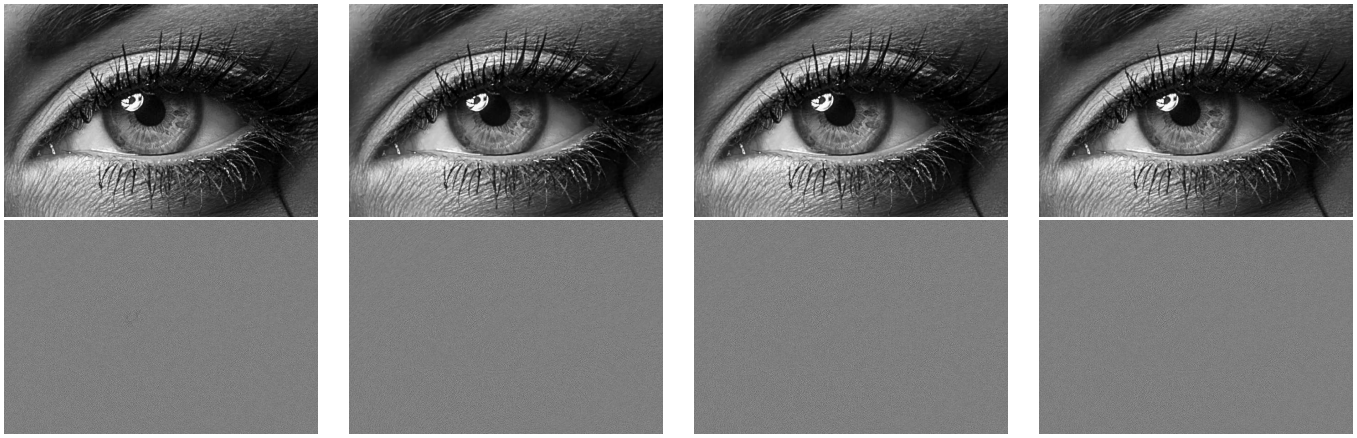
Figure 7: Comparison of different decompression methods on the `text` image compressed by JPEG at quality $Q = 50$. Top: image before compression; middle: decompressed image using different methods; bottom: the difference between the decompressed image and the original image. The number of iterations was set to 15. Blocking artifacts in the background are perfectly suppressed, and the ringing artifacts near the text, especially in the spaces between lines, are reduced.

perfectly suppressed, and the ringing artifacts near the text, especially in the spaces between lines, are reduced.

The fourth experiment was carried out on the `eye` image, which contains highly textured content. The image was JPEG-compressed at quality $Q = 80$, and the number of iterations was set to 5. The decompressed images using different methods are shown in Figure 8, where we can see that the improvements in both visual quality and the PSNR and SSIM scores are limited. For a textured image, the hypothesis of piecewise smoothness does not hold, and the resulting decompressed images do not benefit much from any of the cost functions.

The last experiment was performed on the same `eye` image, with JPEG compression at a higher quality $Q = 90$ and the number of iterations set to 15. This example was designed to showcase the performance of the decompression algorithm in an extreme unfavorable setting: the image is full of textures, the JPEG quality is high, and the decompression method over-optimizes the image after a certain number of iterations. It can be seen in Figure 9 that only $J_D$-decompression slightly improves the image quality, and using either $J_{TV}$ or $J_{aTV}$ degrades the image quality. When zooming in we can see that the delicate textures in the eyebrows are smoothed out when using $J_{TV}$ or $J_{aTV}$ for decompression.

