4 issues/year

Electronic access

- link.springer.com

Subscription information

- springer.com/librarians

International Journal on Document Analysis and Recognition (IJDAR)

Editors-in-Chief: K. Kise; D. Lopresti; S. Marinai

- **Sponsored by the International Association for Pattern Recognition.**
- **Covers all areas related to document analysis and recognition.**
- **Includes contributions dealing with computer recognition of characters, symbols, text, lines, graphics, images, handwriting, and signatures.**
- **Examines automatic analyses of the overall physical and logical structures of documents.**

Sponsored by the International Association for Pattern Recognition, this journal is focused on publishing articles that cover all areas related to document analysis and recognition. This includes contributions dealing with computer recognition of characters, symbols, text, lines, graphics, images, handwriting, signatures, as well as automatic analyses of the overall physical and logical structures of documents, with the ultimate objective of a high-level understanding of their semantic content.

The International Journal on Document Analysis and Recognition (IJDAR) publishes articles of four primary types: original research papers, correspondence, overviews and summaries, and system descriptions. It also features special issues on active areas of research.

Currently indexed in:

Academic Search Alumni Edition, Academic Search Complete, Academic Search Premier, Bibliography of Linguistic Literature, Compendex, Compuscience, Computer Science Index, Current Abstracts, Current Contents/Engineering, Computing, and Technology, DBLP, Google, INSPEC, Journal Citation Reports/Science Edition, OCLC ArticleFirst Database, OCLC FirstSearch Electronic Collections Online, PASCAL, SCOPUS, Science Citation Index Expanded, Summon by Serial Solutions, TOC Premier.

Impact Factor: 0.846 (2018), Journal Citation Reports®

On the homepage of **International Journal on Document Analysis and Recognition (IJDAR)** at **springer.com** you can

- Sign up for our Table of Contents Alerts
- Get to know the complete Editorial Board
- Find submission information

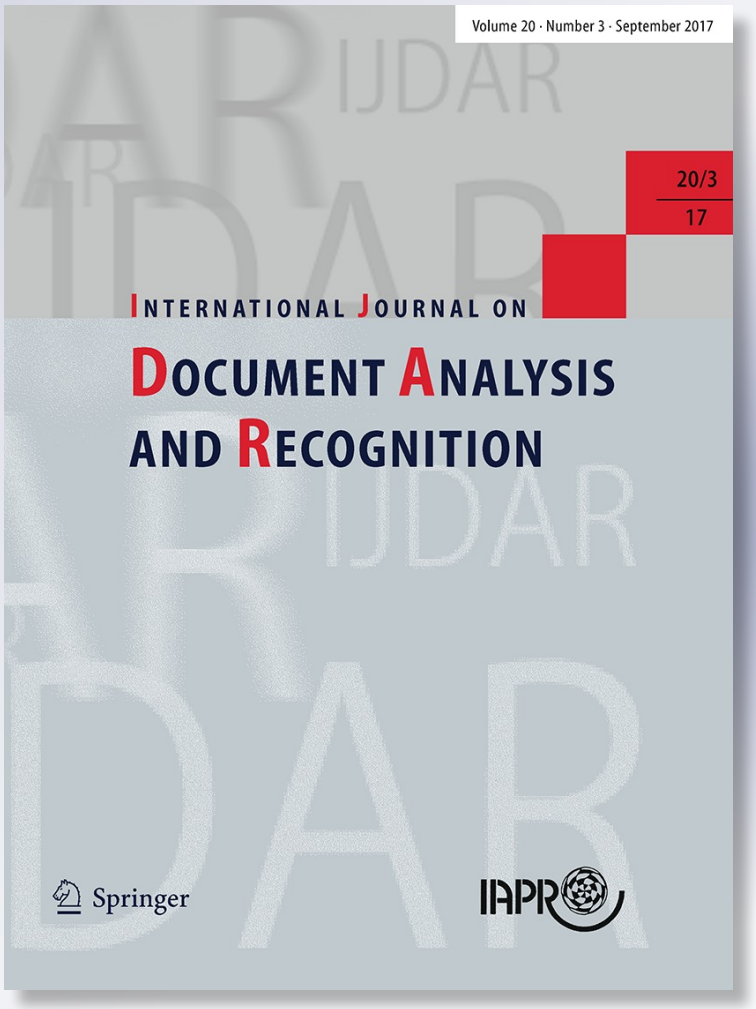
Post-processing coding artefacts for JPEG documents

The-Anh Pham & Mathieu Delalandre

International Journal on Document Analysis and Recognition (IJDAR)

ISSN 1433-2833
Volume 20
Number 3

IJDAR (2017) 20:189-200
DOI 10.1007/s10032-017-0288-4



Springer

IJDAR (2017) 20:189–200
DOI 10.1007/s10032-017-0288-4

ORIGINAL PAPER

Post-processing coding artefacts for JPEG documents

The-Anh Pham[1] · **Mathieu Delalandre**[2]

Received: 30 June 2016 / Revised: 22 May 2017 / Accepted: 24 June 2017 / Published online: 29 June 2017

Abstract Coding artefacts, including ringing and blocking artefacts, are often introduced when document images are compressed using the JPEG standard. These artefacts severely impact visual perception of the image content. Although a number of methods have been presented to deal with coding artefacts, most of them are dedicated to natural images; few works have investigated to work on document content. The current work is an attempt to fill this lack. In contrast to all the approaches taken by previous works, we propose to post-process the coding artefacts by estimating the quantization noise, which is not available on the decoder's side. The estimated noise is then used to reconstruct the image with better quality. A number of experiments were conducted to show the efficiency of the proposed method in comparison with the state-of-the-art methods.

Keywords Compression artefacts · Artefact post-processing · Document decompression

1 Introduction

The JPEG standard [26] has been designed for colour natural images, and most of the methods proposed to process JPEG artefacts are dedicated to them [6,9,13–15,29,30,33]. Document images are mostly composed of background/foreground regions, and the transform coding, as used in JPEG, is unable to process them properly [31]. As a result, lossless compression techniques and formats (e.g. TIFF, PNG, BMP) are mainly recommended in the technical literature for document storage [24]. However, despite the little ability of JPEG and transform coding to preserve the document image content, the JPEG format still constitutes a common target to design document analysis application [10,12,16,28]. This can be explained by two main reasons: (i) the lack of alternatives to the standard transform coding methods for lossy document image compression [3] and (ii) the desirable properties of the lossy compression methods, such as JPEG and JPEG 2000, against lossless compression methods for document images [25] (e.g. control of the compression rate/quality, standard-compliant format, low computational cost).

At low-bit-rate coding, JPEG-encoded images are subject to heavy distortion by blocking and ringing artefacts. Basically, blocking artefact refers to discontinuities of pixel values along block boundaries due to the heavy quantization of the transformed coefficients. On the other hand, ringing artefact refers to the adding of spurious detail along the sharp transitions of the image (i.e. edge locations). Transform coding methods produce very efficient representations of the low-frequency information but cause rough approximation of the high-frequency components such as edges. This matter is of particular importance for human visual perception of decoded document images, because document content is mostly composed of sharp edges such as text and graphics/diagrams.

