Effective decompression of JPEG document images

The-Anh Pham, Mathieu Delalandre

***Abstract*—This work concentrates on developing an effective approach for decompressing JPEG document images. Our main goal is targeted to time-critical applications, especially to those situated on mobile network infrastructures. To this aim, the proposed approach is designed to work either in the transform domain or image spatial plane. Specifically, the image blocks are first classified into smooth blocks (e.g., background, uniform regions) and non-smooth blocks (e.g., text, graphics, line-drawings). Next, the smooth blocks are fully decoded in the transform domain by minimizing the total block boundary variation, which is very efficient to compute. For decoding non-smooth blocks, a novel text model is presented that accounts for the specifics of document content. Additionally, an efficient optimization algorithm is introduced to reconstruct the non-smooth blocks. The proposed approach has been validated by extensive experiments, demonstrating a significant improvement of visual quality, assuming that document images have been encoded at very low bit-rates and thus are subject to severe distortion.**

***Index Terms*—Document decompression, JPEG decoding, total variation, soft classification.**

I. Introduction

Currently, due to the quickly expanding variety of portable digital imaging devices (e.g., cameras, smartphones, electronic camera-pen systems), the acquisition of document images has become much more convenient, thus leading to the huge expansion of document data. In this expeditious evolution, the real challenges in document image analysis (DIA) have shifted toward effective pervasive computing, storage, sharing and browsing of mass digitized documents. A promising and efficient approach is to exploit the benefits of very low bit rate compression technologies. Lossless compression algorithms allow the encoded images to be correctly reconstructed, but the gain of the compression ratio is not sufficiently high. In contrast, lossy compression algorithms provide very low bit rates at the cost of losing a certain degree of image quality. In addition, the level of image quality reduction can be easily controlled by pre-determined parameters. For these reasons, the multimedia data, in their current form, are mostly compressed using a lossy compression scheme.

With the rapid increase of 3G-/4G-based markets, handheld devices and infrastructures, both the pervasive computation of document images and exploitation of related applications are becoming crucial needs for mobile users. Customers want to access and retrieve good quality images while expecting a low bandwidth consumption. Additional needs include fast response time and memory-efficient usage. These constraints imply that document images must be encoded and decoded in a very efficient manner. Several efforts have been carried out for lossless and near lossless compression methods that are devoted to document images such as MRC [1], DjVu [2], DigiPaper [3], and TSMAP/RDOS [4]. Although these attempts were shown to outperform the state-of-the-art compression techniques on a particular class of document images, they require new standards for image representation. This constraint is not always applicable for many applications, especially for those on mobile markets. Mobile users, in practice, prefer to keep the existing standards, such as JPEG [5], for the images being accessed.

This work concentrates on improving the visual quality of document images compressed by the JPEG standard. At low bit-rate coding, JPEG encoded images are subject to heavy distortion of both blocking and ringing artifacts. These artifacts can make the visual perception of document images impaired or even invisible. Although a large number of methods have been proposed to address coding artifacts, they suffer from the major issue of expensive computational cost. This aspect prevents them from being applicable for time-critical applications, especially for those developed on low bandwidth and restricted resource platforms. In addition, most of the existing work is devoted to natural images [6]–[14]. There has been little effort to address the same problem for document content [15]–[17]. Inspired by all of these facts, we attempt to bring an effective approach for decoding the JPEG document images. The proposed approach has been developed to produce a substantial improvement of visual quality while incurring a low computational cost. In doing so, we have restricted our approach to the scenarios that document images have been compressed in very high compression rates and hence they are exposed to heavy disturbance of coding artifacts.

The rest of this paper has been organized in the following structure. Section II reviews the most recent research for post-processing JPEG artifacts. Section III provides a global description of the proposed approach, including three main components: block classification, smooth block decompression, and text block decompression. Next, the block classification process is detailed in Section IV, while the decompression of smooth blocks and text blocks are described in Section V and VI, respectively. Experimental results are presented in Section VII. Finally, we conclude the paper and give several lines of future extensions in Section VIII.

II. Related work

Figure 1 provides a classification of different approaches for JPEG artifact post-processing. Generally speaking, there are two main strategies for dealing with JPEG artifacts, namely, iterative and non-iterative approaches. The former approach is featured by an iterative optimization process in which an objective function is incorporated based on some prior model

The authors are with the Computer Science Lab of Polytech Tours school, 64 avenue Jean Portalis, 37200 Tours, France. Emails: the-anh.pham@hdu.edu.vn, mathieu.delalandre@univ-tours.fr

of the signal or original image. The latter approach is typically composed of two steps: artifact detection and post-processing. Both of the approaches can be processed in the spatial image domain and/or Discrete Cosine Transform (DCT) domain. In what follows, we shall discuss the most representative methods for each approach. We shall also provide deeper analysis, throughout this paper, for the methods that are closely related to our approach.

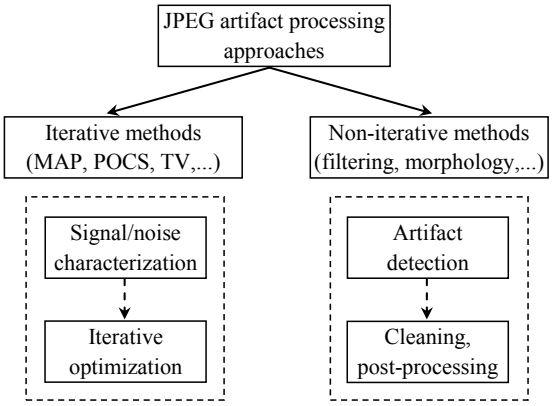


Fig. 1. Classification of different approaches for JPEG artifact processing.

Iterative methods: Typical iterative methods for artifact post-processing include maximum a posteriori estimation (MAP) [7], [17]–[19], projection onto convex sets (POCS) [6], [8], [9], total variation (TV) regularization [10], [20], [21], and sparse representation based on a learned dictionary (SRLD) [11], [13], [14]. Table I shows the main characterizations of each method, and the details are presented hereafter.

TABLE I
CHARACTERIZATION OF ITERATIVE METHODS

Approach	Original signal model	Domain
MAP	Gibbs [7], [17], [18], BS-PM [19]	DCT & Spatial
POCS	Neighboring constraint sets [6], [8], [9]	DCT & Spatial
TV	Gradient magnitude sum [10], [20], [21]	DCT & Spatial
SRLD	Learned dictionaries [11], [13], [14]	DCT & Spatial

The MAP-based approach has been well-developed for image denoising and JPEG artifact treatment. It solves the inverse problem of finding the image X that corresponds to the maximum *a posteriori* probability $P(X|Y)$ given an observed image Y. For transform image coding, building the prior distribution $P(X)$ is a critical task to effectively denoise the corrupted images. To this aim, Gaussian Markov random field (MRF), non-Gaussian MRF, and Gibbs models have been extensively exploited in the literature [7], [17], [18]. A recent model based on block similarity prior model (BS-PM) [19] has been introduced to characterize more efficiently the local structure of image content. Once the prior models have been established, image reconstruction is performed through an iterative algorithm that consists of two steps: updating the latent variables and sanity checking based on the quantization coding constraint. The former is typically done in the image spatial plane, whereas the latter must be processed in the transform domain. In this way, computational complexity becomes a major problem.