(a) original



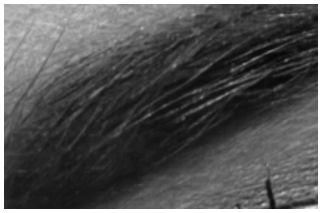
(b) standard decompression PSNR=36.4, SSIM=0.933

(c) $J_{TV}$-decompression PSNR=36.7, SSIM=0.934

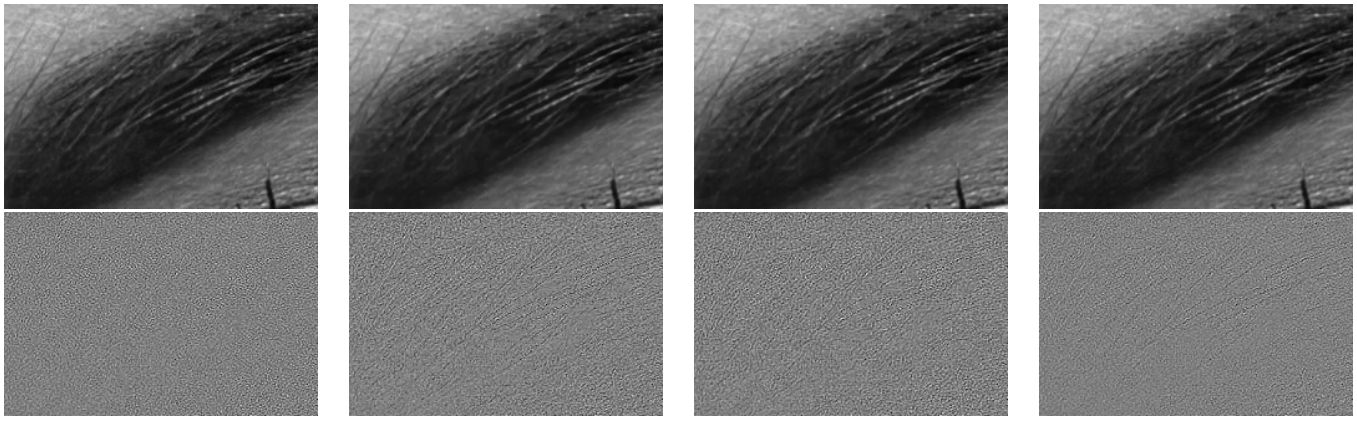
(d) $J_{aTV}$-decompression PSNR=36.8, SSIM=0.936

(e) $J_D$-decompression PSNR=36.7, SSIM=0.936

Figure 8: Comparison of different decompression methods on an image with highly textured content compressed by JPEG at quality $Q = 80$. Top: image before compression; middle: decompressed image using different methods; bottom: the difference between the decompressed image and the original image. The number of iterations is set to 5. The improvements in both visual quality and the PSNR and SSIM metrics are limited. For a textured image the hypothesis of piecewise smoothness does not hold and the resulting decompressed images do not benefit much from any of the cost functions.



(a) original



(b) standard decompression PSNR=42.1, SSIM=0.978

(c) $J_{TV}$-decompression PSNR=41.7, SSIM=0.975

(d) $J_{aTV}$-decompression PSNR=41.8, SSIM=0.976

(e) $J_D$-decompression PSNR=42.4, SSIM=0.980

Figure 9: Comparison of different decompression methods on an image full of textured content compressed by JPEG at quality $Q = 90$. Top: image before compression; middle: decompressed image using different methods; bottom: the difference between the decompressed image and the original image. The number of iterations is set to 15. Only the $J_D$-decompression method slightly boosts the image quality, whereas using either $J_{TV}$ or $J_{aTV}$ degrades it. When zooming in we can see that the delicate textures in the eyebrows are smoothed out when using $J_{TV}$ or $J_{aTV}$ for decompression.

# 6 Conclusion

In this paper we briefly revisited the JPEG decompression method based on the minimization of the standard TV cost, the adapted TV cost and the Dirichlet integral cost in order to reduce the blocking artifacts inherent to the JPEG compression scheme itself. The optimization method based on gradient descent with projection onto a convex set was reviewed both theoretically and algorithmically. Both quantitative and qualitative experiments were conducted to show the effective artifact removal achieved by the decompression methods and the over-optimization problem that smooths out original textures. In general, the standard TV cost results in slightly better performance in terms of PSNR than the other two cost functions, while the Dirichlet integral cost has a weaker impact on removing artifacts but yields higher SSIM scores. The impact of the number of iterations on the decompression performance was also studied qualitatively for both noiseless and noisy images at different compression qualities to serve as a reference for choosing the number of iterations that achieves the best image quality.