To address these issues, the design of dedicated methods to process JPEG artefacts with document images constitutes an alternative solution. However, there has been little effort in the literature to deal with this aspect [16,18,20,28]. The lack of efficient methods for decompressing JPEG document images has motivated us to carry out this work. In

✉ The-Anh Pham
phamtheanh@hdu.edu.vn

Mathieu Delalandre
mathieu.delalandre@univ-tours.fr

[1] Hong Duc University, Thanh Hoa City, Vietnam

[2] Computer Science Lab, 64 Avenue Jean Portalis, 37200 Tours, France

Table 1 Notation and descriptions

Notation	Description
$\mathcal{T}$	Forward discrete cosine transform (DCT)
$\mathcal{T}^{-1}$	Inverse DCT
$f(x, y)$	An 8×8 image block in the spatial domain, $x, y \in \{0, \ldots, 7\}$
$\hat{f}(x, y)$	Estimate of $f(x, y)$
$Q(u, v)$	An 8×8 quantization matrix, $u, v \in \{0, \ldots, 7\}$
$F(u, v)$	DCT coefficient block, $F(u, v) = \mathcal{T}(f(x, y))$
round(s)	Round s to the nearest integer
rclip(s)	Round s to the nearest integer and then clip the result into $[0, 255]$
$F_q(u, v)$	Quantized coefficients, $F_q(u, v) = \text{round}\left(\frac{F(u,v)}{Q(u,v)}\right)$
$F_d(u, v)$	Dequantized coefficients, $F_d(u, v) = F_q(u, v)Q(u, v)$
$\hat{F}(u, v)$	Correction of dequantized coefficients, at beginning $\hat{F}(u, v) = F_d(u, v)$
$F_n(u, v)$	Quantization noise, $F_n(u, v) = F(u, v) - F_q(u, v)Q(u, v)$
$\hat{F}_n(u, v)$	Estimate of $F_n(u, v)$
$\hat{f}_n(x, y)$	Noise caused by rclip() operator

contrast to the approaches taken in the previous attempts [16,18,20,28], we propose in this paper a new approach that handles coding artefacts by estimating the quantization noise, which plays a crucial role for improving the visual quality of decoded images. Experiments showed that the proposed method provides an encouraging improvement in image quality compared with other state-of-the-art methods.

For the rest of this paper, we provide a review of related work in Sect. 2. Our approach is presented in Sect. 3. Experimental results are detailed in Sect. 4. Finally, Sect. 5 concludes the paper and gives several lines for future work. For clarity of presentation, we use the notation defined in Table 1 throughout the paper.

2 Related work

A large number of methods have been presented in the literature to handle JPEG coding artefacts. We categorize them by domain (i.e. natural images versus document content) and then by the approach taken in these methods. In what follows, we first describe the basic steps of the JPEG coding scheme and then review the most representative methods for JPEG artefact post-processing.

2.1 JPEG coding scheme

In the JPEG coding scheme [26], an input image is partitioned into non-overlapping 8×8 blocks, each of which is then individually compressed using a process pipeline consisting of the following steps: discrete cosine transform (DCT), quantization, and entropy coding. The first step of the DCT aims at removing spatial redundancy from the input image. Next, the quantization step produces a compact representation of the coefficients by dividing them by pre-defined constants (i.e. quantization values) and then rounding the results to the nearest integer. The last step creates an organization of the quantized coefficients in such a way that the length of the encoded bit stream is minimized to be efficiently transmitted via a network or simply stored in an external file. On the decoder's side, we apply these steps in an inverse manner to reconstruct the image. Figure 1 illustrates the workflow of both the encoding and decoding algorithms [16,26].

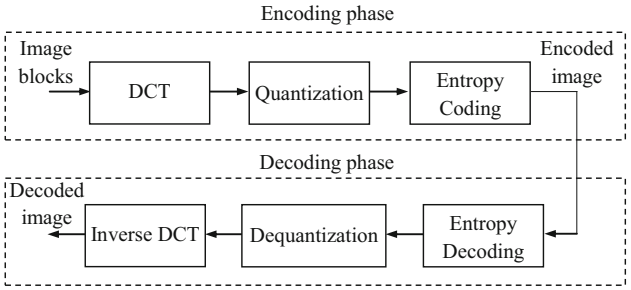


Fig. 1 Workflow of the JPEG encoding and decoding algorithms

Mathematically, the DCT coefficients $F(u, v)$ of an image block $f(x, y)$ are defined as follows:

$$F(u, v) = \mathcal{T}(f) = \frac{e(u)e(v)}{4} \sum_{x=0}^{7} \sum_{y=0}^{7} f(x, y)C(x, u)C(y, v), \quad (1)$$

where $e(s) = \begin{cases} \frac{1}{\sqrt{2}} & \text{if } s = 0 \\ 1 & \text{otherwise} \end{cases}$ and

$$C(m, n) = \cos\left(\frac{(2m + 1)n\pi}{16}\right).$$

The inverse DCT (IDCT) is accordingly defined to recover the original image by:

$$f(x, y) = \mathcal{T}^{-1}(F) = \frac{1}{4} \sum_{u=0}^{7} \sum_{v=0}^{7} e(u)e(v)F(u, v)C(x, u)C(y, v) \quad (2)$$

In the JPEG encoding algorithm, the quantization step takes as an input a quality parameter q ($1 \leq q \leq 100$) with the sense that the higher the value of q, the better the reconstructed image. Setting $q = 1$ corresponds to the worst case, in which much of the image detail is lost, while $q =$

(a) (b)

Fig. 2 Ringing artefact in JPEG coding algorithm ($q = 9$): **a** original text; **b** reconstructed text

100 results in very good image quality for the cost of a low compression rate. Usually, q is set in the range of [40, 80], which often produces a desirable compromise between image quality and compression rate.

The parameter q is used to construct a quantization matrix $Q(u, v)$, which is then used to quantize the DCT coefficients. In the JPEG scheme, $Q(u, v)$ is fixed for both encoder and decoder for a given q. Specifically, the quantized coefficients $F_q(u, v)$ are created as follows:

$$F_q(u, v) = \text{round}\left(\frac{F(u, v)}{Q(u, v)}\right), \tag{3}$$

where round$(\cdot)$ denotes the rounding operator.

In other words, we can establish the following relation in the quantization process:

$$F(u, v) = F_q(u, v)Q(u, v) + F_n(u, v), \tag{4}$$

where $F_n(u, v)$ is called the quantization noise, and $F_n(u, v) \in [-\frac{Q(u,v)}{2}, \frac{Q(u,v)}{2}]$ because of the rounding effect [16].

Because only $F_q(u, v)$ is transmitted to the decoder, the decoding algorithm reconstructs the image $\hat{f}(x, y)$ in the following manner:

$$\hat{f}(x, y) = \text{rclip}\left(\mathcal{T}^{-1}\left(\hat{F}(u, v)\right)\right), \tag{5}$$

where $\hat{F}(u, v) = F_d(u, v) = F_q(u, v)Q(u, v)$, and rclip$(s)$ rounds s to the nearest integer and then clips the result into the image intensity range of [0, 255].

Although the JPEG coding algorithm produces very high compression rates, the decoded images are subject to severe distortion by blocking and ringing artefacts. These artefacts can seriously impact human visual perception as shown in Fig. 2. In considerations of existing approaches for handling these artefacts, much attention has been paid to natural images, while little effort has been devoted to document images. These approaches are reviewed in the following sections.

2.2 Artefact post-processing for natural images

For natural images, the common approaches include maximum a posteriori estimation (MAP) [28,29,32], projection onto convex sets (POCS) [15,30,33], sparse representation based on a learned dictionary (SRLD) [9,13,14], and total variation (TV) regularization [6–8]. Table 2 presents the main characteristics of these methods.