Traditionally, POCS-based theory [6], [8], [9] has been employed for post-processing image coding artifacts. In its essence, a POCS-based method defines a set of constraints, each of which is described by a closed convex set. The artifact-free image is then estimated as the intersection of these convex sets. The very first constraint is formed from the image transform coding (i.e., quantization constraint). Depending on applications, other constraints can be established to describe the smoothness of the original image. Such constraints, as defined in [8], [9], for instance, account for the close correlation between two adjacent pixels in either the horizontal or vertical direction. Because a sufficiently large number of constraints must be created to characterize the original image, one of the defects of the POCS-based method involves a variety of parameters used to define these constraints. In addition, these methods are evidently subject to being computationally intensive.

Total variation (TV) regularization methods have also been widely used to address compression artifacts [10], [20], [21]. The rationale behind the TV approach is realized based on the fact that the total variation of a noisy signal is generally higher than that of the original signal. Consequently, image denoising is handled by minimizing a proper TV function. The work in [20], for instance, suggested that a weighted TV function should be computed by using the L_1-norm. The authors in [10] proposed using the L_2-norm to build up a continuous TV model. The main deficiency, however, with such TV functions is termed as the staircasing effect [22]. Total generalized variation (TGV) has been introduced in [21] to alleviate this defect to some extent. After the TV functions have been determined, a variety of well-established methods in the convex programming field can be applied to solve the minimization problem.

Recent development of JPEG artifact processing has shifted towards dictionary-based sparse representation [11], [13], [14]. The authors in [11] first introduced a denoising model based on sparse and redundant representation. The underlying spirit is to build up a dictionary consisting of the atoms that are used to sparsely represent the images. This dictionary can be constructed in an offline process using a set of noise-free training image patches [11], [13] or in an online phase using the input image itself [11], [14]. In the latter case, image restoration is combined with dictionary learning in one unified process. In doing so, K-singular value decomposition (K-SVD) algorithm [23] and orthogonal matching pursuit (OMP) algorithm [24] are often applied for both dictionary learning and image denoising. While these methods have been shown to produce a substantial improvement of visual quality, the main problem involves the expensive computational cost, making them inapplicable for time-critical applications.

TABLE II
CHARACTERIZATION OF NON-ITERATIVE METHODS

Reference	Artifact detection/localization	Domain
[25]	Edge location and edge's proximity	DCT
[26]	2-D step function	DCT
[27]	2-D step function, edge filtering	DCT
[16]	Background detection	DCT & Spatial

Non-iterative methods: Generally, a non-iterative method treats the JPEG artifacts in two steps: detection of possible artifact locations and artifact cleaning/post-processing. Because ringing artifacts commonly occur around sharp transitions such as edges and contours, a number of methods have been presented to extract edge information. The obtained edge information (i.e., location and orientation) is then used to guide the cleaning process. Table II characterizes the key properties of each method.

In [25], edge map is first extracted in the DCT domain, provided the assumption that each block contains a simple straight edge line. Then, the ringing cleaning process is performed in proximity to the edges using several heuristic criteria (i.e., sharp edge block, edge height, edge fitting quality). The authors in [26] constructed a model of blocking artifact in the DCT domain as a 2-D step function. The rationale of using such a step function is reliant on the fact that the blocking artifact causes the abrupt discontinuities at the boundaries of the blocks. Hence, an intermediate block, formed from the common boundary of two adjacent blocks, is derived in the DCT domain and is then used to estimate the parameters of the step function. The estimated parameters served as an indication of blocking distortion. If the blocking measure is sufficiently high, blocking treatment is performed by replacing the corresponding step function with a linear one. A similar work was also presented in [27] while incorporating an additional process to filter out the real edges from blocking edges.

Document-dedicated methods: All the work discussed so far is specifically designed to address natural images; there has been little effort devoted to post-processing artifacts for JPEG compressed document images [15]–[17]. In [17], the image blocks are first classified into three types: background, text/graphics and picture blocks. Next, a specific model is constructed for each type of block while taking into account the characteristics of the considered blocks. Specifically, the Gaussian Markov random field (GMRF) model was employed to characterize the background blocks, and a document image model was introduced for representing the text/graphics blocks. Block decoding is then performed accordingly to each type of block. Excellent results were reported in the experiments; however, no discussion concerning computational complexity was provided. In [15], a biased reconstruction of JPEG documents is driven by computing the centroid of each code block provided a prior distribution model of the transform coefficients. To this aim, two models (i.e., Laplace and Gaussian distribution) are exploited to estimate the centroids of the code blocks. Experimental results showed slightly better results when compared with the conventional JPEG decoder. A report in [17] further shows that this approach is even worse than the JPEG algorithm for all of the studied datasets.

In [16], a non-iterative and simple computation method was proposed to specifically address the ringing artifact. It is based on the observation that the ringing artifact is more dominated in background regions than in text regions. First, foreground/background segmentation is performed by using an automatic thresholding technique [28] in conjunction with a simple morphological operator. Next, all of the noisy pixels in the background regions are adjusted by the same value, which is estimated as the most frequent gray level of the background. However, ringing reduction is processed only for background regions, not for the text pixels or in proximity to the text's edges. Hence, visual quality improvement is not satisfactory, although the proposed method is very time-efficient.

Concluding remarks: To conclude this section, we wish to highlight that, despite increasing attempts devoted to dealing with compression artifacts, questions have been raised about the computational complexity of the existing work. Additionally, most research efforts for artifact reduction have been targeted to natural images. To the best knowledge of the authors, little attention has been paid to reduce these artifact for document content [15]–[17]. The most noticeable work [17] provides a significant improvement of visual quality, but it is too costly. On the other hand, simple computation methods, such as [15], [16], do not give sufficiently decent results. All of these facts have convinced us to seek for a competitive approach to produce substantially better decoding image quality while also requiring a very low computational cost. In the following section, we describe such an approach.

III. Overview of the Proposed Approach

The proposed approach is briefly described in Figure 2 and is composed of three main components: (C1) block classification, (C2) smooth block reconstruction in the transform domain and (C3) non-smooth block decoding in the spatial image plane. First, the 8×8 DCT blocks are classified into either smooth blocks (e.g., background, uniform areas) or non-smooth blocks (e.g., text, graphics, line-drawings, pictures). Next, reconstruction of smooth blocks is performed solely in the DCT domain based on two sub-processes: fast extracting total block boundary variation (TBBV) and minimizing the TBBV-based objective function. For non-smooth blocks, the decoding process is carried out only in the spatial domain. It consists of the construction of a text document model that accounts for the specific characteristics of document content, followed by an optimization process for decoding the text blocks.

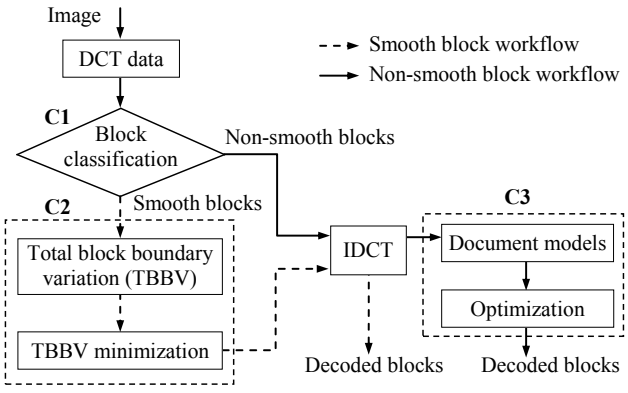


Fig. 2. Overview of the proposed approach.

Before going into the details of all of these steps, we shall review some basic manipulations used in the JPEG codec. Given an image f that has a size of $M \times N$, let B_x and B_y be the number of non-overlapping 8×8 blocks in the vertical

and horizontal directions, respectively (i.e., $B_x = \lceil \frac{M}{8} \rceil$ and $B_y = \lceil \frac{N}{8} \rceil$). For the sake of presentation, we denote a block located at the k^{th} row and l^{th} column by (k, l) with $k = 0, 1, \ldots, B_x - 1$ and $l = 0, 1, \ldots, B_y - 1$. We also denote $f^{k,l}(x, y)$ and $F^{k,l}(m, n)$ as the intensity values and DCT coefficients of the block (k, l), respectively.

