

Reducing Artifact in JPEG Decompression via a Learned Dictionary

Huibin Chang, Michael K. Ng, Tiejong Zeng

Abstract—The JPEG compression is among the most successful compression schemes since it readily provides good compressed results at a rather high compression ratio. However, the decompressed result of the standard JPEG decompression scheme usually contains some visible artifacts, such as blocking artifacts and Gibbs artifacts (ringing), especially when the compression ratio is rather high. In this paper, a novel artifact reducing approach for the JPEG decompression is proposed via sparse and redundant representations over a learned dictionary. Indeed, an effective two-step algorithm is developed. The first step involves dictionary learning and the second step involves the total variation regularization for decompressed images. Numerical experiments are performed to demonstrate that the proposed method outperforms the total variation and weighted total variation decompression methods in the measure of peak of signal to noise ratio, and structural similarity.

Index Terms—JPEG; Decompression; Total Variation; Learned Dictionary; Primal-dual algorithm

I. INTRODUCTION

THE JPEG method [1, 2] is one of the most popular lossy compression schemes. It can be easily implemented and is capable to generate acceptable compressed images at a rather high compression ratio. Basically, the JPEG compression consists of three stages: the first stage is to split the whole image into non-overlapping blocks of size 8×8 , and to apply the discrete cosine transformation (DCT) on each block; the second stage is to divide the above cosine transform coefficients by a quantization table pointwisely, and the quantized values are rounded to their nearest integers; the final stage is to use lossless compression coding (e.g., entropy coding) to generate a compressed data file. The decompression for JPEG images is also very simple. The procedure involves lossless decoding, dequantization and computing the inverse DCT to each block. In Fig. 1, we show the procedure description of the JPEG compression and decompression schemes, see [2] for more details.

One readily sees that the loss of the information for the JPEG compression takes place in the stage of quantization. The decompressed image is thus not exactly equal to the original input image. As round-off errors appear in each block, there are inevitably some artifacts in the decompressed image, see for instance the decompressed image in Fig. 2(b) where the

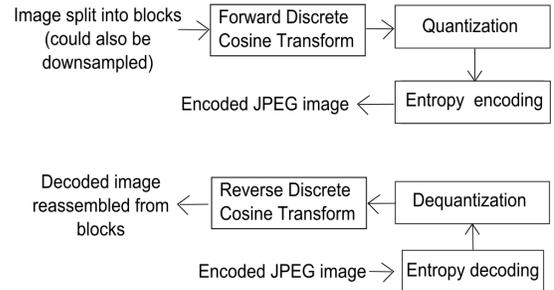


Fig. 1. The upper diagram: JPEG compression; the lower diagram: JPEG decompression.

original input image is shown in Fig. 2(a). In particular, when the compression ratio is high, more visible artifacts appear in the decompressed image, see Fig. 2(c). The main aim of this paper is to study an image processing method to improve the quality of decompressed images.

In literature, there are two main classes of methods to reduce the artifacts of JPEG decompressed images, namely, image enhancement methods and image restoration methods, see the review paper [3]. The image enhancement methods are heuristic approaches to improve the quantity of perceptual sense, while the image restoration approaches are based on the optimization of certain objective criteria with constraints to recover the original input images. The most recent image restoration methods include the projection onto convex sets [4], the stochastic methods as maximum a posteriori estimation (MAP) [5, 6], and the energy based methods [7, 8]. In particular, Bredies et al. [8] studied the total variation method and Alter and Durand [7] presented the weighted total variation method by setting special weights in the total variation term to reduce the artifacts of decompressed images. Although their restoration results are quite good, the basic assumption is that the minimizers of the variational models are piecewise constant, which could be violated for the original input images. Therefore, their decompressed models break the structure of texture regions of decompressed images.

All these methods referred above for JPEG decompression are basically pixel based methods. The image pixels are considered independently and the image features like texture structures, repeated patterns have not been well preserved. The sparse representation techniques are proved be very successful for Gaussian noise removal [9] and multiplicative noise removal [10] for gray images, color image denoising and inpainting [11], image sequence restoration [12], Poisson

H. Chang is with School of Mathematical Sciences, Tianjin Normal University, Tianjin, 300387 China. E-mail: changhui bin@gmail.com. The work is done when Dr. Chang visits the Department of Mathematics, Hong Kong Baptist University in 2013.

M. Ng and T. Zeng are with Centre for Mathematical Imaging and Vision, Department of Mathematics, Hong Kong Baptist University, Kowloon Tong, Hong Kong. Email: {mng,zeng}@hkbu.edu.hk .

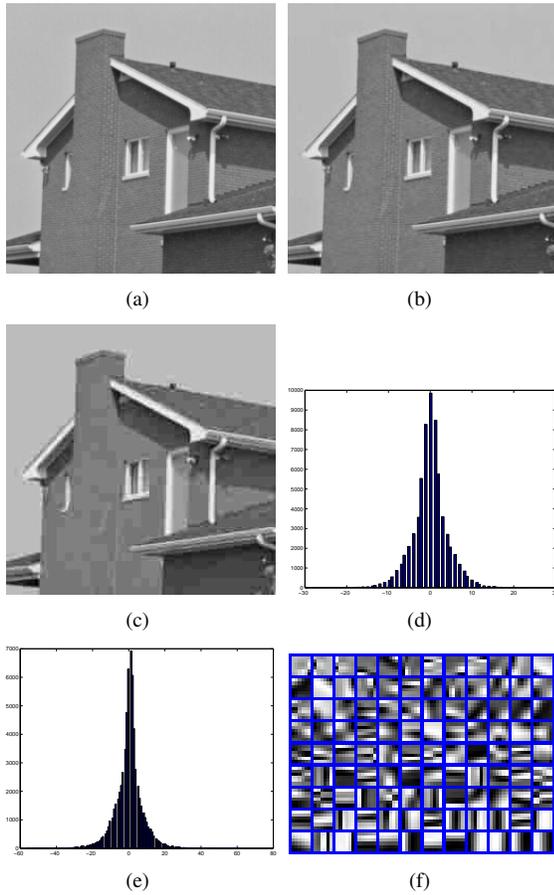


Fig. 2. (a): The original input image; (b): the JPEG decompressed image at a low compression ratio 12:1; (c): the JPEG decompressed image at a high compression ratio 30:1; (d): histogram of the difference between (a) and (b); (e): histogram of the difference between (a) and (c); (f): a dictionary learned from (c) by K-SVD.

image deblurring [13], etc. It is interesting to handle the JPEG decompression by sparse representation techniques for image patches. Jung et al. [14] proposed to derive a general dictionary from a training images set, which was used to remove the block artifacts of JPEG compression images. In this paper, we propose and develop a novel artifact reducing method for restoring JPEG decompressed images. Different to [14], our idea is to build a dictionary attached to the restored image and find a sparse representation over the learned dictionary. The restored image can keep the features of the original input image. Indeed, although the visible artifacts exist in the JPEG images as Fig. 2 (b) and (c), the histograms of the differences between the uncompressed image and the JPEG compressed images are symmetric and rapidly decrease to zero, which are the essential properties of Gaussian distribution, see Fig. 2 (d) and (e). Thus it is possible for us to dig out a dictionary to represent the typical features from the JPEG compressed images. In Fig. 2 (f), we show our learned dictionary by the classical K-SVD method [15] from the JPEG compressed image (Fig. 2 (c)). Clearly, the atoms in the dictionary are somewhat related to the typical patterns in the original image (Fig. 2 (a)) and they should be useful for the coming image decompressed procedure. Indeed, the later optimization problem for our

proposed model involves the summation of the fitting term between the restored image and the JPEG decompressed image (indicator function of a constraint set), the term of the sparse representation of the restored image in the dictionary, and the total variational regularization term. By this, an effective two-step algorithm for JPEG image decompression is thus obtained. The first step involves dictionary learning and the second step involves the total variation regularization for restoring images. Numerical experiments will be performed to demonstrate that the proposed method outperforms the total variation and weighted total variation decompression methods in the measure of peak of signal to noise ratio and structural similarity.