Table 2 Characterization of different JPEG decoding approaches for natural images

Approach	Original signal model
MAP	Gibbs [28,29], BS-PM [32]
POCS	Neighbouring constraint sets [15,30,33]
TV	Gradient magnitude sum [6–8]
SRLD	Learned dictionaries [9,13,14,22,23]

Given an observed image Y (i.e. the image decompressed by the baseline JPEG), the MAP-based approach solves the inverse problem of finding the original image X that corresponds to the maximum a posteriori probability $P(X|Y)$. In doing so, different models have been employed to account for the prior distribution $P(X)$, such as the Gibbs model [28,29] and the block similarity prior model (BS-PM) [32]. After building the prior models, image reconstruction is done by an iterative process that consists of two steps: updating the latent variables and sanity checking based on the quantization constraint. Because the optimization algorithm must operate on both the spatial and frequency domains for all the image blocks of the input image, computational complexity is one of the main issues of the MAP-based methods. In addition, the choosing of an appropriate model to represent the prior distribution of the original coefficients is not trivial.

In contrast to the MAP-based approach, the POCS-based methods [15,30,33] work by building a set of constraints, each of which is described by a closed convex set. The original image is then estimated as the intersection of these convex sets. The POCS-based methods are traditionally subject to intensive computational overhead and to a variety of parameters used to define the constraints.

Sparse representation is also a promising approach for handling JPEG artefacts [9,13,14]. The basic idea is to construct a dictionary consisting of the basis vectors such that an input image patch can be represented by a few vectors from the dictionary. The dictionary can be learned using a training dataset consisting of noise-free image patches [13,14] or using the observed image itself [9,13]. The denoising process is then performed by minimizing an objective function. The K-singular value decomposition (K-SVD) algorithm [1] and the orthogonal matching pursuit (OMP) algorithm [19] are often applied for both dictionary learning and image denoising. One of the benefits of the K-SVD algorithm is that it has the capability of discarding noise content from the corrupted image when learning the dictionary [1,13].

Other sparsity-based techniques based on estimation of the quantization matrix have been studied in [22]. In this

Table 3 Characterization of different decoding approaches for JPEG document images

Method	Text model	Non-text model	Treated artefacts	Time cost
[28]	Bimodal function	GMRF model	Ringing and blocking	High
[18]	Laplacian model	Gaussian model	Ringing and blocking	High
[16]	Not applicable	Binarization	Ringing	Low
[20]	Bimodal Laplacian	Total block variation	Ringing and blocking	Low

work, the quantization matrix on the decoder's side is estimated by using the first few singular vectors obtained from a singular value decomposition process. The estimated quantization matrix can be used to set parameters for compression artefact reduction [23].

The last well-known approach is based on total variation (TV) regularization, which has been widely applied for dealing with JPEG coding artefacts [6–8]. Given an observed image Y, the key idea of a TV-based method is to look for a signal X which has lower variation and is not very far from Y. This involves minimizing a proper objective function that is often composed of a regularization term and a data fitting term. Generally, the TV-based approach gives quite good visual quality improvement, but its performance is highly dependent on the choice of value for the parameter λ, which is often varied from one image to another.

2.3 Artefact post-processing for document content

Table 3 presents the main decoding approaches for document images. A typical compound document can be composed of different content such as background, text, and/or graphics. Therefore, the first step in these approaches is to classify the image blocks into different groups such as background, text/graphics, and picture blocks. Image reconstruction is then performed separately for each type of block. It is worth highlighting that all the methods discussed in the present material are completely different from what goes under the name 'DjVu', such as [4]. Specifically, all the methods presented in the current paper can be considered as post-processing techniques that are applied on the decoder's side after the input images have already been compressed. Hence, the decoder has no information in advance about the original image. In contrast, the DjVu method falls into the category of a pre-processing scheme and hence has complete prior knowledge about the clean signal, enabling it to perform both rate-distortion optimization on the encoder's side as well as post-processing on the decoder's side.

One of the most notable attempts for improving the visual quality of JPEG documents is presented in [28]. In this work, the authors proposed using the Gaussian Markov random field (GMRF) model to represent the background blocks, while the text and graphics blocks are represented as a bimodal function that accounts for the two dominant intensities of these blocks. Artefact reduction is then performed by using a MAP-based algorithm. Their experimental results showed a significant improvement of visual quality compared with the baseline JPEG.

In [16], a different approach is presented, in which the prior knowledge of the DCT coefficient distribution is exploited. Specifically, the authors suggested using the Laplacian and Gaussian models to represent the DCT coefficient distribution of text blocks and pictorial blocks, respectively. These models are then used to compute the centroids of the code blocks, from which the reconstruction of the original image is performed by shifting the dequantized coefficients to these centroids. Experimental results showed a slightly better result when compared with the baseline JPEG algorithm. The highest improvement in peak signal-to-noise ratio (PSNR), for example, for a simple text image is just 0.4 (dB) when the image is compressed at a quite high quality ($q = 40$). In addition, the estimation of parameters for the two models is done based on the quantized DCT data and is not sufficiently accurate.

The last method noted is presented in [18] and specifically addresses the ringing artefact. It is based on the observation that the ringing artefact is more dominant in background regions than in text regions. Therefore, the first step is to segment the image into foreground and background regions by using the Otsu technique [17]. Ringing cleansing is then performed by changing the values of all the noisy pixels in the background regions to the same value, which is that estimated to be the most frequent grey level of the background. In addition, a simple morphological operator is applied to prevent cleansing on the pixels in proximity to the text's edges. Consequently, the text pixels are not subject to ringing reduction.

Recently, an advanced document decoder was presented in [20], which proposed decoding background blocks directly in the transform domain, while the text blocks are efficiently decoded by minimizing the total probability entropy of the image content. This total probability entropy function was designed to account for the error cost of making the decision for each pixel (i.e. background or foreground). Promising results were reported in the paper, and the proposed method is very time-efficient.

Concluding remarks: There has been much effort reported in the literature towards improving the quality of encoded JPEG images, but most of these methods are dedicated to natural images and are subject to high computational

complexity. To the best knowledge of the authors, few methods have been presented to address the same problem for document content. The most notable work [28] provides a significant improvement of visual quality, but it is too costly. The recent document decoder [20] works very well for low-bit-rate document compression but is less effective at intermediate and higher bit rates. On the other hand, simple computation methods, such as [16,18], do not give sufficiently satisfactory results. In this work, our aim is to propose a novel method for improving the quality of JPEG document images when compressed at intermediate and higher bit rates. In what follows, we describe the proposed approach in detail.

3 Post-processing of JPEG coding artefacts

We present in this section our approach for post-processing the JPEG coding artefact. Section 3.1 introduces our approach, and Sect. 3.2 presents our algorithm, which is employed to post-process the JPEG artefacts. For clarity of presentation, we use the notation defined in Table 1 throughout the section.

3.1 The proposed approach

As discussed in Sect. 2 (Related work), because of the omission of quantization noise, the reconstructed image is subject to different coding distortions, mainly blocking and ringing artefacts. The lower the value of parameter q, the more severe the distortion. To address this issue, we propose to estimate the quantization noise for post-processing of the decoded images. In this way, the quality of the JPEG images will be dependent on the accuracy of the quantization noise estimation process. Unfortunately, as we have no prior knowledge of either the quantization noise or the original image, the estimation of quantization noise must rely mainly on the dequantized DCT coefficients $F_d(u, v)$. In the next subsection, we present an algorithm to tackle this task.

The basic idea of our approach is based on the observation that if we know $F_n(u, v)$, we can reconstruct a better quality image $\hat{f}(x, y)$, and vice versa. If $\hat{f}(x, y)$ is given, we can easily compute $F_n(u, v)$ by applying the quantization step (see Eq. 4). The workflow of our approach is depicted in Fig. 3. As in the JPEG document decoders from the literature [16,20,28], our first process is to classify the DCT blocks into text and non-text blocks. For smooth blocks, quantization noise is almost zero or close to zero. Hence, it is unnecessary to handle noise compensation for smooth blocks. For non-smooth blocks, an expectation maximization (EM) process is employed to reconstruct the original text blocks.