The JPEG algorithm divides an input image into non-overlapping 8×8 blocks, each of which is then individually compressed under a pipeline of the following steps: DCT transform, quantization and entropy coding. The first step computes the DCT coefficients of each image block as follows:

$$F^{k,l}(m,n) = \frac{e(m)e(n)}{4} \sum_{x=0}^{7} \sum_{y=0}^{7} f^{k,l}(x,y) C_{16}^{(2x+1)m} C_{16}^{(2y+1)n} \tag{1}$$

where $m, n \in \{0, 1, \ldots, 7\}$, and the two functions C_b^a and $e(m)$ are represented in the following expressions:

$$C_b^a = \cos(\frac{a\pi}{b}) \tag{2}$$

$$e(m) = \begin{cases} \frac{1}{\sqrt{2}} & \text{if } m = 0 \\ 1 & \text{otherwise} \end{cases} \tag{3}$$

The DCT is a linear and invertible transform so that the inverse DCT (IDCT) is given as follows:

$$f^{k,l}(x,y) = \frac{1}{4} \sum_{m=0}^{7} \sum_{n=0}^{7} e(m)e(n) F^{k,l}(m,n) C_{16}^{(2x+1)m} C_{16}^{(2y+1)n} \tag{4}$$

The quantization step divides the DCT coefficients by using a quantization matrix $Q(m, n)$ and then rounds the results to the nearest integers. Formally, the quantized coefficients, $F_q^{k,l}(m, n)$, of the block (k, l) are derived by (5):

$$F_q^{k,l}(m,n) = \text{round}\left(\frac{F^{k,l}(m,n)}{Q(m,n)}\right) \tag{5}$$

To decompress the image, the IDCT (4) is applied to the dequantized DCT coefficients, $F_d^{k,l}(m, n)$, of each image block. Here, the dequantized coefficients are reconstructed in the following form:

$$F_d^{k,l}(m,n) = F_q^{k,l}(m,n) Q(m,n) \tag{6}$$

At low bit-rate compression, because most of the quantized coefficients are close to zero, the dequantized coefficients cannot be fully reconstructed. As a result, undesired artifacts are added to the decoded images. In the following sections, we present an effective approach for reducing these artifacts. The presentation of the proposed approach is organized in the following logical order: block classification, smooth block reconstruction and non-smooth block decompression.

IV. Block classification

Block classification is processed in the DCT domain based on AC energy due to its simple computation and effective performance. In fact, AC energy is often used in the literature to differentiate background blocks from text and picture blocks [15], [17], [29]. In the current work, the DCT blocks are classified into two types: smooth blocks (i.e., areas of high correlated information) and non-smooth blocks (i.e., text blocks, graphics, pictures). Artifact post-processing is carried out separately for each type of block. We employ a simple and efficient criterion based on AC energy to perform block classification. In its essence, the AC energy of a DCT block is simply computed as the sum of the squares of AC coefficients of that block. For a smooth block, most of the AC coefficients are zero or close to zero. Therefore, the corresponding AC energy should be pretty low. In contrast, a non-smooth block often has high AC energy. Consequently, block classification is done by simply thresholding the AC energy by using a pre-determined threshold parameter T_{seg} (see the experiment section). Because the ringing artifact causes severe disturbance in document readability, it is advised to not miss the true text blocks. Therefore, the parameter T_{seg} is preferably set to a relative low value as to cover the true text blocks.

V. Smooth block decoding in DCT domain

We propose to exploit the total variation (TV) regularization model to reconstruct the smooth blocks. Traditional TV-based methods need to process in both spatial and compression domains [10], [11], [20], [21], [30]. In other words, DCT and inverse DCT are iteratively applied many times during the regularization process and thus computational load becomes a major issue. Differentiating from these methods, we propose to use the total block boundary variation (TBBV) model to reconstruct the smooth blocks. Specifically, the TBBV model is directly processed in the DCT domain and built to account for distortion at the boundaries of the blocks. In addition, the iterative regularization process is fully performed in the DCT domain, avoiding the need of switching back to the spatial image plane. As a result, it is unnecessary to apply the costly inverse DCT, making the whole reconstruction process very efficient.

In what follows, we first provide a formal definition of total block boundary variation. Next, we present a means to efficiently compute TBBV in the DCT domain for the full image level. Computation of TBBV for the smooth blocks is then simply derived. Finally, we employ Newton's method to perform TBBV regularization in the DCT domain.

A. Definition of total block boundary variation

To enjoy the benefits of efficient computation, we propose to analyze the block variation directly in the DCT domain at the super-pixel level. Specifically, each block (k, l) is partitioned into 16 subregions, each of which is regarded as a super-pixel corresponding to a local window that has a size of 2×2. Each super-pixel (u, v) is assigned with an average intensity value $S_{uv}^{k,l}$ $(u, v \in \{0, 1, 2, 3\})$ that is computed as follows:

$$S_{uv}^{k,l} = \frac{1}{4} \sum_{i=0}^{1} \sum_{j=0}^{1} f^{k,l}(2u + i, 2v + j) \tag{7}$$

Only the super-pixels located at the common boundary of two adjacent blocks are considered to compute the block variation as illustrated in Figure 3.

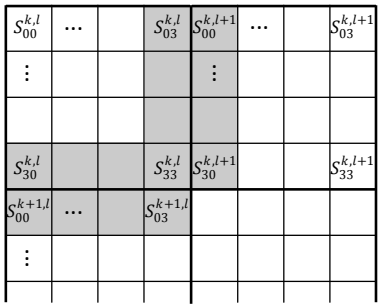


Fig. 3. Computing block boundary variation at 2×2 super-pixel level.

Formally, the total block boundary variation, $TBBV(f)$, of the image f is defined as follows:

$$TBBV(f) = \sum_{k=0}^{B_x-2} \sum_{l=0}^{B_y-2} (G_H(f^{k,l})^2 + G_V(f^{k,l})^2) \tag{8}$$

where

$$G_H(f^{k,l}) = \sum_{i=0}^{3} (S_{i0}^{k,l+1} - S_{i3}^{k,l})$$

$$G_V(f^{k,l}) = \sum_{i=0}^{3} (S_{0i}^{k+1,l} - S_{3i}^{k,l})$$

The two components, $G_H(f^{k,l})$ and $G_V(f^{k,l})$, are called the horizontal and vertical block boundary variation of the block (k,l), respectively.

B. Fast computation of total block boundary variation

In this subsection, we investigate a means for the fast computation of $TBBV(f)$ in the DCT domain. The following materials are targeted to computing $G_H(f^{k,l})$, although the same process can be applied to compute $G_V(f^{k,l})$.