# Image Credits

 by M. Colom, CC-BY

 text

 eye

# References

[1] A. Abdelhamed, S. Lin, and M. S. Brown, *A High-Quality Denoising Dataset for Smartphone Cameras*, in IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2018. https://doi.org/10.1109/CVPR.2018.00182.

[2] F. Alter, S. Durand, and J. Froment, *Adapted Total Variation for Artifact Free Decompression of JPEG Images*, Journal of Mathematical Imaging and Vision, 23 (2005), pp. 199–211. https://doi.org/10.1007/s10851-005-6467-9.

[3] A. Beck and M. Teboulle, *Fast Gradient-Based Algorithms for Constrained Total Variation Image Denoising and Deblurring Problems*, IEEE Transactions on Image Processing, 18 (2009), pp. 2419–2434. https://doi.org/10.1109/TIP.2009.2028250.

[4] S. Boyd, L. Xiao, and A. Mutapcic, *Subgradient Methods*. Lecture notes of EE392o, Stanford University, Autumn Quarter, 2003. https://web.stanford.edu/class/ee392o/subgrad_method.pdf.

[5] K. Bredies and M. Holler, *A Total Variation--Based JPEG Decompression Model*, SIAM Journal on Imaging Sciences, 5 (2012), pp. 366–393. https://doi.org/10.1137/110833531.

[6] ——, *Artifact-Free Decompression and Zooming of JPEG Compressed Images with Total Generalized Variation*, in International Joint Conference on Computer Vision, Imaging and Computer Graphics. Theory and Application (VISIGRAPP), Springer, 2013, pp. 242–258. https://doi.org/10.1007/978-3-642-38241-3_16.

[7] A. Chambolle, *An Algorithm for Total Variation Minimization and Applications*, Journal of Mathematical Imaging and Vision, 20 (2004), pp. 89–97. https://doi.org/10.1023/B:JMIV.0000011325.36760.1e.

[8] A. Chambolle and T. Pock, *A First-Order Primal-Dual Algorithm for Convex Problems with Applications to Imaging*, Journal of Mathematical Imaging and Vision, 40 (2011), pp. 120–145. https://doi.org/10.1007/s10851-010-0251-1.

[9] T. F. Chan, J. Shen, and H.-M. Zhou, *Total Variation Wavelet Inpainting*, Journal of Mathematical Imaging and Vision, 25 (2006), pp. 107–125. https://doi.org/10.1007/s10851-006-5257-3.

[10] H. Chang, M. K. Ng, and T. Zeng, *Reducing Artifacts in JPEG Decompression Via a Learned Dictionary*, IEEE Transactions on Signal Processing, 62 (2013), pp. 718–728. https://doi.org/10.1109/TSP.2013.2290508.

[11] D.-T. Dang-Nguyen, C. Pasquini, V. Conotter, and G. Boato, *Raise: A Raw Images Dataset for Digital Image Forensics*, in ACM Multimedia Systems Conference, 2015, pp. 219–224. https://doi.org/10.1145/2713168.2713194.

[12] J. Duran, B. Coll, and C. Sbert, *Chambolle's Projection Algorithm for Total Variation Denoising*, Image Processing On Line, 3 (2013), pp. 311–331. https://doi.org/10.5201/ipol.2013.61.

[13] M. Ehrlich, L. Davis, S.-N. Lim, and A. Shrivastava, *Quantization Guided JPEG Artifact Correction*, in European Conference on Computer Vision (ECCV), Springer, 2020, pp. 293–309. https://link.springer.com/chapter/10.1007/978-3-030-58598-3_18.