To be more specific, if we again perform encoding of the image $\hat{f}(x, y)$ with the same quantization matrix $Q(u, v)$, we would expect to extract some approximation $\hat{F}_n(u, v)$ of

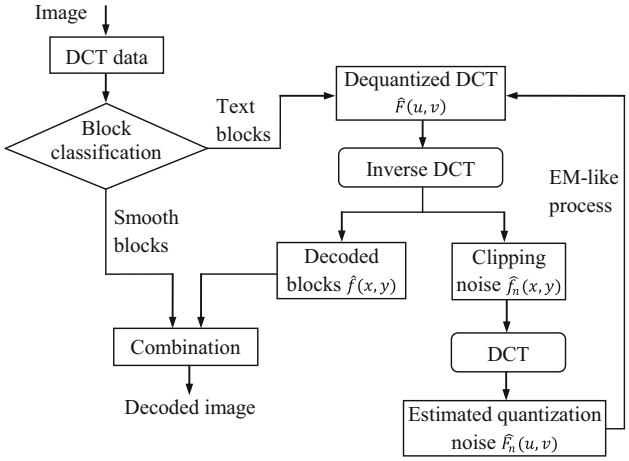


Fig. 3 Workflow of the proposed approach

the quantization noise. This is easily seen because $\hat{f}(x, y)$ is supposed to be close to the original image $f(x, y)$. Hence, $\hat{F}_n(u, v)$ is expected to resemble $F_n(u, v)$. The extracted noise is then used to enhance the quality of the image $\hat{f}(x, y)$. After that, the whole process can be repeated a number of times until the desired result is obtained. Formally, we can derive the following expression from Eq. (5):

$$\hat{f}(x, y) = \mathcal{T}^{-1}\left(\hat{F}(u, v)\right) + \hat{f}_n(x, y), \tag{6}$$

where $\hat{f}_n(x, y)$ denotes the clipping noise caused by the rclip(·) operator. By applying DCT to the left term of (6), we obtain:

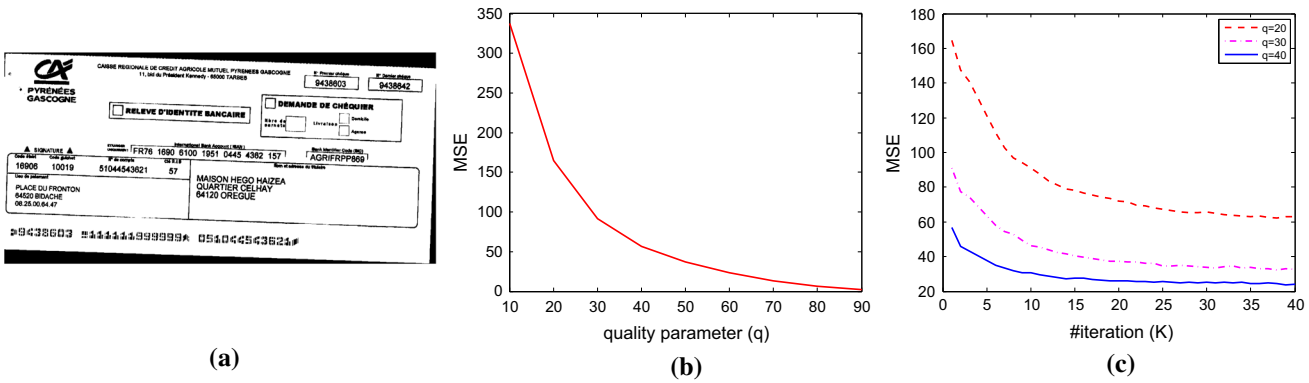
$$\mathcal{T}\left(\hat{f}(x, y)\right) = \mathcal{T}\left(\mathcal{T}^{-1}\left(\hat{F}(u, v)\right) + \hat{f}_n(x, y)\right). \tag{7}$$

Based on the DCT given in Eq. (1), it is straightforward to derive the additive property of the DCT for two functions $f(x, y)$ and $g(x, y)$ as follows:

$$\begin{aligned}\mathcal{T}(f + g) &= \frac{e(u)e(v)}{4}\sum_{x=0}^{7}\sum_{y=0}^{7} f(x, y)C(x, u)C(y, v) \\ &\quad + \frac{e(u)e(v)}{4}\sum_{x=0}^{7}\sum_{y=0}^{7} g(x, y)C(x, u)C(y, v) \\ &= \mathcal{T}(f) + \mathcal{T}(g).\end{aligned} \tag{8}$$

Hence, we can simplify Eq. (7) to the following expression:

$$\begin{aligned}\mathcal{T}\left(\hat{f}(x, y)\right) &= \mathcal{T}\left(\mathcal{T}^{-1}\left(\hat{F}(u, v)\right)\right) + \mathcal{T}\left(\hat{f}_n(x, y)\right) \\ &= \hat{F}(u, v) + \mathcal{T}\left(\hat{f}_n(x, y)\right) \\ &= \hat{F}(u, v) + \hat{F}_n(u, v).\end{aligned} \tag{9}$$



(a) (b) (c)

Fig. 4 **a** A test image used in our study; **b** mean square error (MSE) between original quantization noise and estimated noise against the coding quality; **c** MSE between original quantization noise and estimated noise against parameter K for three coding qualities

Here, we denote $\hat{F}_n(u, v) = \mathcal{T}\left(\hat{f}_n(x, y)\right)$ as the estimate of quantization noise.

Combining (4) and (9), we can compute the mean square error (MSE) between the original image and the reconstructed one in the frequency domain as follows[1]:

$$\begin{aligned} \text{MSE}\left(\mathcal{T}(f), \mathcal{T}(\hat{f})\right) &= \frac{1}{N}\|\mathcal{T}(f) - \mathcal{T}(\hat{f})\|_2^2 \\ &= \frac{1}{N}\|F_n - \hat{F}_n\|_2^2 \\ &= \text{MSE}\left(F_n, \hat{F}_n\right), \end{aligned} \tag{10}$$

where N is the total number of samples in f. Here, MSE is employed to simply justify the difference between the original image and the reconstructed one. It is a common score and is closely related to the peak signal-to-noise ratio (PSNR) [5,16]. According to Parseval's theorem and because of the unitary property of the DCT [5,16,21], MSE in the transformed domain behaves in the same manner as it does in the spatial domain. Hence, the closer $\hat{f}$ is to f, the lower the MSE is. As a result, the estimated noise $\hat{F}_n$ will approach the original noise F_n depending on how close $\hat{f}$ is to f. When the original image is compressed at very low bit rates (e.g. < 0.1 bpp), the reconstructed image $\hat{f}$ would be far from f, and thus, the estimate of the quantization noise would be not sufficiently accurate. However, for higher bit rates, the decoded image is supposed to be close to the original one. Consequently, we can make use of this fact to estimate $\hat{F}_n$ as the compensation of quantization noise for the decoder.

Figure 4b shows the fitness of quantization noise estimation as an MSE function of coding quality (i.e. parameter q). We compute the MSE between F_n and $\hat{F}_n$ by using Eq. (10). As expected, the error greatly decreases as parameter q increases. Given a sufficiently high value of q ($q > 40$, for example), we can assume that the estimated noise $\hat{F}_n$ is accurate enough for optimizing the decoding process. In practice, the proposed algorithm performs very well even for much lower values of q (e.g. $q = 20$) as we will see in the experimental section.