The remaining part of the paper is organized as follows. A brief review of the previous works on K-SVD and total variation decompression models is given in Section II. In Section III, we present the proposed model and algorithm. In Section IV, the numerical tests are done to demonstrate the efficiency of the proposed method. We conclude the paper in Section IV.

II. PREVIOUS WORKS

In the following two subsections, we give the brief review of K-SVD for the dictionary based model and the total variation decompression model.

A. K-SVD

Elad and Aharon [9] first proposed the K-SVD denoising method, which assumes that each image patch can be represented sparsely using a linear combination of the atoms from a special chosen dictionary. Specifically, it is “sparse” on the sense that $\mathbf{Y} \approx \mathbf{DX}$ with the sparse coefficient matrix \mathbf{X} , where \mathbf{D} is the dictionary, and the matrix \mathbf{Y} consists of all the image patches selected from the given image rewritten as a column vector. In order to remove some noise from \mathbf{f} with N pixels of size $n \times n$, which is rewritten as column vector $\mathbf{f} \in \mathbb{R}^N$ using the lexicographical ordering, the K-SVD denoising model was given

$$\min_{\{\gamma_{i,j}\}, \mathbf{D}, \mathbf{u}} \lambda \|\mathbf{f} - \mathbf{u}\|^2 + \frac{1}{2} \sum_{(i,j) \in \mathcal{P}} \left(\|\mathbf{D}\gamma_{i,j} - \mathbf{R}_{i,j}\mathbf{u}\|^2 + \mu_{i,j} \|\gamma_{i,j}\|_0 \right) \quad (1)$$

to generate a learned dictionary $\mathbf{D} \in \mathbb{R}^{m^2 \times c}$ and the recovered result \mathbf{u} . \mathbf{D} is a dictionary of size m^2 -by- c attached to the restored image with c atoms in the dictionary; $\mathbf{R}_{i,j}$ is the sampling matrix of size m^2 -by- N to construct a patch for the part of \mathbf{u} ; $\gamma_{i,j}$ is a vector of size c -by-1 containing the encoding coefficients for the patch of \mathbf{u} represented in the dictionary; $\mathcal{P} = \{1, 2, \dots, n - m + 1\}^2$ denotes the index set for different patches of \mathbf{u} ; $\|\cdot\|_2$ denotes the Euclidean norm of a vector; $\|\cdot\|_0$ denotes the number of non-zero elements; The parameter λ is a positive parameter of data fitting term, and $\mu_{i,j}$ is the positive patch-specific weight.

The first term in (1) is the data fitting term; the second term requires the image patch $\mathbf{R}_{i,j}\mathbf{u}$ can be represented by the given dictionary \mathbf{D} with the coefficient $\gamma_{i,j}$, which is sparse as

the third term requires the number of its nonzero elements is small. A two-step algorithm was adopted in [9]: determining the dictionary \mathbf{D} and sparse coefficients $\gamma_{i,j}$ by K-SVD [15] in the first step and taking the average of the denoising patches and the noisy image with the weight λ to obtain the final results in the second step.

B. Total variation decomposition model for JPEG

Based on the energy minimization methods, Bredies and Holler [8] proposed a total variation based model

$$\min_{\mathbf{u} \in \mathbf{U} \cap BV(\Omega)} TV(\mathbf{u}) \quad (2)$$

where

$$TV(\mathbf{u}) = \sup_{\mathbf{p} \in C_c^1(\Omega, \mathbb{R}^2)} \left\{ - \int_{\Omega} \mathbf{u} \operatorname{div} \mathbf{p} \, dx : |\mathbf{p}(x)| \leq 1 \right\},$$

$$|\mathbf{p}| = \sqrt{p_1^2 + p_2^2}, \text{ with } \mathbf{p} = (p_1, p_2),$$

$$BV(\Omega) = \{ \mathbf{u} \in L^1(\Omega) : TV(\mathbf{u}) < +\infty \},$$

and ∇ , div are the gradient and divergence operator respectively; Alter and Durand [7] presented a weighted total variation model

$$\min_{\mathbf{u} \in \mathbf{U}} TV_{\alpha}(\mathbf{u}) = \int_{\Omega} \alpha(x) |\nabla \mathbf{u}| \, dx \quad (3)$$

where values of weight function $\alpha(x)$ were chosen to be larger on the boundary pixels of the 8×8 blocks of the compression images. The task of the above models is to recover the image from all the possible solutions belonging to the set \mathbf{U} , which is generated by JPEG compression (see details in the following section). In [8] Bredies and Holler considered a general case of \mathbf{U} which defined a pointwise restriction set with respect to any L^2 -orthonormal basis. Rigorous analysis was given in the continuous setting, and a fast primal-dual algorithm was proposed to solve the given model efficiently. However, the total variation based model assumed that the minimizer was piecewise constant. Different artifacts (i.e. staircase artifacts) are introduced, especially for the images with more textures. In the following section, based on the ideas of sparse representation and energy minimization methods, we will propose a novel decomposition method via a learned dictionary in order to decompress JPEG images with less artifacts and better features.

III. THE PROPOSED MODEL

The original input image is of size $n \times n$, i.e., $N = n^2$ pixels, and it is represented by a vector \mathbf{u} of size N in lexicographical ordering. Let us define a block discrete cosine transform N -by- N matrix \mathbf{A} which converts each 8×8 block of the original input image to its frequency domain. The quantization matrix is denoted by \mathbf{M}_q of size 8×8 with the quality index q . Here \mathbf{M}_q consists of delicately selected integers to balance the image quality and the storage size controlled by the index q . Let \mathbf{z} be the JPEG compressed data of \mathbf{u} . It holds for the round-off error in the stage of JPEG quantization

$$|[\mathbf{A}\mathbf{u}]_{(k-1)*n+l} / [\mathbf{M}]_{k,l} - [\mathbf{z}]_{(k-1)*n+l}| \leq 1/2, \quad 1 \leq k, l \leq n,$$

where \mathbf{M} is a matrix of size n -by- n with $M_{k,l} = [\mathbf{M}_q]_{\bar{k}, \bar{l}}$. \bar{k} and \bar{l} refer to the modulo operation of k and l respectively by 8. By setting

$$[\mathbf{b}]_{(k-1)*n+l} = [\mathbf{M}]_{(k-1)*n+l} / 2, \quad 1 \leq k, l \leq n$$

and

$$[\mathbf{w}]_{(k-1)*n+l} = [\mathbf{M}]_{(k-1)*n+l} \times [\mathbf{z}]_{(k-1)*n+l}, \quad 1 \leq k, l \leq n,$$

the above inequality reduces to

$$|[\mathbf{A}\mathbf{u}]_{(k-1)*n+l} - [\mathbf{w}]_{(k-1)*n+l}| \leq [\mathbf{b}]_{(k-1)*n+l}, \quad 1 \leq k, l \leq n.$$

For simplicity, we write

$$|\mathbf{A}\mathbf{u} - \mathbf{w}| \leq \mathbf{b}. \quad (\text{entrywise inequality})$$

The image restoration task is to recover \mathbf{u} from the following set

$$\mathbf{U} = \{ \mathbf{u} : |\mathbf{A}\mathbf{u} - \mathbf{w}| \leq \mathbf{b} \}.$$

In this paper, we develop a reducing artifact model for restoring JPEG decompressed images in the discrete setting:

$$\min_{\{\gamma_{i,j}\}, \mathbf{D}, \mathbf{u}} \sum_{(i,j) \in \mathbf{P}} \left(\frac{1}{2} \|\mathbf{R}_{i,j} \mathbf{u} - \mathbf{D} \gamma_{i,j}\|_2^2 + \mu_{i,j} \|\gamma_{i,j}\|_0 \right) + \lambda \mathcal{T}\mathcal{V}(\mathbf{u}) + \mathcal{U}(\mathbf{u}). \quad (4)$$