By substituting (4) into (7) and rearranging the terms in a similar manner to that given in [31], we obtain the following expression:

$$S_{uv}^{k,l} = \sum_{m=0}^{7} \sum_{n=0}^{7} F^{k,l}(m,n) w_{uv}(m,n) \tag{9}$$

where $w_{uv}(m,n)$ is derived as follows:

$$w_{uv}(m,n) = \frac{1}{4} e(m)e(n) C_{16}^{m} C_{8}^{(2u+1)m} C_{16}^{n} C_{8}^{(2v+1)n}$$

To compute $G_H(f^{k,l})$, we define the sub-terms D_i with $i \in \{0,1,2,3\}$ given in (10):

$$D_i = S_{i0}^{k,l+1} - S_{i3}^{k,l} \tag{10}$$

Specifically, D_0 is computed hereafter and D_i can be derived accordingly:

$$\begin{aligned} D_0 &= S_{00}^{k,l+1} - S_{03}^{k,l} \\ &= \sum_{m=0}^{7} \sum_{n=0}^{7} (F^{k,l+1}(m,n) w_{00}(m,n) - F^{k,l}(m,n) w_{03}(m,n)) \\ &= \sum_{m=0}^{7} \sum_{n=0}^{7} T(m,n) (F^{k,l+1}(m,n) C_8^n - F^{k,l}(m,n) C_8^{7n}) \end{aligned}$$

where

$$T(m,n) = \frac{e(m)e(n) C_{16}^{n} C_{16}^{m} C_{8}^{m}}{4}$$

Note that $C_8^{7n} = (-1)^n C_8^n$; hence, we obtain:

$$D_0 = \sum_{m=0}^{7} \sum_{n=0}^{7} \frac{e(m)e(n) C_{16}^{n} C_{16}^{m}}{4} C_8^m C_8^n R(m,n) \tag{11}$$

with $R(m,n) = F^{k,l+1}(m,n) - (-1)^n F^{k,l}(m,n)$. In the same manner, the D_i terms ($1 \le i \le 3$) are derived by (12):

$$D_i = \sum_{m=0}^{7} \sum_{n=0}^{7} \frac{e(m)e(n) C_{16}^{n} C_{16}^{m}}{4} C_8^{(2i+1)m} C_8^n R(m,n) \tag{12}$$

Denote $z_k(m,n) = \frac{1}{4} e(m)e(n) C_{16}^{n} C_{16}^{m} C_8^{km} C_8^n$ with $k \in \{1,3,5,7\}$, it is straightforward to derive the following properties from $z_k(m,n)$:

- $\frac{z_7(m,n)}{z_1(m,n)} = \frac{z_5(m,n)}{z_3(m,n)} = (-1)^m$
- $k_m = \frac{z_3(m,n)}{z_1(m,n)} = \frac{C_8^{3m}}{C_8^m}$ (see Table III)
- $z_k(m,n) = 0$ for either $m = 4$ or $n = 4$

TABLE III
PRECOMPUTATION OF k_m

m	0	1	2	3	5	6	7
k_m	1	$\frac{C_8^3}{C_8^1}$	-1	$-\frac{C_8^1}{C_8^3}$	$\frac{C_8^1}{C_8^3}$	-1	$-\frac{C_8^3}{C_8^1}$

Finally, the horizontal block boundary variation, $G_H(f^{k,l})$, is computed by summing up D_i in (12) as follows:

$$\begin{aligned} G_H(f^{k,l}) &= \sum_{i=0}^{3} \sum_{m=0}^{7} \sum_{n=0}^{7} z_{2i+1}(m,n) R(m,n) \\ &= \sum_{m=0}^{7} \sum_{n=0}^{7} R(m,n) \sum_{i=0}^{3} z_{2i+1}(m,n) \\ &= \sum_{m=0}^{7} \sum_{n=0}^{7} R(m,n) Z(m,n) \end{aligned} \tag{13}$$

where

$$\begin{aligned} Z(m,n) &= \sum_{i=0}^{3} z_{2i+1}(m,n) \\ &= z_1(m,n)(1 + k_m + k_m(-1)^m + (-1)^m) \\ &= z_1(m,n)(1 + (-1)^m)(1 + k_m) \end{aligned}$$

It is worthwhile showing that $Z(m,n) = 0$ for either $m \in \{1,3,5,7,2,4,6\}$ (see k_m in Table III) or $n = 4$. As a result, $G_H(f^{k,l})$ is simplified to (14):

$$\begin{aligned} G_H(f^{k,l}) &= 4 \sum_{\substack{n=0 \\ n \neq 4}}^{7} R(0,n) z_1(0,n) \\ &= 4 \sum_{\substack{n=0 \\ n \neq 4}}^{7} z_1(0,n) (F^{k,l+1}(0,n) - (-1)^n F^{k,l}(0,n)) \end{aligned} \tag{14}$$

Similarly, the vertical block boundary variation, $G_V(f^{k,l})$, of the block (k,l) can be derived by (15):

$$G_V(f^{k,l}) = 4\sum_{\substack{m=0\\m\neq 4}}^{7} z_1(m,0)(F^{k+1,l}(m,0) - (-1)^m F^{k,l}(m,0)) \tag{15}$$

The computational complexity of (14) in terms of the number of multiplication (M) and addition (A) is simply 7M + 13A. Consequently, computation of both $G_H(f^{k,l})$ and $G_V(f^{k,l})$ can be done using 14M + 26A, which is extremely efficient compared with applying full IDCT (i.e., 4096M + 4096A) or even fast IDCT (i.e., 94M + 454A), as reported in [32]. As we have already mentioned earlier, the process of TBBV regularization is applied to the smooth blocks only. Therefore, we derive an appropriate form, $TBBV_S(f)$, to compute the total block boundary variation for the smooth blocks as follows:

$$TBBV_S(f) = \sum_{(k,l)\in B_S} (G_H(f^{k,l})^2 + G_V(f^{k,l})^2) \tag{16}$$

where B_S denotes the set of all smooth blocks that are obtained from the block classification stage (Section IV).

C. Efficient reconstruction of smooth blocks in DCT domain

The decompression of smooth blocks is driven by minimizing a proper objective function that is developed based on $TBBV_S(f)$ while being subject to the quantization constraint of transform-based coding. Specifically, the quantization constraint is established as follows:

$$F_q^{k,l}(m,n) - \frac{1}{2} \leq \frac{F^{k,l}(m,n)}{Q(m,n)} \leq F_q^{k,l}(m,n) + \frac{1}{2} \tag{17}$$

Reconstruction of the smooth blocks from the quantized coefficients is shifted to the problem of finding the optimal solution $\hat{F}$, which minimizes the following objective function:

$$\hat{F} = \arg\min_{F\in U} f_{obj}(F) \tag{18}$$

where

$$f_{obj}(F) = \sum_{(k,l)\in B_S} (G_H(f^{k,l})^2 + G_V(f^{k,l})^2) + \lambda \sum_{(k,l)\in B_S} \sum_{(m,n)\in E} (F^{k,l}(m,n) - F_d^{k,l}(m,n))^2$$

Here, the set U contains all possible F satisfying the constraint in (17) and λ is the Lagrange constant driving the restoration process. The set E contains 13 DCT coefficients that are involved in computing $G_H(f^{k,l})$ and $G_V(f^{k,l})$ for each block (k,l). Formally, $E = \{(u,0)\} \cup \{(0,v)\}$ where $u \in \{0,1,2,3,5,6,7\}$ and $v \in \{1,2,3,5,6,7\}$. This means that it is unnecessary to post-process all of the DCT coefficients. Instead, only the DCT coefficients of the topmost row and leftmost column of each smooth block are considered.