3.2 The post-processing algorithm

To support our approach, we propose a three-step algorithm as shown in Fig. 3, including block classification, text block processing, and combination. The block combination step results in a simple combination of text and smooth blocks in the spatial domain. We detail here the block classification and the text block treatment.

Block classification: Compound documents can be composed of text, graphics, pictures, and background information. Optimal algorithms for decompression of JPEG documents [16,28] often perform a block classification step to cluster the image blocks into different categories such as text blocks, pictorial blocks, and background blocks. In this work, we simply classify the image blocks into two classes: smooth blocks (i.e. areas of highly correlated information) and non-smooth blocks (i.e. text blocks, graphics, pictures). Artefact filtering is carried out only for non-smooth blocks. For block classification, we employ a simple and efficient criterion based on AC energy, which is computed as the sum of the squares of the AC coefficients of the block [11].

For a smooth block, the AC energy is fairly low because most of the AC coefficients are zero. In contrast, the AC energy of a non-smooth block is relatively high. Consequently, block classification is done by thresholding the AC energy. The block classification thus turns into a traditional binary classification problem with a precision/recall evaluation protocol. In our context, it is preferable not to miss the true text/graphics blocks (i.e. true positives) because these blocks account for the foreground content that characterizes the main information of a document. It does not matter if some smooth blocks are misclassified as text/graphics blocks

[1] The indexes are removed for simplification.

(i.e. false positives) except that this will increase the computational overhead.

Post-processing of text blocks: As explained previously, the treatment of text blocks is performed in an EM fashion in which the two variables $\hat{f}(x, y)$ and $\hat{F}_n(u, v)$ recursively update each other. During each iteration, each one updates and enhances the other. The process is repeated a specified number of times or until a satisfactory result is obtained. This process takes as an input the quantized data $F_q(u, v)$ of each non-smooth block. Its output is the decompressed image with higher quality. It is assumed that the quantization matrix $Q(u, v)$ is known at both encoding and decoding. The main step of the process is outlined as follows.

For each non-smooth block:

1. Initialization:
   - Set $\hat{F}_n^{(0)}(u, v) = 0$ for each $(u, v) \in [0, 7] \times [0, 7]$.
   - Set $\hat{F}^{(0)}(u, v) = F_d(u, v) = F_q(u, v)Q(u, v)$.
2. Loop: for each iteration $t = 1, \ldots, K$:
   - Update $\hat{f}(x, y)$ for given $\hat{F}_n(u, v)$ using Eq. (5):

$$\hat{f}^{(t)}(x, y) = \text{rclip}\left(\mathcal{T}^{-1}\left(\hat{F}^{(0)}(u, v) + \hat{F}_n^{(t-1)}(u, v)\right)\right).$$

   - Update $\hat{F}_n(u, v)$ for given $\hat{f}(x, y)$ using Eq. (9):

$$\hat{F}_n^{(t)}(u, v) = \mathcal{T}\left(\hat{f}^{(t)}(x, y)\right) - \hat{F}^{(t)}(u, v),$$

   where

$$\hat{F}^{(t)}(u, v) = \text{round}\left(\frac{\mathcal{T}\left(\hat{f}^{(t)}(x, y)\right)}{Q(u, v)}\right) Q(u, v).$$

3. Output: $\hat{f}^{(K)}(x, y)$ is the optimal decoding block.

Here, it is unnecessary to explicitly compute the estimated noise by $\hat{F}_n(u, v) = \mathcal{T}\left(\hat{f}_n(x, y)\right)$. Instead, $\hat{F}_n^{(t)}(u, v)$ is computed by performing requantization of the currently fitting DCT data to save processing time in the iteration process. Figure 4c shows the fitness of quantization noise estimation as an MSE function of the parameter K (i.e. the number of iterations). We compute the MSE between F_n and $\hat{F}_n$ for all the non-smooth blocks of the input image in Fig. 4a. The parameter q is set to 20, 30, and 40 in this test. As can be seen, the MSE for all three cases is reduced quickly at early iterations (i.e. $K < 10$) and gradually decreased after that. We can also see that the MSE in the case $q = 40$ is much lower than that in the others (i.e. $q = 20$ and $q = 30$) for all the iterations.

In practice, we can generalize the computation of $\hat{F}^{(t)}(u, v)$ in the decoding algorithm by the following expression:

$$\hat{F}^{(t)}(u, v) = \text{round}\left(\frac{\mathcal{T}\left(\hat{f}^{(t)}(x, y)\right)}{\hat{Q}(u, v)}\right) Q(u, v), \tag{11}$$

where $\hat{Q}(u, v)$ is chosen to be close to $Q(u, v)$. Note that in the JPEG scheme [26], the quantization matrix $Q(u, v)$ is defined as a function of parameter q $(1 \leq q \leq 100)$ as follows:

$$Q(u, v; q) = \begin{cases} \text{round}\left(\frac{50Q_0(u,v)}{q}\right) & \text{if } q < 50 \\ \text{round}\left(\frac{2(100-q)Q_0(u,v)+40}{100}\right) & \text{if } q \geq 50 \end{cases}, \tag{12}$$

where $Q_0(u, v)$ is the standard quantization matrix given as follows:

$$Q_0(u, v) = \begin{bmatrix} 16 & 11 & 10 & 16 & 24 & 40 & 51 & 61 \\ 12 & 12 & 14 & 19 & 26 & 58 & 60 & 55 \\ 14 & 13 & 16 & 24 & 40 & 57 & 69 & 56 \\ 14 & 17 & 22 & 29 & 51 & 87 & 80 & 62 \\ 18 & 22 & 37 & 56 & 68 & 109 & 103 & 77 \\ 24 & 35 & 55 & 64 & 81 & 104 & 113 & 92 \\ 49 & 64 & 78 & 87 & 103 & 121 & 120 & 101 \\ 72 & 92 & 95 & 98 & 112 & 100 & 103 & 99 \end{bmatrix}.$$

4 Experimental results

4.1 Experimental settings

4.1.1 Comparison methods

For comparative evaluation, we selected four representative methods dedicated to post-processing JPEG artefacts in the literature. These methods are described in Table 4 and include the classical JPEG decoder [26], the morphological artefact post-processing method [18] ('Mor' for short), the total variation (TV) method [6], and dictionary-based sparse representation ('Dic' for short) [9]. Here, the JPEG decoder serves as a baseline method for all the other systems. The Mor method is specifically designed to work on document images and thus was deemed useful for our subjective comparison. The last two methods (i.e. TV and Dic) were selected because they are considered to be the state-of-the-art methods dedicated to natural images. Hence, it would be interesting to see how well they perform on document images.

As evaluation metrics, peak signal-to-noise ratio (PSNR) and structural similarity (SSIM) [27] were selected for performance evaluation. These metrics have been commonly

Table 4 Methods for comparative evaluation

Method	Description	Code
JPEG [26]	Classical JPEG decoder	C++
Mor [18]	Dedicated to document images	C++
Our method	Dedicated to document images	C++
TV [6]	Dedicated to natural images	MATLAB
Dic [9]	Dedicated to natural images	MATLAB

Table 5 Datasets used for evaluation

No.	Dataset	# Images	Description
1	MAR-Full	293	Biomedical journal documents (300 dpi)
2	MAR-Text	536	Text zones of MAR-Full (300 dpi)
3	MAR-LowRes	293	Low resolution (150 dpi) of MAR-Full
4	ADM-Doc	684	Administrative documents (roughly 200 dpi)

used for quality assessment of JPEG artefact post-processing; see [6,9,16,28] for examples. Detailed descriptions of these metrics are given in their corresponding references.