The notations used in the above objective functional are given as follows: $\mathcal{T}\mathcal{V}(\mathbf{u})$ is the discrete version [16] of the total variation $TV(u)$ introduced in Section II-B

$$\mathcal{T}\mathcal{V}(\mathbf{u}) := \sum_{i=1}^N \sqrt{([\nabla \mathbf{u}^{(x)}]_k)^2 + ([\nabla \mathbf{u}^{(y)}]_k)^2},$$

∇ is the discrete gradient operator with Neumann boundary conditions, $[\nabla \mathbf{u}^{(x)}]_k$ and $[\nabla \mathbf{u}^{(y)}]_k$ are the x -derivative and y -derivative values at the k -th pixel ($1 \leq k \leq N$) discretized by forward difference schemes;

$$\mathcal{U}(\mathbf{u}) = \begin{cases} 0, & \text{if } \mathbf{u} \in \mathbf{U} \\ +\infty, & \text{otherwise,} \end{cases} \quad (5)$$

and other notations are the same as Section II-A. In the optimization problem (4), the first term $\|\mathbf{R}_{i,j} \mathbf{u} - \mathbf{D} \gamma_{i,j}\|_2^2$ is related to the representation of the restored image in the dictionary. The second term $\|\gamma_{i,j}\|_0$ is used to require the encoding coefficients vector to be sparse. The combination of the first and second terms requires the restored image is a sparse linear combination of elements in the dictionary. The third term $\mathcal{T}\mathcal{V}(\mathbf{u})$ is used to minimize the total variation of the restored image. The fourth term $\mathcal{U}(\mathbf{u})$ is used to require the solution belonging to the given data requirement. The advantage of this model is that we determine a sparse representation of the restored image in the learned dictionary so that the restored image can keep the features of the original input image.

A. The Algorithm

In [9–12], a fixed dictionary is learned via K-SVD, and the restored images are then derived based on the learned dictionary. Similarly, we propose a two-step algorithm approximately to solve (4). We note from (4) that there are three sets of unknowns: $\{\gamma_{i,j}\}$, \mathbf{D} and \mathbf{u} . In this paper, the following two-step algorithm to determine these unknowns is given:

Algorithm I

1. Initialization: Parameters $\lambda, \mu_{i,j}$ and the JPEG compressed image \mathbf{u}^0 .

2. • (Step 1) Solve

$$(\gamma_{i,j}^*, \mathbf{D}^*) = \arg \min_{\gamma_{i,j}, \mathbf{D}} \sum_{(i,j) \in \mathcal{P}} \left(\frac{1}{2} \|\mathbf{R}_{i,j} \mathbf{u}^0 - \mathbf{D} \gamma_{i,j}\|_2^2 + \mu_{i,j} \|\gamma_{i,j}\|_0 \right) \quad (6)$$

• (Step 2) Solve

$$\mathbf{u}^* = \min_{\mathbf{u} \in \mathcal{U}} \lambda \mathcal{T} \mathcal{V}(\mathbf{u}) + \sum_{(i,j) \in \mathcal{P}} \frac{1}{2} \|\mathbf{R}_{i,j} \mathbf{u} - \mathbf{D}^* \gamma_{i,j}^*\|_2^2 \quad (7)$$

3. Output \mathbf{u}^* as the decompressed result.

In (6), the encoding coefficients $\gamma_{i,j}^*$ and the dictionary \mathbf{D}^* are required to be determined. We employ K-SVD [15] to solve this subproblem. More precisely, it is a two-step iterative method. In each iteration, the first step is to use the orthonormal matching pursuit (OMP) algorithm [17] to update the encoding coefficients:

$$\gamma_{i,j}^{*,s} = \arg \min_{\gamma_{i,j}} \|\gamma_{i,j}\|_0, \quad \text{s.t.} \quad \|\mathbf{R}_{i,j} \mathbf{u}^0 - \mathbf{D}^{*,s-1} \gamma_{i,j}\|_2 \leq \delta, \quad (8)$$

for each $(i,j) \in \mathcal{P}$; the second step is to use SVD to update the dictionary:

$$\min_{\mathbf{D}} \sum_{(i,j) \in \mathcal{P}} \frac{1}{2} \|\mathbf{R}_{i,j} \mathbf{u}^0 - \mathbf{D} \gamma_{i,j}^{*,s}\|_2^2,$$

where the index s denotes the s -th iteration.

In order to solve the subproblem in (7), we introduce the following operators first

$$\hat{\mathcal{F}}(\mathbf{p}) = \sum_{1 \leq i \leq N} \sqrt{\mathbf{p}_{i,1}^2 + \mathbf{p}_{i,2}^2}, \quad \forall \mathbf{p} \in \mathbb{R}^{N \times 2},$$

$$\hat{\mathcal{G}}(\mathbf{u}) = \sum_{(i,j) \in \mathcal{P}} \frac{1}{2\lambda} \|\mathbf{R}_{i,j} \mathbf{u} - \mathbf{D}^* \gamma_{i,j}^*\|_2^2, \quad \forall \mathbf{u} \in \mathbb{R}^N,$$

and express (7) as follows:

$$\min_{\mathbf{u} \in \mathcal{U}} \hat{\mathcal{F}}(\nabla \mathbf{u}) + \hat{\mathcal{G}}(\mathbf{u}).$$

In order to deal with the constraints and the non-differential total variation term, auxiliary vectors $\mathbf{v}, \mathbf{d} \in \mathbb{R}^N$, and the dual variable $\mathbf{p} \in \mathbb{R}^{N \times 2}$ are introduced as

$$\max_{\mathbf{p}, \mathbf{d}} \min_{\mathbf{v}, \mathbf{u}} \langle \mathbf{p}, \nabla \mathbf{u} \rangle + \langle \mathbf{d}, \mathbf{u} - \mathbf{v} \rangle + \hat{\mathcal{G}}(\mathbf{u}) + \mathcal{U}(\mathbf{v}),$$

where the first term is introduced due to the dual formulation of total variation, the second term is introduced to deal with the equality $\mathbf{u} = \mathbf{v}$, $\langle \cdot, \cdot \rangle$ usually denotes the inner product of two vectors, and we still use this notation to denote the inner product of $\mathbf{p}, \mathbf{q} \in \mathbb{R}^{N \times 2}$ as

$$\langle \mathbf{p}, \mathbf{q} \rangle = \sum_{1 \leq i \leq N} ([\mathbf{p}]_{i,1} [\mathbf{q}]_{i,1} + [\mathbf{p}]_{i,2} [\mathbf{q}]_{i,2}).$$

For simplicity, it is equivalent to solve the following saddle point problem [16, 18]

$$\min_{\mathbf{u}, \mathbf{v}} \max_{\mathbf{p}, \mathbf{d}} \langle \mathbf{p}, \mathbf{d} \rangle, K(\mathbf{u}, \mathbf{v}) + \mathcal{G}(\mathbf{u}, \mathbf{v}) - \mathcal{F}(\mathbf{p}, \mathbf{d}),$$

where

$$K(\mathbf{u}, \mathbf{v}) = \langle \nabla \mathbf{u}, \mathbf{u} - \mathbf{v} \rangle,$$

\mathcal{F} and \mathcal{G} are denoted as

$$\mathcal{F}(\mathbf{p}, \mathbf{d}) = \begin{cases} 0, & \text{if } ([\mathbf{p}]_{i,1})^2 + ([\mathbf{p}]_{i,2})^2 \leq 1, \quad \forall 1 \leq i \leq N \\ +\infty, & \text{otherwise} \end{cases} \quad (9)$$

and

$$\mathcal{G}(\mathbf{u}, \mathbf{v}) = \hat{\mathcal{G}}(\mathbf{u}) + \mathcal{U}(\mathbf{v})$$

respectively; the inner product is denoted as

$$\langle \langle \mathbf{p}, \mathbf{d} \rangle, \langle \mathbf{q}, \mathbf{b} \rangle \rangle = \langle \mathbf{p}, \mathbf{q} \rangle + \langle \mathbf{d}, \mathbf{b} \rangle, \quad \forall (\mathbf{p}, \mathbf{d}), (\mathbf{q}, \mathbf{b}) \in \mathbb{R}^{N \times 2} \otimes \mathbb{R}^N.$$

Based on the above definitions, we employ the first order primal-dual algorithm [16, 18] to update the restored image. The algorithm for solving (7) is listed as follows.