It is worth pointing out that the objective function (18) and the constraint set U are both convex. Furthermore, the objective function is twice-differentiable. This fact drives the selection of Newton's method to solve the problem in (18). In its essence, Newton's method is an iterative process of finding the stationary point of the objective function $f_{obj}(F)$ while taking into consideration the quantization constraint. Specifically, the solution is updated at each iteration as follows:

$$F^{(t+1)} = P_U\left[F^{(t)} - \frac{f'_{obj}(F^{(t)})}{f''_{obj}(F^{(t)})}\right] \tag{19}$$

where:

- $F^{(0)}$ is initialized by concatenating the dequantized DCT coefficients from all the smooth blocks.
- $F^{(t)}$ is the solution at the iteration t.
- $f'_{obj}(F^{(t)})$ and $f''_{obj}(F^{(t)})$ are the first order and second order derivatives of f_{obj} at $F^{(t)}$, respectively.
- $P_U[X]$ is a clipping operator to project X into the set U. Basically, the clipping operator of a scalar real-valued x projected into the interval $U = [u_1, u_2]$ is defined as follows:

$$P_U[x] = \begin{cases} x & \text{if } u_1 \leq x \leq u_2 \\ u_2 & \text{if } x > u_2 \\ u_1 & \text{if } x < u_1 \end{cases} \tag{20}$$

Because of the very fast convergence of Newton's method, decent performance can be obtained after a few iterations. In our experiments, the number of iterations is set to two, unless otherwise stated.

To this time, we obtain the reconstructed DCT coefficients that are taking part in the computation of $TBBV_S(f)$, while the remaining DCT data is left unchanged. Next, inverse DCT transform is applied to represent the image in the spatial domain. For non-smooth blocks, a dedicated post-processing algorithm is introduced which takes into consideration the specificities of document content. This algorithm is presented in the next section.

VI. Non-smooth block decoding in spatial domain

One of the key properties of document content is realized by the unbalanced distribution of non-smooth blocks and smooth blocks. Generally, the non-smooth blocks are much less dominated than the smooth blocks. Therefore, the non-smooth blocks can be efficiently handled in the spatial domain. In this work, we consider the non-smooth blocks as those comprising texts and graphics/line-drawing objects. For simplification, we refer to them as text blocks. In what follows, we first introduce a dedicated model for representing the text blocks. Then, an effective optimization algorithm is presented to reconstruct the text blocks by combining Bayes' framework and log-likelihood analysis. All of these process are detailed in the following subsections.

A. Constructing text document model

Let X_b be a text block and $\mu_{i,j} \in \mathbb{R}$ be the intensity value observed at the point (i,j) in the block X_b. For the sake of presentation, it is assumed that $\mu_{i,j} \in [0,1]$. Furthermore, let μ_F and μ_B be the foreground and background intensities of X_b, respectively, where $\mu_F < \mu_B$ and $\mu_F, \mu_B \in [0,1]$. In general, the values μ_F and μ_B can be determined in advance

by employing a foreground/background separation process of the image decoded by the conventional JPEG algorithm. For instance, Otsu's method and the k-means algorithm were employed in [16], [17] for such a purpose.

To optimize the decoding process of text blocks, we formulate the reconstruction problem as a *soft* decision making problem. Specifically, we define two hypotheses H_0 and H_1 for each text block as follows:

- H_0: The pixel (i,j) is a foreground point.
- H_1: The pixel (i,j) is a background point.

From the classification point of view, $\mu_{i,j}$ can be treated as the *evidence* score observed at the pixel (i,j). Generally, the closer to μ_F (*resp.* μ_B), the higher the likelihood that the pixel (i,j) is a foreground pixel (*resp.* background pixel). However, the final decision must be made while taking into account other context information extracted from neighboring areas. A simple context rule, for instance, can be extracted as follows: if two pixels $(i,j-1)$ and $(i,j+1)$ are background pixels, then it is likely that the pixel (i,j) is also a background pixel. To exploit such context information in a systematic fashion, it is needed to define an appropriate cost function in order to associate specific punishment with each decision being made. Specifically, each decision (i.e., H_0 or H_1) made for each pixel (i,j) is associated with a specific error cost that is determined based on the certainty or confidence of the system about that decision. The higher is the confidence, the lower is the error cost and vice versa. The problem now can be stated as follows: finding the decisions for all pixels of input image so as to minimize the total error cost. To solve this problem, we shall first employ a cost function, $C_{log}(\mu)$, introduced in [33] and then propose a novel algorithm to minimize the cost function.

Cost function for soft decision making: Given a text block X_b whose intensities are denoted by $\mu_{i,j}$, the error cost function, $C_{log}(\mu)$, of making decisions for all pixels of X_b is defined as follows:

$$C_{log}(\mu) = \sum_{(i,j)} \sum_{k=0}^{1} p(H_k|\mu_{i,j}) C'_{log}(H_k, y_{i,j}) \tag{21}$$

where $p(H_k|\mu_{i,j})$ is the posterior probability at the pixel (i,j), $y_{i,j}$ is the confidence score estimated at (i,j), and $C'_{log}(h,y)$ is the cost of making a decision $h \in \{H_0, H_1\}$ with a confidence score $y \in \mathbb{R}$, which is defined by (22):

$$C'_{log}(h,y) = \begin{cases} \log(1+\exp\{-y - \text{logit}(p(H_0)\}) & \text{if } h = H_0 \\ \log(1+\exp\{y + \text{logit}(p(H_0)\}) & \text{if } h = H_1 \end{cases} \tag{22}$$

where the logit function is the inverse of the logistic function, which maps a probabilistic value $p \in (0,1)$ to a real value $y \in (-\infty, +\infty)$:

$$y = \text{logit}(p) = \log\left(\frac{p}{1-p}\right)$$

The confidence score $y_{i,j}$ at the point (i,j) is estimated in terms of the log-likelihood ratio (LLR):

$$y_{i,j} = LLR(\mu_{i,j}) = \log\left(\frac{p(\mu_{i,j}|H_0)}{p(\mu_{i,j}|H_1)}\right) \tag{23}$$

Equivalently, $y_{i,j}$ can be represented in the following form by using Bayes' rule:

$$\begin{aligned} y_{ij} + \text{logit}(p(H_0)) &= \text{logit}(p(H_0|\mu_{i,j})) \\ &= \log\left(\frac{p(H_0|\mu_{i,j})}{1-p(H_0|\mu_{i,j})}\right) \end{aligned} \tag{24}$$

Combining (21) and (24), we can define an individual cost, $E_{prob}(i,j)$, for every point (i,j):

$$\begin{aligned} E_{prob}(i,j) &= \sum_{k=0}^{1} p(H_k|\mu_{i,j}) \log(1 + \frac{p(H_{1-k}|\mu_{i,j})}{p(H_k|\mu_{i,j})}) \\ &= -\sum_{k=0}^{1} p(H_k|\mu_{i,j}) \log(p(H_k|\mu_{i,j})) \end{aligned} \tag{25}$$

Denoting the two-element set $\{p(H_0|\mu_{i,j}), p(H_1|\mu_{i,j})\}$ as the probability distribution of the point (i,j), the individual cost $E_{prob}(i,j)$ can be interpreted as the Shannon entropy of the point (i,j) in terms of posterior probability [33]. Consequently, the total error cost $C_{log}(\mu)$ is the sum of probability entropy computed for every pixel (i,j) and can be represented in the following form:

$$C_{log}(\mu) = \sum_{i,j} E_{prob}(i,j) \tag{26}$$

The error cost expressed in (26) can also be interpreted as the entropy of the entire image f in terms of posterior probability.

The rationale of constructing the cost $C'_{log}(h,y)$ is interpreted as follows. From (23), it can be seen that y goes to positive infinity as long as the hypothesis H_0 is true. Similarly, y goes to negative infinity if the hypothesis H_1 is true. Ideally, it is expected that $h = H_0$ corresponds to $y \in [0, +\infty)$ and $h = H_1$ corresponds to $y \in (-\infty, 0)$. Therefore, if a decision is made with a strong confidence (i.e., $|y| = \infty$), the cost $C'_{log}(h,y)$ for that decision is zero. Otherwise, if the system is not confident about its decision (i.e., $|y| < \infty$), then a specific cost is given by (22).