4.1.2 Datasets

Table 5 describes the four datasets used for our experiments. We selected a public dataset, namely Medical Archive Records (MAR), from the U.S. National Library of Medicine.[2] This dataset is composed of 293 real documents that have been scanned at 300-dpi resolution in TIF format, covering different types of biomedical journals. The average image size is 2544 × 3296. A common property of these images is known as an unbalanced distribution between background (homogeneous regions) and foreground (text, graphics, etc.). This may lead to bias in computing the evaluation scores because all of the selected criteria are pixel-based metrics. Therefore, we created another dataset by extracting all the text zones from the full MAR dataset. This process resulted in 536 text zones, each of which is quite balanced in density between foreground and background. For clarity, we denote this dataset as MAR-Text and the original dataset as MAR-Full.

To obtain deep insights into the performance of all the methods, we also created a low-resolution version of the MAR-Full dataset. The goal is to study the robustness of the methods when images are acquired by low-resolution devices. To this end, we converted the images in the MAR-Full dataset to images having a resolution of 150 dpi. The resulting dataset is termed MAR-LowRes. In addition to these public datasets, we selected an internal dataset, ADM-Doc, which is a subset of the Itesoft2 dataset used in [2]. This dataset contains 684 administrative documents scanned at an intermediate resolution of approximately 200 dpi.

There is a computational issue with the TV and Dic methods. Because of the high computational complexity of the optimization algorithms designed in these methods, it is too costly to run them on full datasets (e.g. approximately 40–50 min for decompressing an image of size 2544×3296). Therefore, we were not able to run these methods on all the datasets. Instead, these two methods were evaluated on a small subset consisting of 10 images randomly selected from the MAR-Text dataset. The number of iterations was set to 50 for both the TV and the Dic methods, while the other parameters were kept at their defaults.

4.1.3 Parameter settings

There are two parameters used in our algorithm. The first, threshold T_{AC}, is used to classify a block as a smooth or non-smooth block. In our experiments, we set $T_{\mathrm{AC}} = 25$, although it was found from our observations that any value of T_{AC} in [5, 50] will give little change in performance. The second threshold, K, is the number of iterations used in our post-processing algorithm. Setting a high value for K will produce very good decoding quality at the cost of raising computational overhead. Here, we set $K = 15$, and higher values of K ($K > 20$) will also be considered in order to determine the marginal performance of the proposed method. For each dataset, we performed compression and decompression for varying values of parameter q in the range of $\{10, 15, 20, 25, 30, 35, 40, 45\}$. The final scores were then averaged via the quality parameter q. For each value of q, the quantization matrix was generated as $Q(u, v; q)$ by using Eq. (12), and then the corresponding $\hat{Q}(u, v)$ was chosen by $\hat{Q}(u, v) = Q(u, v; q + 0.5)$ to compute the estimated quantization noise.

4.2 Results and discussion

Figure 5 shows the PSNR scores on four datasets for three methods: our method, Mor, and JPEG. The first remark we can make here is that all three methods behave quite similarly on all the datasets except MAR-LowRes. Specifically, the proposed method performs best on all the tests with a significant gap of visual quality (i.e. PSNR) compared with the JPEG standard and the Mor method. On average, our method, for example, gives a PSNR improvement of 6.2685 (dB), whereas the Mor method offers an improvement of 2.4301 (dB) over the baseline JPEG when each is applied to

[2] https://www.nlm.nih.gov/.

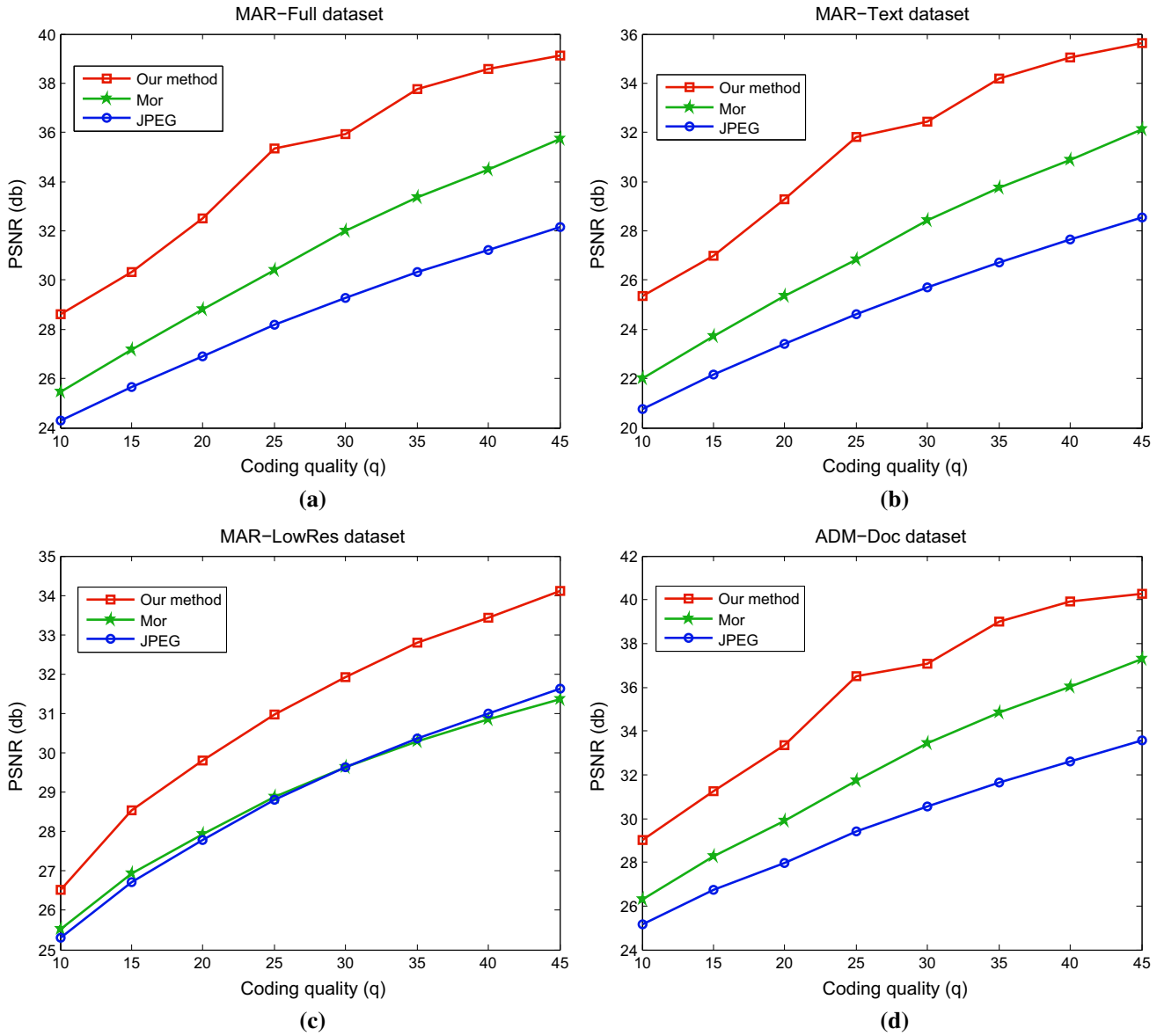


Fig. 5 PSNR scores of three methods (our method, Mor, and JPEG) on four datasets

the MAR-Full dataset. The same observation can be deduced from the results on the MAR-Text and ADM-Doc datasets.

As briefly noted previously, the performance of the Mor method is significantly decreased when applied to the MAR-LowRes dataset (Fig. 5c). This time, the Mor method is roughly in the same PSNR interval as the baseline JPEG. On the other hand, the proposed method still outperforms the others up to about 2.1 (dB) on average. The reason behind this situation is attributed to the fact that when the images are down-scaled, the resulting images are distorted by resizing effects (e.g. grey values around the text edges due to interpolation, blurriness). Although the proposed method is impacted in part by these additional distortions, it still maintains a high level of performance compared with the other methods. This confirms the robustness and efficiency of the proposed method.