Algorithm II for solving (7)

1. Initialization: Parameters τ, σ , and $\mathbf{u}_0, \mathbf{v}_0, \mathbf{p}_0, \mathbf{d}_0, \bar{\mathbf{u}}_0, \bar{\mathbf{v}}_0$

2. Compute $\mathbf{p}_k, \mathbf{d}_k, \mathbf{u}_k, \mathbf{v}_k, \bar{\mathbf{u}}_k$ and $\bar{\mathbf{v}}_k$ iteratively for $k = 1, 2, \dots, L$

• (Step 1) Compute

$$(\mathbf{p}_k, \mathbf{d}_k) = (\mathcal{I} + \sigma \partial \mathcal{F})^{-1}((\mathbf{p}_{k-1}, \mathbf{d}_{k-1}) + \sigma K(\bar{\mathbf{u}}_{k-1}, \bar{\mathbf{v}}_{k-1})) \quad (10)$$

• (Step 2) Compute

$$(\mathbf{u}_k, \mathbf{v}_k) = (\mathcal{I} + \tau \partial \mathcal{G})^{-1}((\mathbf{u}_{k-1}, \mathbf{v}_{k-1}) - \tau K^*(\mathbf{p}_k, \mathbf{d}_k)) \quad (11)$$

• (Step 3) Compute

$$(\bar{\mathbf{u}}_k, \bar{\mathbf{v}}_k) = 2(\mathbf{u}_k, \mathbf{v}_k) - (\mathbf{u}_{k-1}, \mathbf{v}_{k-1})$$

3. Output \mathbf{u}_L

In Algorithm II, \mathcal{I} refers to the identity operator, and one readily obtains the conjugate operator K^* of K as

$$K^*(\mathbf{p}, \mathbf{d}) = (-\text{div} \mathbf{p} + \mathbf{d}, -\mathbf{d}),$$

where div denotes the discrete divergence operator satisfying the following relation

$$\langle \nabla \mathbf{u}, \mathbf{p} \rangle = -\langle \mathbf{u}, \text{div} \mathbf{p} \rangle.$$

Now the key issue is to solve the two subproblems in (10) and (11). For (10), define

$$(\mathbf{p}^*, \mathbf{d}^*) := (\mathcal{I} + \sigma \partial \mathcal{F})^{-1}(\hat{\mathbf{p}}, \hat{\mathbf{d}}),$$

where the closed-form solutions are given by

$$[\mathbf{p}^*]_{i,j} = \begin{cases} [\hat{\mathbf{p}}]_{i,j}, & \text{if } (\hat{\mathbf{p}}_{i,1})^2 + (\hat{\mathbf{p}}_{i,2})^2 \leq 1; \\ \frac{\hat{\mathbf{p}}_{i,j}}{\sqrt{(\hat{\mathbf{p}}_{i,1})^2 + (\hat{\mathbf{p}}_{i,2})^2}}, & \text{otherwise,} \end{cases}$$

for $1 \leq i \leq N$ and $1 \leq j \leq 2$, and $\mathbf{d}^* = \hat{\mathbf{d}}$.

For the subproblem in (11), we get

$$\begin{aligned} & (\mathbf{u}^*, \mathbf{v}^*) \\ &= (\mathcal{I} + \tau \partial \mathcal{G})^{-1}(\hat{\mathbf{u}}, \hat{\mathbf{v}}) \\ &= \arg \min_{\mathbf{u}, \mathbf{v}} \frac{1}{2\tau} \|\mathbf{u} - \hat{\mathbf{u}}\|_2^2 + \sum_{(i,j) \in \mathcal{P}} \frac{1}{2\lambda} \|\mathbf{R}_{i,j} \mathbf{u} - \mathbf{D}^* \gamma_{i,j}^*\|_2^2 \\ &\quad + \frac{1}{2\tau} \|\mathbf{v} - \hat{\mathbf{v}}\|_2^2 + \mathcal{U}(\mathbf{v}) \\ &:= \arg \min_{\mathbf{u}, \mathbf{v}} \mathcal{E}_1(\mathbf{u}) + \mathcal{E}_2(\mathbf{v}) \end{aligned} \quad (13)$$

where

$$\mathcal{E}_1(\mathbf{u}) = \frac{1}{2\tau} \|\mathbf{u} - \hat{\mathbf{u}}\|_2^2 + \sum_{(i,j) \in \mathcal{P}} \frac{1}{2\lambda} \|\mathbf{R}_{i,j} \mathbf{u} - \mathbf{D}^* \gamma_{i,j}^*\|_2^2,$$

and

$$\mathcal{E}_2(\mathbf{v}) = \frac{1}{2\tau} \|\mathbf{v} - \hat{\mathbf{v}}\|_2^2 + \mathcal{U}(\mathbf{v}).$$

The minimization problem in (13) can be solved by minimizing $\mathcal{E}_1(\mathbf{u})$ and $\mathcal{E}_2(\mathbf{v})$ independently.

We first express $\frac{1}{2} \sum_{i,j} \|\mathbf{R}_{i,j} \mathbf{u} - \mathbf{D}^* \gamma_{i,j}^*\|_2^2$ by

$$\frac{1}{2} \langle \mathbf{W} \mathbf{u}, \mathbf{u} \rangle + \frac{1}{2} \sum_{i,j} \|\mathbf{D}^* \gamma_{i,j}^*\|_2^2 - \langle \boldsymbol{\psi}, \mathbf{u} \rangle,$$

where \mathbf{W} is a diagonal matrix with its main diagonal entries given by $\sum_{i,j} \mathbf{R}_{i,j}^* \mathbf{R}_{i,j}$ (it is used to count how many times each pixel is used to construct the patches), $\boldsymbol{\psi} = \sum_{i,j} \mathbf{R}_{i,j}^* \mathbf{D}^* \gamma_{i,j}^*$.

Thus the subproblem of minimizing $\mathcal{E}_1(\mathbf{u})$ is reduced to

$$\begin{aligned} \mathbf{u}^* &= \arg \min_{\mathbf{u}} \frac{\lambda}{2\tau} \|\mathbf{u} - \hat{\mathbf{u}}\|_2^2 + \frac{1}{2} \langle \mathbf{W} \mathbf{u}, \mathbf{u} \rangle - \langle \boldsymbol{\psi}, \mathbf{u} \rangle \\ &= (\lambda \mathcal{I} + \tau \mathbf{W})^{-1} (\lambda \hat{\mathbf{u}} + \tau \boldsymbol{\psi}). \end{aligned} \quad (14)$$

Since the matrix \mathbf{W} is diagonal, the equation can be solved by the pointwise division.

The subproblem of minimizing $\mathcal{E}_2(\mathbf{v})$ has a closed form solution given by $\mathbf{v} = \mathcal{Q}(\hat{\mathbf{v}})$, where

$$[\mathcal{Q}(\hat{\mathbf{v}})]_i = \begin{cases} [\hat{\mathbf{v}}]_i, & |[\hat{\mathbf{v}}]_i - [\mathbf{w}]_i| \leq [\mathbf{b}]_i, \\ [\mathbf{w}]_i + [\mathbf{b}]_i, & [\hat{\mathbf{v}}]_i > [\mathbf{w}]_i + [\mathbf{b}]_i, \\ [\mathbf{w}]_i - [\mathbf{b}]_i, & [\hat{\mathbf{v}}]_i < [\mathbf{w}]_i - [\mathbf{b}]_i, \end{cases}$$

for $1 \leq i \leq N$.

By [16], one can easily prove the convergence of Algorithm II if $\tau\sigma < \frac{1}{\|K\|^2}$. Thus one shall estimate the norm of K .