Laplace function to model the likelihoods: From the cost function introduced previously Eq. (21), we have employed Bayes' rule to estimate the posterior probabilities $p(H_k|\mu_{i,j})$ and adopted a statistic distribution (Laplace) to model the likelihoods $p(\mu_{i,j}|H_0)$ and $p(\mu_{i,j}|H_1)$. To be more specific, the posterior probabilities $p(H_k|\mu_{i,j})$ with $k \in \{0,1\}$ are computed based on Bayes' rule, as follows:

$$p(H_k|\mu_{i,j}) = \frac{p(\mu_{i,j}|H_k)p(H_k)}{p(\mu_{i,j})} \tag{27}$$

where $p(H_k)$ is the prior probability of the corresponding hypothesis, $p(\mu_{i,j}|H_k)$ is the probability of finding the evidence score $\mu_{i,j}$ given the hypothesis H_k, and $p(\mu_{i,j})$ is the prior probability of $\mu_{i,j}$.

The likelihoods $p(\mu_{i,j}|H_0)$ and $p(\mu_{i,j}|H_1)$ can be modeled based on the fact that $p(\mu_{i,j}|H_0)$ (*resp.* $p(\mu_{i,j}|H_1)$) is the density of $\mu_{i,j}$ under the condition of the hypothesis H_0 (*resp.* H_1). In addition, as document content is mainly composed of foreground and background information, it is expected that $p(\mu_{i,j}|H_0)$ peaks at $\mu_{i,j} = \mu_F$ and converges to zero for the values that are further away from μ_F. Similarly, $p(\mu_{i,j}|H_1)$

peaks at $\mu_{i,j} = \mu_B$ and converges to zero elsewhere. We experimentally modeled these distributions by using Gaussian and Laplace functions, and it was found that better results are achieved with the Laplace model. In addition, the Laplace model provides the benefits of efficient computation in the subsequent process of our text block reconstruction algorithm. Hence, we have adopted the Laplace model to characterize the two likelihoods of (28) and (29):

$$p(\mu_{i,j}|H_0) = \exp\{-k_1|\mu_{i,j} - \mu_F|\} \tag{28}$$

$$p(\mu_{i,j}|H_1) = \exp\{-k_2|\mu_{i,j} - \mu_B|\} \tag{29}$$

where k_1 and k_2 are the parameters controlling the marginal spreads of the corresponding densities. Figure 4 shows the marginal densities of $p(\mu_{i,j}|H_0)$ and $p(\mu_{i,j}|H_1)$ with respect to $k_1 = k_2 = 12$, $\mu_F = 0.3$ and $\mu_B = 0.8$.

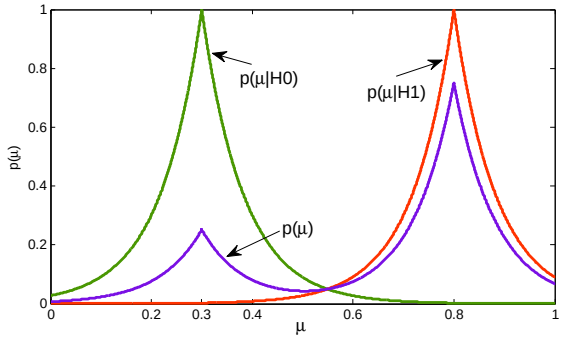


Fig. 4. The marginal densities of $p(\mu_{i,j}|H_0)$, $p(\mu_{i,j}|H_1)$ and $p(\mu_{i,j})$.

The marginal density of $p(\mu_{i,j})$ is then derived from $p(\mu_{i,j}|H_0)$ and $p(\mu_{i,j}|H_1)$ by (30):

$$p(\mu_{i,j}) = p(\mu_{i,j}|H_0)p(H_0) + p(\mu_{i,j}|H_1)p(H_1)$$

$$= \exp\{-k_1|\mu_{i,j} - \mu_F|\}p(H_0) + \exp\{-k_2|\mu_{i,j} - \mu_B|\}p(H_1) \tag{30}$$

Here, the prior probabilities $p(H_0)$ and $p(H_1)$ are chosen to model the unbalanced distribution of the foreground and background in document content. Generally speaking, the background pixels make up a large part of a document's content. Figure 4 shows the marginal density of $p(\mu_{i,j})$, in which $p(H_0) = 0.25$ and $p(H_1) = 0.75$. As seen, $p(\mu_{i,j})$ is modeled as a bimodal function, representing the fact that most of the pixels of a text block X_b are distributed at two predominant intensities μ_F and μ_B. However, the background pixels occur much more frequently than the foreground pixels. The bimodal function was also employed in [17] to construct a text model, but it assumes that the distribution of foreground and background is balanced and that every pixel value must fall inside the range between the foreground and background.

Once the prior probabilities and likelihoods are determined, the posterior probabilities are simply estimated by using (27). Figure 5 shows the marginal densities of the posterior probabilities $p(H_0|\mu_{i,j})$ and $p(H_1|\mu_{i,j})$ with respect to the parameter setting aforementioned.

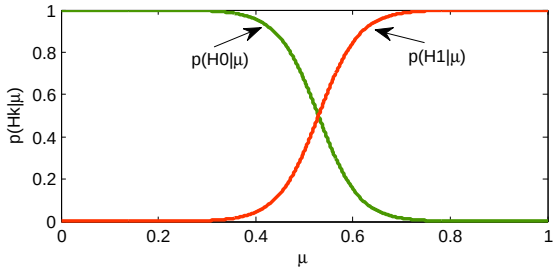


Fig. 5. The marginal densities of $p(H_0|\mu_{i,j})$ and $p(H_1|\mu_{i,j})$.

B. Effective text block reconstruction

To reconstruct the text blocks, we seek for the optimal values $\hat{\mu}_{i,j}$ such that the total error cost $C_{log}(\hat{\mu})$ is minimized subject to the constraint that the final solution $\hat{\mu}_{i,j}$ must be driven from the currently observed data $\mu_{i,j}$. In doing so, we propose a post-processing algorithm whose process is evolved based on the two main phases. The first phase makes a better estimate of the posterior probabilities by using the currently observed evidence scores while accounting for the close interaction among the pixels in a small neighborhood. The second phase updates the log-likelihood ratios and evidence scores by using the newly obtained posterior probabilities. The whole process is then repeated a number of iterations until the desired convergence is obtained. In what follows, it is assumed that the initial intensity values $\mu_{i,j}$ are computed by using the image decoded by the JPEG scheme. At each iteration $t = 1, 2, \ldots$, the post-processing algorithm performs the following main tasks:

- Step 1: Initializating the parameters:
  - Use Otsu's method [28] to automatically find the two dominant intensities μ_F and μ_B of the considered image block.
  - Initialize the prior probabilities $p(H_0)$ and $p(H_1)$ based on the histograms of the foreground and background values.
  - Initialize the posterior probabilities $p^{(t)}(H_k|\mu_{i,j})$ by using Equation (27) with $k \in \{0, 1\}$.
- Step 2: Estimate the new posterior probabilities $p^{(t+1)}(H_k|\mu_{i,j})$ for each pixel (i, j). In practice, due to the close interaction of the point (i, j) with other points in a small neighborhood, the categorical property (i.e., foreground or background) of the location (i, j) is highly correlated to that of its neighbors. Hence, the posterior probabilities must be estimated while taking into account the information observed from the neighbors of (i, j). Let $R_{i,j}$ be the set consisting of K neighbors of the point (i, j), the posterior probabilities $p^{(t+1)}(H_k|\mu_{i,j})$ are estimated as a fusing score as follows:

$$p^{(t+1)}(H_k|\mu_{i,j}) = \alpha p^{(t)}(H_k|\mu_{R_{i,j}}) + (1-\alpha)p^{(t)}(H_k|\mu_{i,j}) \tag{31}$$

with the context probabilities $p^{(t)}(H_k|\mu_{R_{i,j}})$ defined by:

$$p^{(t)}(H_k|\mu_{R_{i,j}}) = \frac{S^{(t)}_{k,R_{i,j}}}{S^{(t)}_{k,R_{i,j}} + S^{(t)}_{1-k,R_{i,j}}} \tag{32}$$

where

$$S^{(t)}_{k,R_{i,j}} = p(H_k) \prod_{(u,v)\in R_{i,j}} p^{(t)}(\mu_{u,v}|H_k).$$

In our implementation, we consider a 4-point neighborhood system, i.e., $K = 4$ and $R_{i,j} = \{(i, j-1), (i, j+1), (i-1, j), (i+1, j)\}$. The fusing weight $\alpha = 0.5$ is used by default.