There is a slight difference in the results between the MAR-Text and MAR-Full datasets (Fig. 5a, b). The average PSNR scores of the three methods on MAR-Text is approximately 3.8 (dB) lower than those for the MAR-Full dataset, although the former dataset is strictly extracted from the latter dataset. The main reason is the unbalanced distribution between foreground and background for the images in the MAR-Full dataset. When applied to the MAR-Text dataset, the image content is quite balanced between text and background, while the artefacts are focused on the transitions of background and foreground. The PSNR scores are thus less impacted by the background information.

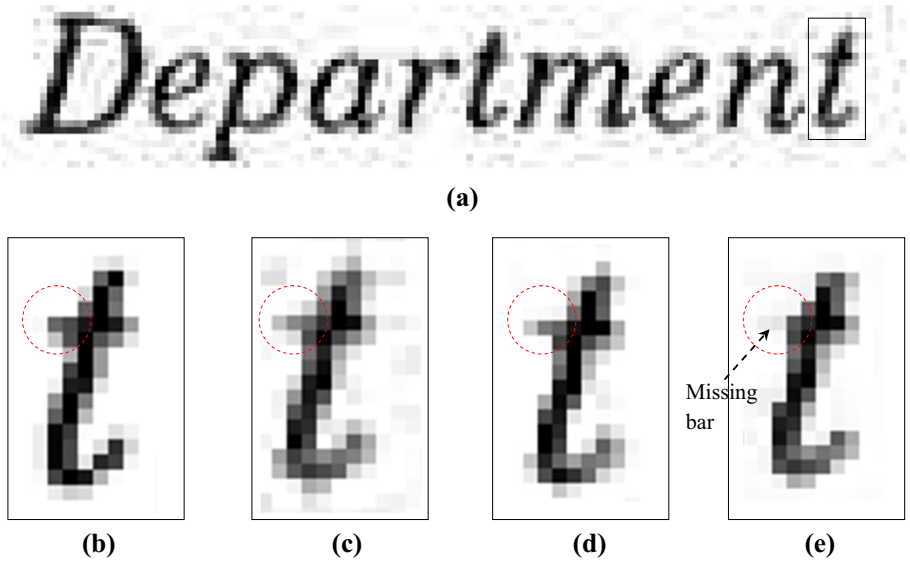


Fig. 6 Visual results of three methods: **a** JPEG-encoded text image ($q = 20$); **b** original text for the portion in (**a**); **c** magnified version of JPEG result; **d** result using our method; **e** result using Mor method

Figure 6 illustrates a visual example of the results obtained by the three methods. Figure 6a shows a text image that is decoded by the JPEG standard with a coding quality of $q = 20$. Figure 6b, c shows the magnified versions of the original text and the JPEG decoding result, respectively, for the clipped portion in Fig. 6a. Figure 6d,e shows the results of our method and of the Mor method for the clipped portion, respectively. As can be seen, both the Mor method and our method are able to remove the ringing artefact, and the results obtained are quite close to the original text in Fig. 6b. However, the Mor method tends to remove true edges because of the binarization effect. Figure 6e, for example, shows that the horizontal line of the character '*t*', marked by the dashed circle, has been partially removed, while this edge line is preserved in our result. The same result was also observed for the Mor method in the original paper (Fig. 6a in [18]).

Figure 7 shows a comparison of the PSNR scores of five methods: Mor, JPEG, TV, Dic, and the proposed method. Here, we compute PSNR for a small subset consisting of 10 images randomly selected from the MAR-Text dataset. Basically, the performance of the Mor method, JPEG, and our method is the same as that presented in Fig. 5b. Interestingly, the TV method performs quite well and even gives a slightly better result than the Mor method. On the other hand, the Dic method works less effectively and only slightly outperforms the basic JPEG algorithm. All things considered, the proposed method performs best and gives a substantial PSNR improvement over all the other methods.

Table 6 shows the behaviour of the five methods according to the SSIM metric. In this test, all the methods were evaluated on the MAR-Text dataset, and the SSIM scores were computed at four different coding qualities (i.e. $q \in \{10, 15, 20, 25\}$). In addition, we have provided the results of our method ('Ours' for short) using two different settings

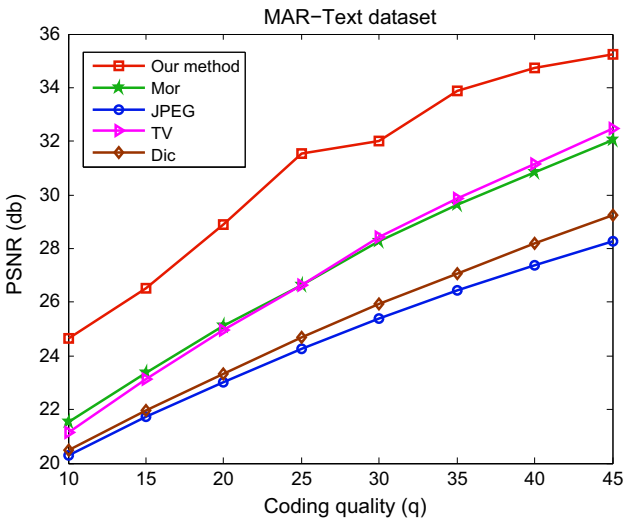


Fig. 7 PSNR scores of five methods: our method, Mor, JPEG, TV, and Dic

for the parameter K (i.e. the number of iterations used in our post-processing algorithm). Generally, the SSIM results are consistent with the PSNR scores as presented previously for all the methods. Specifically, the JPEG decoder is outperformed by the other four methods, whereas the proposed method gives the best results for all the coding qualities. It can be also observed that the performance gaps between Mor, TV, and the proposed method are less than those using the PSNR score. This is partially explained by the intrinsic characteristics of each metric. In addition, the SSIM metric is designed to work on scene or greyscale images. When working on binary images, as in our case, the structural properties of the SSIM metric are not fully exploited. Table 6 also reveals that the proposed method works more effectively with higher values

Table 6 Performance of five methods using structural similarity (SSIM) metric

Method	Coding quality (q)			
	$q = 10$	$q = 15$	$q = 20$	$q = 25$
JPEG [26]	0.8795	0.8972	0.9145	0.9230
Mor [18]	0.9491	0.9635	0.9751	0.9818
TV [6]	0.9423	0.9615	0.9761	0.9827
Dic [9]	0.9093	0.9253	0.9402	0.9536
Ours ($K = 15$)	0.9541	0.9652	0.9784	0.9843
Ours ($K = 35$)	0.9632	0.9765	0.9862	0.9895

Table 7 Post-processing times (ms)

Image size	Mor	TV	Our method	
			$K = 15$	$K = 35$
512×512	14	588	40	100
1600×1200	68	4459	350	780
4200×2800	430	28,257	1220	2580

of parameter K. However, increasing the value of K forces the proposed method to incur an extra overhead of computational cost. Details of this aspect are further investigated in the following subsection.

4.3 Running time analysis

This section provides an evaluation of the processing time of the methods studied. All the tests were run on a CPU machine[3] without parallel implementation. As the TV and Dic methods are implemented in MATLAB, it is difficult to directly compare these methods with the others. In general, these two methods have been known to incur high overhead for computational complexity; see [6,9] for examples. For reference purposes, we have reproduced here the processing time of the TV method (C++ code) reported in the original paper [6]. For our method, we provide the running time for two settings of parameter K (i.e. $K \in \{15, 35\}$) to determine the time complexity of the post-processing algorithm.