Since $\|\nabla\|^2 \leq 8$ in [19],

$$\begin{aligned} \|K(\mathbf{u}, \mathbf{v})\|^2 &= \|\nabla \mathbf{u}\|_2^2 + \|\mathbf{u} - \mathbf{v}\|_2^2 \\ &\leq 9\|\mathbf{u}\|_2^2 + \|\mathbf{v}\|_2^2 + 2\langle \mathbf{u}, \mathbf{v} \rangle \\ &\leq (9 + 1/t)\|\mathbf{u}\|_2^2 + (1 + t)\|\mathbf{v}\|_2^2, \end{aligned}$$

for the arbitrary positive constant t . By choosing $t = \sqrt{17} + 4$, s.t. $9 + 1/t = 1 + t$, we obtain $\|K\|^2 \leq 5 + \sqrt{17}$.

Proposition 1: Algorithm II is convergent if $\tau\sigma < \frac{5 - \sqrt{17}}{8}$.

Let us analyze the computational complexity of the proposed method to solve (4). The complexity of K-SVD in Step 1 of Algorithm I is $O(cm^2 J \xi N)$ [9] where J is the number of K-SVD iterations, $m \times m$ is the image patch size, ξ is the average of number of nonzero elements in the encoding coefficients vectors $\gamma_{i,j}$, c is the number of atoms in the dictionary and N is the number of pixels. The computational cost of Algorithm II is of $O(NL)$ operations where L is the number of iterations required for Algorithm II. Finally, the total complexity of the proposed method is $O((cm^2 J \xi + L)N)$ which is linear with respect to the number of pixels of the decompressed image.

IV. NUMERICAL EXAMPLES

In this section, we conduct experiments to illustrate the performance of the proposed method. We use q in between 0 and 100 to represent the quality of compression. When q is large (or small), the compression ratio is low (or high). The standard quantization matrix [1] \mathbf{M}_{50} for $q = 50$ is given as follows:

$$\begin{bmatrix} 16 & 11 & 10 & 16 & 24 & 40 & 51 & 61 \\ 12 & 12 & 14 & 19 & 26 & 58 & 60 & 55 \\ 14 & 13 & 16 & 24 & 40 & 57 & 69 & 56 \\ 14 & 17 & 22 & 29 & 51 & 87 & 80 & 62 \\ 18 & 22 & 37 & 56 & 68 & 109 & 103 & 77 \\ 24 & 35 & 55 & 64 & 81 & 104 & 113 & 92 \\ 49 & 64 & 78 & 87 & 103 & 121 & 120 & 101 \\ 72 & 92 & 95 & 98 & 112 & 100 & 103 & 99 \end{bmatrix}.$$

This quantization matrix is designed based on human visual system. According to \mathbf{M}_{50} , we generate the other quantization matrices with $q \leq 50$ where

$$\mathbf{M}_q = \text{round}(50\mathbf{M}_{50}/q).$$

as suggested in [20]. Seven uncompressed 256×256 images are tested and shown in the first column of Fig. 3, whose names are ‘‘Wall’’, ‘‘Letters’’, ‘‘Lena’’, ‘‘Barbara’’, ‘‘Boat’’, ‘‘House’’, and ‘‘Peppers’’ from top to bottom, respectively. Note that $N = 256^2$.

We compare the proposed DicTV algorithm with the TV algorithm [8] and the WTV algorithm in [7]. Primal dual algorithm is used to solve TV, and the step-size parameters σ and τ are set to be $\sigma = \tau = \sqrt{1/8}$. The weighted total variation term for WTV [7] is discretized as

$$TV_\alpha(\mathbf{u}) = \sum_{1 \leq i,j \leq n} \sqrt{a_{i,j}^2 + b_{i,j}^2 + c_{i,j}^2 + d_{i,j}^2},$$

with

$$\begin{aligned} a_{i,j} &= \alpha_i \% 8 ([\mathbf{u}]_{i*n+j} - [\mathbf{u}]_{(i-1)*n+j}), \\ b_{i,j} &= \alpha_j \% 8 ([\mathbf{u}]_{(i-1)*n+j+1} - [\mathbf{u}]_{(i-1)*n+j}), \\ c_{i,j} &= \alpha_{(i-1)} \% 8 ([\mathbf{u}]_{(i-1)*n+j} - [\mathbf{u}]_{(i-2)*n+j}), \\ d_{i,j} &= \alpha_{(j-1)} \% 8 ([\mathbf{u}]_{(i-1)*n+j} - [\mathbf{u}]_{(i-1)*n+j-1}), \end{aligned}$$

where $\%$ is the modulo operator. The weight α_i is set to be

$$\alpha_0 = \alpha_6 = 2, \alpha_1 = \alpha_5 = 1.5, \alpha_2 = \alpha_3 = \alpha_4 = 1, \alpha_7 = 3.$$

Projected gradient descent algorithm is used to solve WTV with iteration-dependent step-size of $\frac{1}{i+1}$, i is the iteration number. We stop TV and WTV when the relative error of u_k satisfies $\frac{\|u_k - u_{k-1}\|_2}{\|u_k\|_2} \leq 1 \times 10^{-3}$ or the iteration numbers reach 100. The maximum PSNR values can often be reached before the final iteration.

In the proposed DicTV algorithm, the Matlab package by Rubinstein [21] is used directly to realize the K-SVD computation in Step 1 of Algorithm I. The image patch size is set to be 6×6 (i.e., $m = 6$), and therefore the number of patches is equal to 251×251 . The number of atoms in the dictionary is set to be $108 = 6^2 \times 3$ (the redundancy factor is chosen to be 3 empirically), and the size of the dictionary is equal to $6^2 \times 108$. In Algorithm I, λ is set to be 50 for texture images “Barbara”, “Wall” and “Letters”, and to be 150 for others. In Step 1 of Algorithm I, the number J of iterations is set to be 20 to update the encoding coefficients and learn the dictionary iteratively. The parameters σ, τ are set to be $\sigma = \tau = \sqrt{1/10}$ in Algorithm II. The same stopping condition is used for Algorithm II as that used by TV and WTV. We also consider the case without TV term (“Dic” for short) by setting $\lambda = 0$ in (4).

Two evaluation criteria are used to measure the quality of the restored image from the decompression. The first one is peak of signal to noise ratio (PSNR):

$$\text{PSNR}(\mathbf{u}, \mathbf{u}_r) = 10 \log_{10} \left(\frac{255^2}{\sqrt{\sum_{1 \leq i \leq N} \frac{(\mathbf{u}_i - [\mathbf{u}_c]_i)^2}{N}}} \right).$$

The second one is the SSIM metric [22]. Assuming that $\mathbf{u}(i)$ and $\mathbf{u}_r(i)$ are the 11-by-11 sub-images (rewritten as column vector using the lexicographical ordering) centered at the i -th pixel location of two images \mathbf{u} and \mathbf{u}_r respectively, the local SSIM index is defined by

$$\begin{aligned} & \text{SSIM}_{local}(\mathbf{u}(i), \mathbf{u}_r(i)) \\ &= \frac{[2\mu(\mathbf{u}(i))\mu(\mathbf{u}_r(i)) + c_1][2\sigma(\mathbf{u}(i)\mathbf{u}_r(i)) + c_2]}{[\mu^2(\mathbf{u}(i)) + \mu^2(\mathbf{u}_r(i)) + c_1][\sigma^2(\mathbf{u}(i)) + \sigma^2(\mathbf{u}_r(i)) + c_2]} \end{aligned}$$

where $\mu(\mathbf{u}(i))$ and $\mu(\mathbf{u}_r(i))$ are the mean values of $\mathbf{u}(i)$ and $\mathbf{u}_r(i)$, $\sigma(\mathbf{u}(i))^2$ and $\sigma(\mathbf{u}_r(i))^2$ are the related variances, respectively; $\sigma(\mathbf{u}(i)\mathbf{u}_r(i))$ is the covariance of $\mathbf{u}(i)$ and $\mathbf{u}_r(i)$, and c_1, c_2 are two constants dependent on the dynamic range of \mathbf{u} and \mathbf{u}_r . The average SSIM index,

$$\text{SSIM}(\mathbf{u}, \mathbf{u}_r) = \frac{1}{N} \sum_{i=1}^N \text{SSIM}_{local}(\mathbf{u}(i), \mathbf{u}_r(i)),$$

is used to evaluate the overall image quality. The larger the value is, the better the restoration result we have. TV, WTV and DicTV are all coded in **Matlab** and the numerical tests are done by **MatlabR2011a** on desktop computer with Inter(R) Core(TM) 2 Quad Q9450@2.66GHz CPU and 4G Ram.