- Step 3: Update the confidence score $y_{i,j}$ by using Bayes' rule:

$$y_{i,j} \leftarrow \text{logit}(p^{(t+1)}(H_0|\mu_{i,j})) - \text{logit}(p(H_0)) \quad (33)$$

- Step 4: Update $\mu_{i,j}$ using the newly derived $y_{i,j}$. By taking the inverse logistic transform of (23) in accordance with (28) and (29), we have:

$$y_{i,j} = k_2|\mu_{i,j} - \mu_B| - k_1|\mu_{i,j} - \mu_F| \quad (34)$$

Generally, solving (34) may lead to more than one solution and the one that is closest to the original value is selected.

- Step 5: Update the new error cost:

$$C^{(t+1)}_{log}(\mu) = \sum_{(i,j)} \sum_{k=0}^{1} p^{(t+1)}(H_k|\mu_{i,j}) C'_{log}(H_k, y_{i,j})$$

- Step 6: Increment t by one and repeat the above steps for a number of iterations or until the desired converge is reached.

To tolerate the smoothness among the neighboring blocks, the Otsu's method (i.e., Step 1) is applied to a local window that is positioned at the block center and has a size of 12×12. In addition, if a text block is in proximity to some smooth blocks, a non-local strategy can be applied to correctly compute the background value μ_B of that text block. The value μ_B can be, for example, chosen as the median intensity value among those of the neighboring smooth blocks.

C. Convergence analysis

In this section, we provide a brief discussion of the convergence of the proposed text block decoding algorithm. The underlying point here is to show that the total error cost $C_{log}(\mu)$, as reformulated by (26) in terms of probability entropy, is getting dropped as the algorithm evolves.

Recall that, at each iteration of the algorithm, the posterior probabilities are averaged from the current one and the ones in a small neighborhood. As such, the total variation of the posterior probabilities are alleviated and become more and more smoothed after each iteration. In other words, the correlation between each pixel and its neighbors is incremented from the posterior probability's point of view. This process not only makes the pixels more correlated but also diminishes the randomness of information: the more the algorithm evolves, the less random the information is. As a result, the total probability entropy of the image is decreased from time to time and the algorithm would converge to a fixed point after a number of iterations.

Although the convergence is achieved after a finite number of iterations, it was found that satisfactory results can be obtained just in the very first iteration. Consequently, the proposed algorithm runs into a non-iterative fashion. This finding is extremely desired for our target of supporting real time applications on 3G/4G mobile platforms. As shown in the experiment section, the proposed approach gives promising decoding results at a low cost of computational complexity.

VII. Experimental results

A. Experimental settings

The proposed approach is evaluated against four other JPEG artifact post-processing methods, including the classical JPEG decoder [5], morphological post-processing JPEG document decoder [16] (i.e., "*Mor*" for short), total variation (TV) method [10] and sparse representation using a learned dictionary (i.e., "*Dic*" for short) [14]. The former two methods, as well as our system, are all implemented in the C++ platform, and the latter two baseline methods are run in Matlab 2012a. The TV and *Dic* methods are selected for experiments, as they are considered as the state-of-the-art *iterative* decoding schemes. The *Mor* method is dedicated to the artifact post-processing of JPEG document content. It assumes that document images have been segmented into a uniform background and homogeneous foreground. Then, Otsu's method is applied to find the representative intensity value of the background (i.e., the most frequent one). Next, the cleaning step proceeds by assigning the representative intensity to all of the noisy pixels in the background region. A last step of sanity checking is performed in the DCT domain to prevent over-cleaning.

The public dataset Medical Archive Records (MAR) from the U.S. National Library of Medicine[1] is selected for our performance evaluation. This dataset consists of 293 real documents, scanned at 300dpi resolution, covering different types of biomedical journals and thus could be helpful to evaluate the performance of the methods on real scenarios. In addition to these journal documents, we also include a specific type of administrative documents collected by ITESOFT[2] company. This dataset has been used in several works for document image quality assessment [34]. Figure 6 shows several thumbnails of these images.

Following the conventional evaluation protocol in the literature [10], [14], Peak Signal-to-Noise Ratio (PSNR) is used as the evaluation criterion for assessing the goodness and robustness of the methods. Formally, the PSNR of two images I_1 and I_2 having the same size $M \times N$ is computed by (35):

$$PSNR(I_1, I_2) = 10 \log_{10} \left(\frac{255^2}{MSE} \right) \quad (35)$$

where MSE is the mean squared error, which is computed as follows:

$$MSE = \frac{1}{M \times N} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} (I1(m,n) - I2(m,n))^2$$

Because the TV and *Dic* methods are too costly to run on the full dataset (i.e., approximately 40 to 50 minutes for decompressing each image), the PSNR is thus computed for

[1] http://marg.nlm.nih.gov/

[2] http://www.itesoft.com/

a small subset consisting of 10 images randomly selected from the full dataset. All of our experiments are conducted on the following machine configuration: Windows 7 (64-bit), Intel Core i7-4600U (2.1 GHz), 16Gb RAM. Finally, Table IV reports the parameters used in the proposed approach. The choice of these values shall be thoroughly analyzed at the end of this section.

TABLE IV
PARAMETER SETTING IN THE PROPOSED APPROACH

Parameter	Value	Description
T_{seg}	15	Block classification threshold (Section IV)
λ	8	Lagrange constant for smooth block optimization
α	0.5	Defined in Equation (31)
k_1	20	Defined in Equation (28)
k_2	21	Defined in Equation (29)

B. Results and discussion

As the proposed decoding algorithm performs block decoding based on the block classification results, we first provide a few results and analysis of the block segmentation process. In the literature [15], [17], image blocks are often classified into three groups: background, picture, and text blocks. In all these works, the AC-based energy is a common criterion to differentiate the background blocks from the rest. While the background and text blocks can be handled effectively by designing specific decoding algorithms, the picture blocks present little interest in document content and are often left unchanged [17]. In the current work, we also restricted our approach to work on smooth block (e.g., background) and non-smooth blocks (e.g., text/graphics), that is often the case for a wide range of document content, from text documents (e.g., books, papers) to administrative documents (bank cheques, forms, receipts). Block classification thus enters in a traditional binary classification problem with a precision/recall evaluation protocol. In our context, it is preferred not to missing 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 mis-classified as text/graphics blocks (i.e., false positives) except the rise of computational overhead. Consequently, the parameter T_{seg} is favorably set to a relative low value. In Figure 7, we provide some visual results of block classification. Here, the smooth blocks are presented in white value and the non-smooth blocks are in dark gray color. As can be seen, the segmentation results look quite satisfactory although the AC-based energy is a conceptually simple criterion. The obtained smooth blocks mainly corresponds to the background regions, whereas the non-smooth blocks explain for the text elements and graphics content. As will be seen latter in the parameter setting part, more detailed analysis on the sensitivity of the parameter T_{seg} shall be discussed.