The results are presented in Table 7 for different image sizes. As we can see, the Mor method works very efficiently because it is a non-iterative approach. The TV method and our method are more computationally intensive because of the recursive process used in the post-processing algorithm. However, the computational overhead of the proposed method is less than that of the TV method. Our method (with $K = 15$) is roughly 12 times faster than the TV method. This result is encouraging when considering the gain in the visual quality performance of the proposed method.

[3] Windows 7 (64-bit), Intel Core i7-4600U (2.1 GHz), 16 GB RAM.

5 Conclusions

In this paper, we have presented a new approach for post-processing JPEG coding artefacts for document images. The key idea is to produce a compensation for quantization noise when post-processing the JPEG documents. This is done by an expectation maximization algorithm that recursively computes the quantization noise and then reconstructs the image. We have conducted a number of experiments to show the robustness and efficiency of the proposed approach. One of the major advantages of the proposed method is that, while it can to a large extent remove the ringing artefacts, it does not smooth out true edges as other methods do (e.g. the Mor and TV methods). Although the proposed approach works well for images that contain high variation of document content (e.g. binary text and graphics), it is less effective for images that consist of content with little variation, such as complex greyscale or colour document images. This would be an interesting extension of the current work in the future. Furthermore, noise estimation in the DCT space would be a good idea to speed up the iteration process. Finally, extension of this work to colour document images should be investigated as well.

References

1. Aharon, M., Elad, M., Bruckstein, A.: The K-SVD: an algorithm for designing of overcomplete dictionaries for sparse representation. IEEE Trans. Signal Process. **54**(11), 4311–4322 (2006)
2. Alaei, A., Delalandre, M., Girard, N.: Logo detection using painting based representation and probability features. In: International Conference on Document Analysis and Recognition (ICDAR 2013), pp. 1235–1239 (2013)
3. Aung, A., Ng, B.P., Shwe, C.T.: A new transform for document image compression. In: 2009 7th International Conference on Information, Communications and Signal Processing (ICICS), pp. 1–5 (2009)
4. Bottou, L., Haffner, P., Howard, P.G., Simard, P., Bengio, Y., LeCun, Y.: High quality document image compression with 'DjVu'. J. Electron. Imaging **7**(3), 410–425 (1998)
5. Brandão, T., Queluz, M.P.: No-reference image quality assessment based on DCT domain statistics. Signal Process. **88**(4), 822–833 (2008)
6. Bredies, K., Holler, M.: A total variation-based JPEG decompression model. SIAM J. Sci. Comput. **5**(1), 366–393 (2012)
7. Bredies, K., Kunisch, K., Pock, T.: Total generalized variation. SIAM J. Imaging Sci. **3**(3), 492–526 (2010)
8. Chambolle, A.: An algorithm for total variation minimization and applications. J. Math. Imaging Vis. **20**(1–2), 89–97 (2004)
9. Chang, H., Ng, M., Zeng, T.: Reducing artifact in JPEG decompression via a learned dictionary. IEEE Trans. Signal Process. **62**(3), 718–728 (2013)
10. Darwiche, M., Pham, T.A., Delalandre, M.: Comparison of JPEG's competitors for document images. In: 2015 International Conference on Image Processing Theory, Tools and Applications (IPTA 2015), pp. 487–493 (2015)
11. de Queiroz, R.: Processing JPEG-compressed images and documents. IEEE Trans. Image Process. **8**(12), 1661–1672 (1998)

12. de Franca Pereira e Silva, G., Lins, R.D.: Assessing the OCR degradation in the generation of JPEG, PNG, and TIFF files from Adobe PDF. In: ITS 2010 IEEE-SBrT International Telecommunications Symposium (2010)
13. Elad, M., Aharon, M.: Image denoising via sparse and redundant representations over learned dictionaries. IEEE Trans. Image Process. **15**(12), 3736–3745 (2006)
14. Jung, C., Jiao, L., Qi, H., Sun, T.: Image deblocking via sparse representation. Signal Process. Image Commun. **27**(6), 663–677 (2012)
15. Kartalov, T., Ivanovski, Z., Panovski, L., Karam, L.: An adaptive POCS algorithm for compression artifacts removal. In: 9th International Symposium on Signal Processing and Its Applications, 2007. ISSPA 2007, pp. 1–4 (2007)
16. Lam, E.Y.: Compound document compression with model-based biased reconstruction. J. Electron. Imaging **13**(1), 191–197 (2004)
17. Otsu, N.: A threshold selection method from gray-level histograms. IEEE Trans. Syst. Man Cybernet. **9**(1), 62–66 (1979)
18. Oztan, B., Malik, A., Fan, Z., Eschbach, R.: Removal of artifacts from JPEG compressed document images. In: Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series, vol. 6493, pp. 1–9 (2007)
19. Pati, Y.C., Rezaiifar, R., Krishnaprasad, P.S.: Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition. In: Conference Record of the 27th Asilomar Conference on Signals, Systems and Computers, vol. 1, pp. 40–44 (1993)
20. Pham, T.A., Delalandre, M.: Effective decompression of JPEG document images. IEEE Trans. Image Process. **25**(6), 3655–3670 (2016)
21. Prost, R., Ding, Y., Baskurt, A.: JPEG dequantization array for regularized decompression. IEEE Trans. Image Process. **6**(6), 883–888 (1997)
22. Samadani, R.: Characterizing and estimating block DCT image compression quantization parameters, pp. 1230–1234 (2005)
23. Samadani, R., Sundararajan, A., Said, A.: Deringing and deblocking DCT compression artifacts with efficient shifted transforms, pp. 1799–1802 (2004)
24. Saraswat, N., Ghosh, H.: A study on size optimization of scanned textual documents. Lect. Notes Comput. Sci. **9431**, 75–86 (2016)
25. Savakis, A.E.: Evaluation of lossless compression methods for gray scale document images. In: Proceedings 2000 International Conference on Image Processing (Cat. No. 00CH37101), vol. 1, pp. 136–139 (2000)
26. Wallace, G.K.: The JPEG still picture compression standard. Commun. ACM **34**(4), 30–44 (1991)
27. Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P.: Image quality assessment: from error visibility to structural similarity. IEEE Trans. Image Process. **13**(4), 600–612 (2004)
28. Wong, T., Bouman, C., Pollak, I., Fan, Z.: A document image model and estimation algorithm for optimized JPEG decompression. IEEE Trans. Image Process. **18**(11), 2518–2535 (2009)
29. Yang, S., Hu, Y.H., Tull, D.: Blocking effect removal using robust statistics and line process. In: 1999 IEEE 3rd Workshop on Multimedia Signal Processing, pp. 315–320 (1999)
30. Yang, Y., Galatsanos, N., Katsaggelos, A.: Projection-based spatially adaptive reconstruction of block-transform compressed images. IEEE Trans. Image Process. **4**(7), 896–908 (1995)
31. Zhang, P., Wang, S., Wang, R.: Reducing frequency-domain artifacts of binary image due to coarse sampling by repeated interpolation and smoothing of radon projections. J. Visual Commun. Image Represent. **23**, 697–704 (2012)
32. Zhang, X., Xiong, R., Fan, X., Ma, S., Gao, W.: Compression artifact reduction by overlapped-block transform coefficient estimation with block similarity. IEEE Trans. Image Process. **22**(12), 4613–4626 (2013)
33. Zou, J.J., Yan, H.: A deblocking method for BDCT compressed images based on adaptive projections. IEEE Trans. Circuits Syst. Video Technol. **15**(3), 430–435 (2005)