We give the decompressed results by different methods when $q = 50/3$ in Fig. 3. The corresponding zoomed parts are shown in Fig. 4. We see that the proposed Dic and DicTV methods produce decompressed images with more features than the others, and do not introduce other new artifacts. JPEG artifacts are reduced greatly as well. Specifically, texture parts are well restored than others in 1st, 2nd and 4th rows of Fig. 4; edges are well restored in 5th and 6th rows of Fig. 4; flat regions are much smoother in the 3rd and 7th rows of Fig. 4. In Table I, we test several values of q and report the PSNR and SSIM values of decompressed images by different methods. The maximum PSNR values among all iterations and corresponding SSIM values are chosen to be shown for TV and WTV. According to this table, The PSNR and SSIM values of decompression images by WTV are not always better than those by TV, as [8] has pointed out. One readily sees that the proposed DicTV method is better than the other two methods to derive higher quality images, especially for images with more textures as “Barbara”, “Wall”, and “Letters”. The average PSNR and SSIM values are given in the last two columns of Table I. The average PSNR values for the decompressed images by using our proposed methods “Dic” and “DicTV” are almost 0.5db higher than those by the TV and WTV methods.

By comparing the decompressed images by Dic and DicTV, the PSNR and SSIM differences are small. However, Dic without the TV regularization will decompress the image with some visible residual artifacts. One can observe the differences between the images by DicTV and Dic from Fig. 5 and Fig. 6, especially in the regions marked by green rectangles. Some visible artifacts exist near the edges of Fig. 5 (c) and (e). Similar artifacts exist in Fig. 6 (c) and (g) as well. Dic method just learns the repeated patterns, and can not guarantee to removal the artifacts from the JPEG compressed images completely. By adding the TV term, it helps to remove the visible artifacts better than the Dic method, see Fig. 5 (d), (f) and Fig. 6 (d), (h) for details.

It takes 11 seconds (37 iterations), 11 seconds (36 iterations), and 19 seconds (9s dictionary learning and 10s for Algorithm II) for TV, WTV and DicTV to test the image “Peppers” with $q = 50/3$, respectively. The computation time of K-SVD is about 50% of total computational time for DicTV. One may consider how to accelerate the K-SVD algorithm or adapt other more fast dictionary learning schemes instead to reduce the total computation cost for DicTV.

We study the sensitivity of the parameter λ . Different values of λ are set to be 1, 10, 100, 1000 when $q = 50/3$, and the results are put in Fig. 7. By observing the given results, our proposed model is rather robust for a wide range of values of λ . By setting larger $\lambda = 1000$, the decompressed result is quite close to that by TV, and obvious staircase artifacts appear as well. Therefore, we propose to choose smaller λ for our proposed model.

q	Name		JPEG	TV	WTV	Dic	DicTV
10	Wall	PSNR	23.25	23.76	23.96	25.07	25.00
		SSIM	0.8892	0.8891	0.8949	0.9208	0.9174
	Letters	PSNR	19.51	20.32	20.33	21.08	21.09
		SSIM	0.7823	0.8880	0.8795	0.9181	0.9194
	Lena	PSNR	28.31	29.26	29.24	29.28	29.39
		SSIM	0.8132	0.8505	0.8551	0.8555	0.8565
	Barbara	PSNR	26.28	26.72	26.88	27.28	27.24
		SSIM	0.7879	0.8014	0.8100	0.8158	0.8141
	Boat	PSNR	26.94	27.67	27.67	27.92	27.89
		SSIM	0.7703	0.7911	0.7937	0.7952	0.7936
	House	PSNR	30.55	31.60	31.56	31.79	31.95
		SSIM	0.8183	0.8443	0.8448	0.8484	0.8502
	Peppers	PSNR	29.92	31.13	31.16	31.29	31.39
		SSIM	0.8379	0.8826	0.8863	0.8907	0.8920
50/3	Wall	PSNR	25.71	26.41	26.44	27.86	27.90
		SSIM	0.9378	0.9425	0.9434	0.9613	0.9606
	Letters	PSNR	21.29	22.40	22.32	23.15	23.22
		SSIM	0.8266	0.9380	0.9258	0.9497	0.9490
	Lena	PSNR	30.14	30.89	30.88	31.01	31.00
		SSIM	0.8657	0.8878	0.8892	0.8883	0.8881
	Barbara	PSNR	28.50	28.99	29.13	29.69	29.67
		SSIM	0.8640	0.8743	0.8785	0.8797	0.8807
	Boat	PSNR	28.74	29.43	29.41	29.63	29.61
		SSIM	0.8240	0.8402	0.8413	0.8410	0.8394
	House	PSNR	32.45	33.27	33.19	33.30	33.50
		SSIM	0.8520	0.8667	0.8648	0.8667	0.8687
	Peppers	PSNR	32.13	33.24	33.21	33.35	33.47
		SSIM	0.8918	0.9195	0.9201	0.9227	0.9234
25	Wall	PSNR	27.72	28.40	28.33	29.57	29.70
		SSIM	0.9604	0.9641	0.9639	0.9741	0.9743
	Letters	PSNR	22.87	24.29	24.14	24.90	24.99
		SSIM	0.8590	0.9619	0.9530	0.9661	0.9673
	Lena	PSNR	31.39	32.05	32.02	32.17	32.17
		SSIM	0.8947	0.9106	0.9102	0.9126	0.9108
	Barbara	PSNR	30.41	30.80	30.92	31.63	31.52
		SSIM	0.9034	0.9115	0.9127	0.9153	0.9154
	Boat	PSNR	30.11	30.79	30.75	30.94	30.94
		SSIM	0.8585	0.8710	0.8706	0.8706	0.8691
	House	PSNR	33.72	34.41	34.29	34.42	34.53
		SSIM	0.8741	0.8829	0.8803	0.8808	0.8831
	Peppers	PSNR	33.76	34.77	34.75	34.91	34.95
		SSIM	0.9193	0.9384	0.9378	0.9396	0.9396
Average	PSNR	28.27	29.08	29.08	29.54	29.58	
	SSIM	0.8586	0.8884	0.8884	0.8959	0.8958	

TABLE I
PSNR AND SSIM VALUES FOR DIFFERENT q

Finally, the convergent study of Algorithm II is done. Define

$$E_k = \lambda TV(\mathbf{u}_k) + \frac{1}{2} \langle \mathbf{W} \mathbf{u}_k, \mathbf{u}_k \rangle - \langle \psi, \mathbf{u}_k \rangle$$

The histories of relative errors of \mathbf{u} , error between \mathbf{u}_k and \mathbf{v}_k , PSNR values and objective functional energy values E_k are shown in Fig. 8, which demonstrate that the proposed algorithm converges well.