[14] V. Fedorov, G. Facciolo, and P. Arias, *Variational Framework for Non-Local Inpainting*, Image Processing On Line, 5 (2015), pp. 362–386. https://doi.org/10.5201/ipol.2015.136.

[15] X. Fu, X. Wang, A. Liu, J. Han, and Z.-J. Zha, *Learning Dual Priors for JPEG Compression Artifacts Removal*, in IEEE/CVF International Conference on Computer Vision (ICCV), 2021, pp. 4086–4095. https://doi.org/10.1109/ICCV48922.2021.00405.

[16] X. Fu, Z.-J. Zha, F. Wu, X. Ding, and J. Paisley, *JPEG Artifacts Reduction Via Deep Convolutional Sparse Coding*, in IEEE/CVF International Conference on Computer Vision (ICCV), 2019, pp. 2501–2510. https://doi.org/10.1109/ICCV.2019.00259.

[17] P. Getreuer, *Rudin-Osher-Fatemi Total Variation Denoising Using Split Bregman*, Image Processing On Line, 2 (2012), pp. 74–95. https://doi.org/10.5201/ipol.2012.g-tvd.

[18] ——, *Total Variation Deconvolution Using Split Bregman*, Image Processing On Line, 2 (2012), pp. 158–174. https://doi.org/10.5201/ipol.2012.g-tvdc.

[19] ——, *Total Variation Inpainting Using Split Bregman*, Image Processing On Line, 2 (2012), pp. 147–157. https://doi.org/10.5201/ipol.2012.g-tvi.

[20] C. Jung, L. Jiao, H. Qi, and T. Sun, *Image Deblocking Via Sparse Representation*, Signal Processing: Image Communication, 27 (2012), pp. 663–677. https://doi.org/10.1016/j.image.2012.03.002.

[21] V. Le Guen, *Cartoon + Texture Image Decomposition by the TV-L1 Model*, Image Processing On Line, 4 (2014), pp. 204–219. https://doi.org/10.5201/ipol.2014.103.

[22] S. Masnou and J.-M. Morel, *Level Lines Based Disocclusion*, in International Conference on Image Processing (ICIP), IEEE, 1998, pp. 259–263. https://doi.org/10.1109/ICIP.1998.999016.

[23] K. Papafitsoros, C. B. Schoenlieb, and B. Sengul, *Combined First and Second Order Total Variation Inpainting Using Split Bregman*, Image Processing On Line, 3 (2013), pp. 112–136. https://doi.org/10.5201/ipol.2013.40.

[24] D. Perrone and P. Favaro, *Total Variation Blind Deconvolution: The Devil Is in the Details*, in IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 2909–2916. https://doi.org/10.1109/CVPR.2014.372.

[25] F. Shi, J. Cheng, L. Wang, P.-T. Yap, and D. Shen, *LRTV: MR Image Super-Resolution with Low-Rank and Total Variation Regularizations*, IEEE Transactions on Medical Imaging, 34 (2015), pp. 2459–2466. https://doi.org/10.1109/TMI.2015.2437894.

[26] M. Sun, X. He, S. Xiong, C. Ren, and X. Li, *Reduction of JPEG Compression Artifacts Based on DCT Coefficients Prediction*, Neurocomputing, 384 (2020), pp. 335–345. https://doi.org/10.1016/j.neucom.2019.12.015.

[27] Z. Xiong, M. T. Orchard, and Y.-Q. Zhang, *A Deblocking Algorithm for JPEG Compressed Images Using Overcomplete Wavelet Representations*, IEEE Transactions on Circuits and Systems for Video Technology, 7 (1997), pp. 433–437. https://doi.org/10.1109/76.564123.

[28] Q. Yuan, L. Zhang, and H. Shen, *Hyperspectral Image Denoising Employing a Spectral--Spatial Adaptive Total Variation Model*, IEEE Transactions on Geoscience and Remote Sensing, 50 (2012), pp. 3660–3677. https://doi.org/10.1109/TGRS.2012.2185054.