CONCLUSION

In this paper, we have presented a JPEG image decompression approach to reduce artifact via the learned dictionary, which outperforms the total variation and weighted total variation decompression methods. An efficient algorithm is given to solve the proposed model as well. Experimental results have shown that the proposed method is better than the other testing methods. As a future research, we would like to explore how to design an efficient dictionary algorithm to

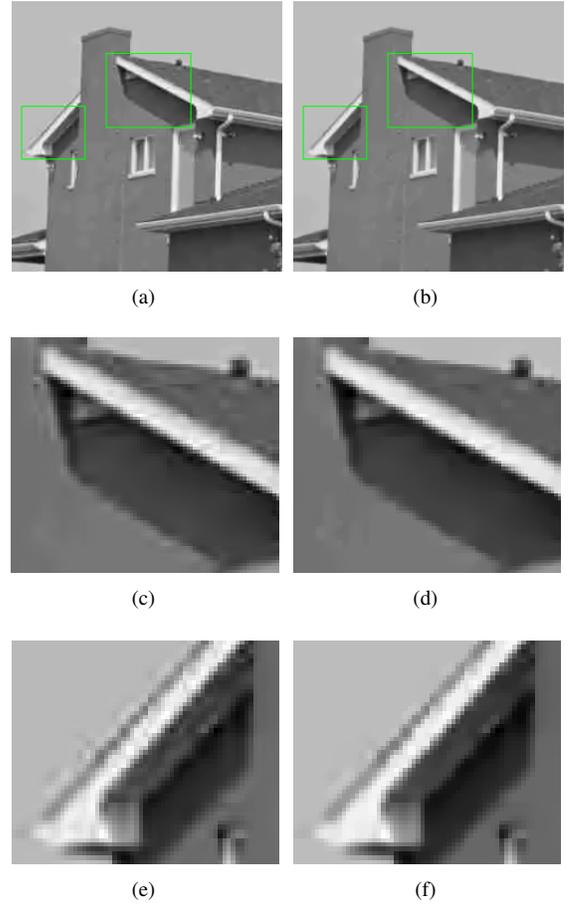


Fig. 5. (a): Decompressed images of House by Dic; (b): decompressed images of House by DicTV; (c) and (e): zoomed parts by Dic; (d) and (f): zoomed parts by DicTV when $q = 50/3$.

reduce the computational time. On the other hand, because some high frequency information is lost after quantization, it is interesting to design other compression schemes with the proposed method together to restore highly compressed images.

ACKNOWLEDGMENT

We would like to thank Prof. Sylvain Durand, in M.A.P.5, Université Paris Descartes and Prof. Jacques Froment in Université de Bretagne Sud for kindly providing us with their c source codes for the weighted total variation based decompression model.

REFERENCES

- [1] G. Wallace, “The JPEG still picture compression standard,” *Commun. ACM*, vol. 34, no. 4, pp. 30–44, 1991.
- [2] Wikipedia, “JPEG,” <http://en.wikipedia.org/wiki/JPEG>.
- [3] M.-Y. Shen and C.-C. J. Kuo, “Review of postprocessing techniques for compression artifact removal,” *J. Visual Commun. Image Representation*, vol. 9, pp. 2–14, 1998.
- [4] Y. Yang, N. Galatsanos, and A. Katsaggelos, “Regularized reconstruction to reduce blocking discrete cosine transform compressed images,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 3, pp. 421–432, 1993.

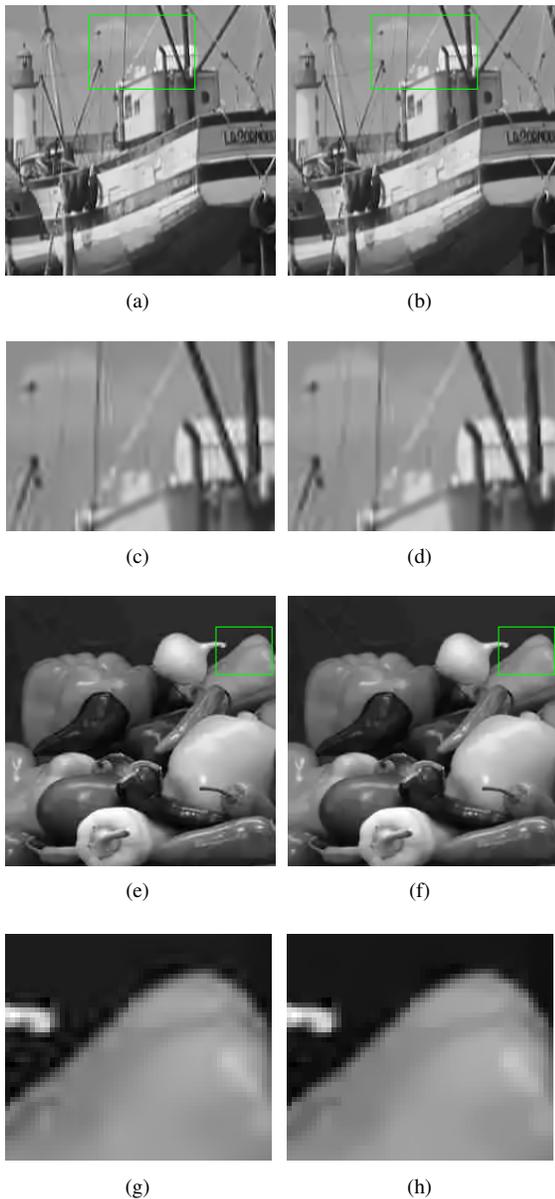


Fig. 6. (a): Decompressed images of Boat by Dic; (b): decompressed images of Boat by DicTV; (c): zoomed parts of Boat by Dic; (d): zoomed parts of Boat by DicTV; (e): decompressed images of Peppers by Dic; (f): decompressed images of Peppers by DicTV; (g): zoomed parts of Peppers by Dic; (h): Zoomed parts of Peppers by DicTV when $q = 50/3$.

- [5] T. Ozcelik, J. Brailean, and A. Katsaggelos, "Image and video compression algorithms based on recovery techniques using mean field annealing," in *Proc. IEEE*, vol. 83, 1995.
- [6] H. Noda, S. Haraguchi, and M. Niimi, "Local map estimation for quality improvement of compressed color images," in *TENCON 2010 - 2010 IEEE Region 10 Conference*, 2010.
- [7] F. Alter, S. Durand, and J. Froment, "Adapted total variation for artifact free decomposition of JPEG images," *J. Math. Imaging Vision*, vol. 23, no. 2, pp. 199–211, 2005.
- [8] K. Bredies and M. Holler, "A total variation-based JPEG decomposition model," *SIAM J. Imaging Sci.*, vol. 5,

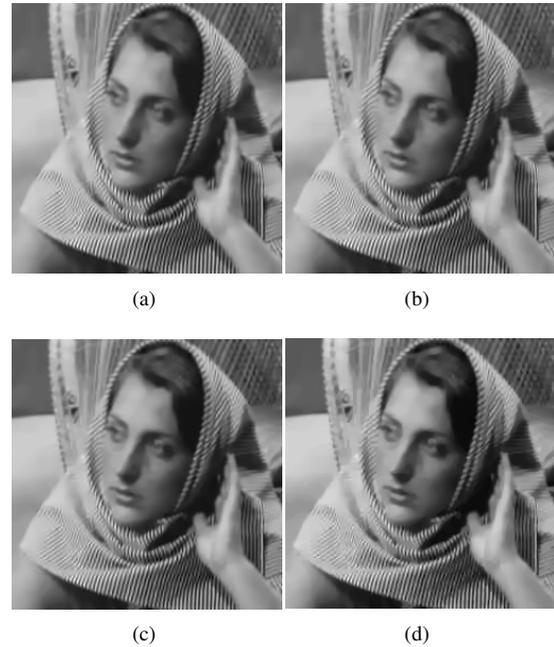


Fig. 7. (a): $\lambda = 1$ with PSNR=29.67; (b): $\lambda = 10$ with PSNR=29.75; (c): $\lambda = 100$ with PSNR=29.51; (d): $\lambda = 1000$ with PSNR=28.55 when $q = 50/3$.

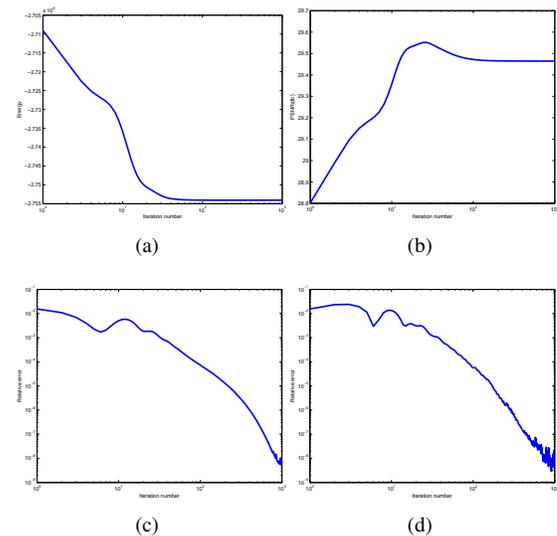


Fig. 8. Histories for (a) objective functional values, (b) PSNR, (c) $\frac{\|u_k - u_{k-1}\|_2}{\|u_k\|_2}$, (d) $\frac{\|u_k - v_k\|_2}{\|u_k\|_2}$ when $\lambda = 100$ for DicTV.

- no. 1, pp. 366–393, 2012.
- [9] M. Elad and M. Aharon, "Image denoising via sparse and redundant representations over learned dictionaries," *IEEE Trans. Image Process.*, vol. 15, no. 12, pp. 3736–3745, 2006.
- [10] Y. Huang, L. Moisan, M. Ng, and T. Zeng, "Multiplicative noise removal via a learned dictionary," *IEEE Trans. Image Process.*, vol. 21, no. 11, pp. 4534–4543, 2012.
- [11] J. Mairal, M. Elad, and G. Sapiro, "Sparse representation for color image restoration," *IEEE Trans. Image Process.*,

- vol. 17, no. 1, pp. 53–69, 2008.
- [12] M. Protter and M. Elad, “Image sequence denoising via sparse and redundant representations,” *IEEE Trans. Image Process.*, vol. 18, no. 1, pp. 27–35, 2009.
- [13] L. Ma, L. Moisan, J. Yu, and T. Zeng, “A dictionary learning approach for poisson image deblurring,” *IEEE Trans. Medical Imaging*, March 2013.
- [14] C. Jung, L. Jiao, H. Qi, and T. Sun, “Image deblocking via sparse representation,” *Signal Process. Image Commun.*, vol. 27, no. 6, pp. 663 – 677, 2012.
- [15] M. Aharon, M. Elad, and A. Bruckstein, “The K-SVD: An algorithm for designing of overcomplete dictionaries for sparse representation,” *IEEE Trans. Signal Process.*, vol. 54, no. 11, pp. 4311–4322, 2006.
- [16] A. Chambolle and T. Pock, “A first-order primal-dual algorithm for convex problems with applications to imaging,” *J. Math. Imaging Vision*, vol. 40, no. 1, pp. 120–145, 2011.
- [17] Y. C. Pati, R. Rezaifar, and P. S. Krishnaprasad, “Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition,” in *Conf. Rec. 27th Asilomar Conf. Signals, Syst. Comput.*, vol. 1, 1993.
- [18] E. Esser, X. Zhang, and T. F. Chan, “A general framework for a class of first order primal-dual algorithms for convex optimization in imaging science,” *SIAM J. Imaging Sci.*, vol. 3, no. 4, pp. 1015–1046, 2010.
- [19] A. Chambolle, “An algorithm for total variation minimization and applications,” *J. Math. Imaging Vision*, vol. 20, pp. 89–97, 2004.
- [20] J. Kornblum, “Using jpeg quantization tables to identify imagery processed by software,” in *Proceedings of the Digital Forensic Workshop*, 2008.
- [21] R. Rubinstein, “KSVD-Box,” <http://www.cs.technion.ac.il/~ronrubin/software.html>.
- [22] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, “Image quality assessment: From error visibility to structural similarity,” *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, 2004.



Fig. 3. The first column: uncompressed images of size 256×256 ; the second column: JPEG decompressed images (the quality index for all images is $q = 50/3$, and the compression ratios are 9:1, 8:1, 18:1, 15:1, 15:1, 24:1, and 20:1 from top to bottom); the third column: decompressed images by the TV method; the fourth column: decompressed images by the WTV method; the fifth column: decompressed images by the Dic method; the sixth column: decompressed images by the DicTV method.

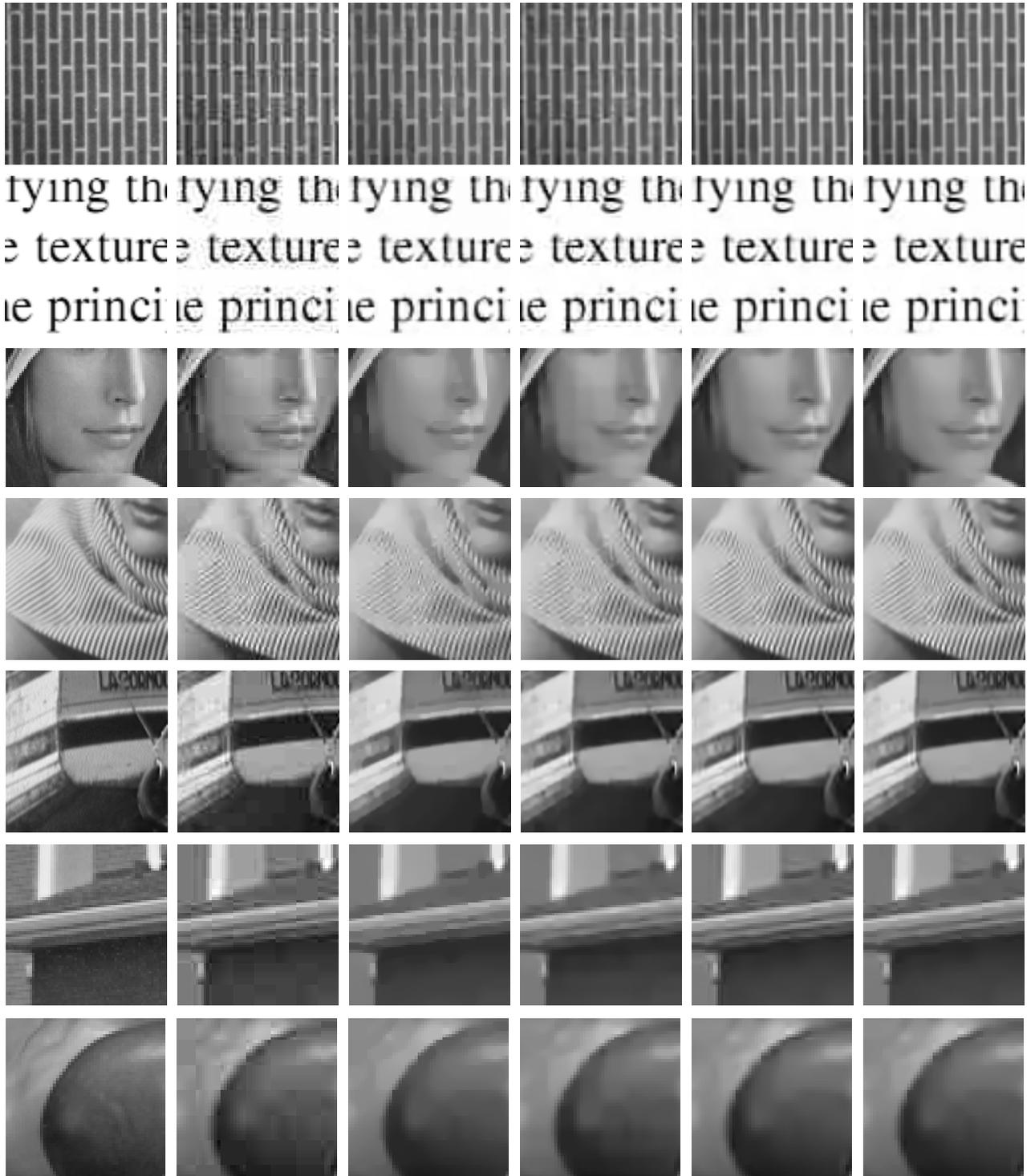


Fig. 4. Zoomed parts of images in Fig. 3. The first column: zoomed part of uncompressed images of size 256×256 ; the second column: zoomed part of JPEG decompressed images; the third column: zoomed part of decompressed images by the TV method; the fourth column: zoomed part of decompressed images by the WTV method; the fifth column: zoomed part of decompressed images by the Dic method; the sixth column: zoomed part of decompressed images by the DicTV method.