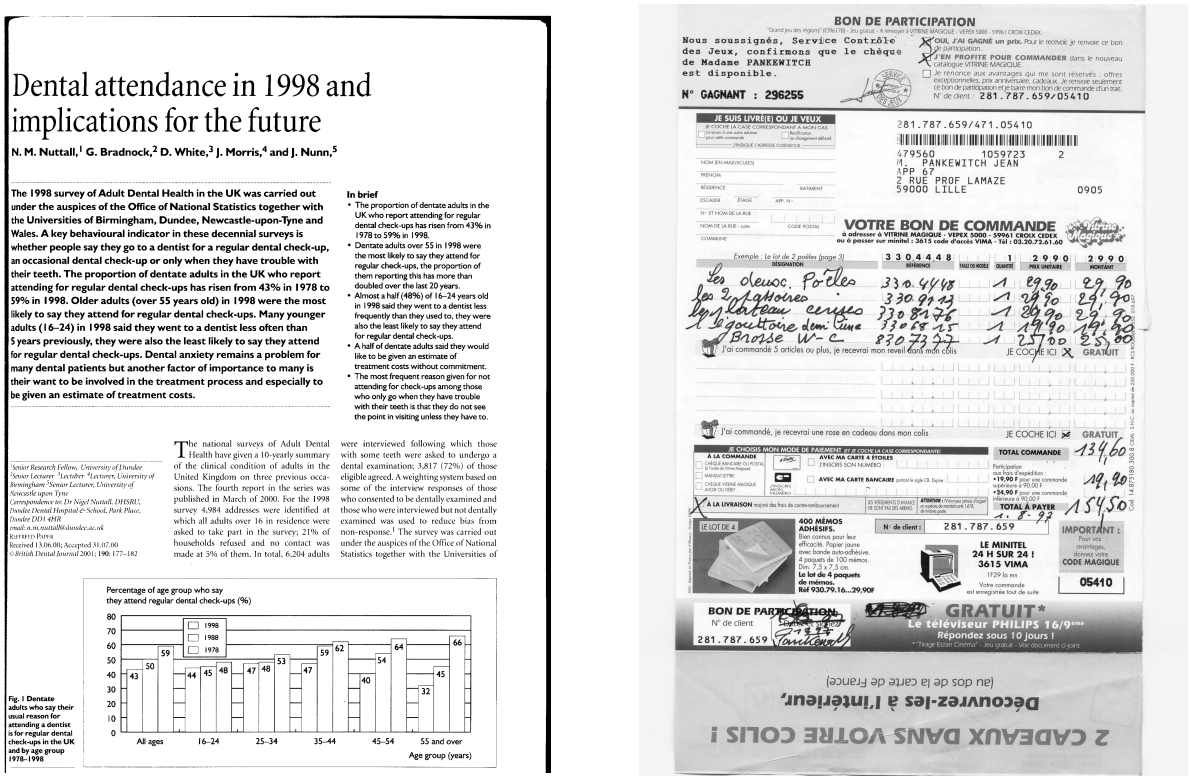
Next, we present the comparative results of all the studied methods in terms of PSNR score. Figure 8(a) shows the PSNR results of five methods for ten images randomly selected from the dataset. Each image is encoded at five qualities (i.e., $\{2, 4, 6, 8, 10\}$) and then the overall bit-rates are averaged, using uniform interval, for the ten images to create the following data points of bit-rate: $\{0.154, 0.190, 0.226, 0.262, 0.298, 0.334, 0.370, 0.406, 0.442, 0.478\}$. The PSNR measures are finally averaged via these data points. For detail of this procedure, readers are referred to the Appendix accompanying with this paper.

As seen, the proposed method significantly outperforms all of the others at every bit-rate, especially when the bit-rate is sufficiently high (i.e., > 0.25). On average, the proposed method gives an improvement of 2.1 (dB) compared with the conventional JPEG decoder. In contrast, the *Dic* and TV schemes give slightly better results than the JPEG algorithm. The TV method is even outperformed by the JPEG decoder at low bit-rates (i.e., < 0.22). These results may indicate that the two computational intensive methods are not well suited to handle JPEG document artifacts, although they are considered as the state-of-the-art decoding schemes for natural images.

The *Mor* method works reasonably well when considering the fact that it is a non-iterative method that was specifically developed to have the benefit of low computational cost. It is worth highlighting that the MAR dataset is composed of binary document images. Hence, the images have perfectly uniform backgrounds and homogeneous foregrounds, which is a key assumption for the *Mor* method. Furthermore, the sanity check step of the *Mor* method is always kept active to ensure that better decoding results are always obtained. However, the quality improvement of the *Mor* method is still less than half of that of our approach. This is because the *Mor* algorithm performs artifact cleaning only on the background pixels. Hence, the distortion appearing inside the text and around the text's edges is not treated. The outstanding results of our approach evidently confirm the robustness and goodness of the proposed text models for document content. The results in Figure 8(a) also reveal that the ten selected images have varying document content such that the PSNR at the bit-rate of approximately 0.4 (bpp) is even lower than that at the bit-rate of 0.3 (bpp).

Figure 8(b) additionally provides the PSNR measures of our approach, the *Mor* method and the JPEG decoder for the full MAR dataset. We observe the same behavior of the three methods as before. The proposed approach is highly superior to the JPEG decoder, with a quality improvement of up to 2.3 (dB) when the bit-rate is higher than 0.3 (bpp). At lower bit-rates, the quality gain is smaller yet still remarkably high. This observation is featured by the fact that, at low bit-rate coding, much of the information is eliminated in addition to the incorporation of spurious artifacts. The proposed approach handles very well the spurious artifacts (i.e., true negatives) but is less effective at reconstructing the original information (i.e., true positives). This situation also happens to many other coding artifact post-processing schemes. Figure 8(a) experimentally accounts for this proposition, where the *Dic* and TV methods do not significantly outperform the JPEG algorithm for bit-rates < 0.25 (bpp).

Figure 9 visually shows the decoding results of the proposed method in comparison with the JPEG algorithm. The top row presents several original text images. The middle row shows the results obtained by the JPEG decoder, and the bottom row



(a) (b)

Fig. 6. Examples of images used in our experiments: (a) binary journal document, (b) gray-scale administrative document.

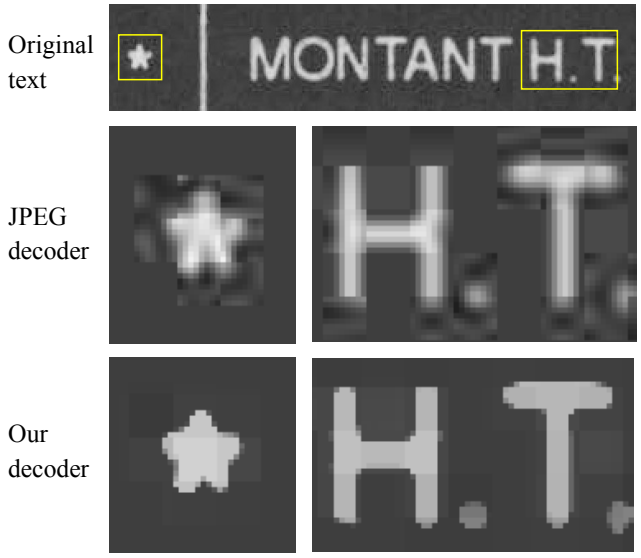
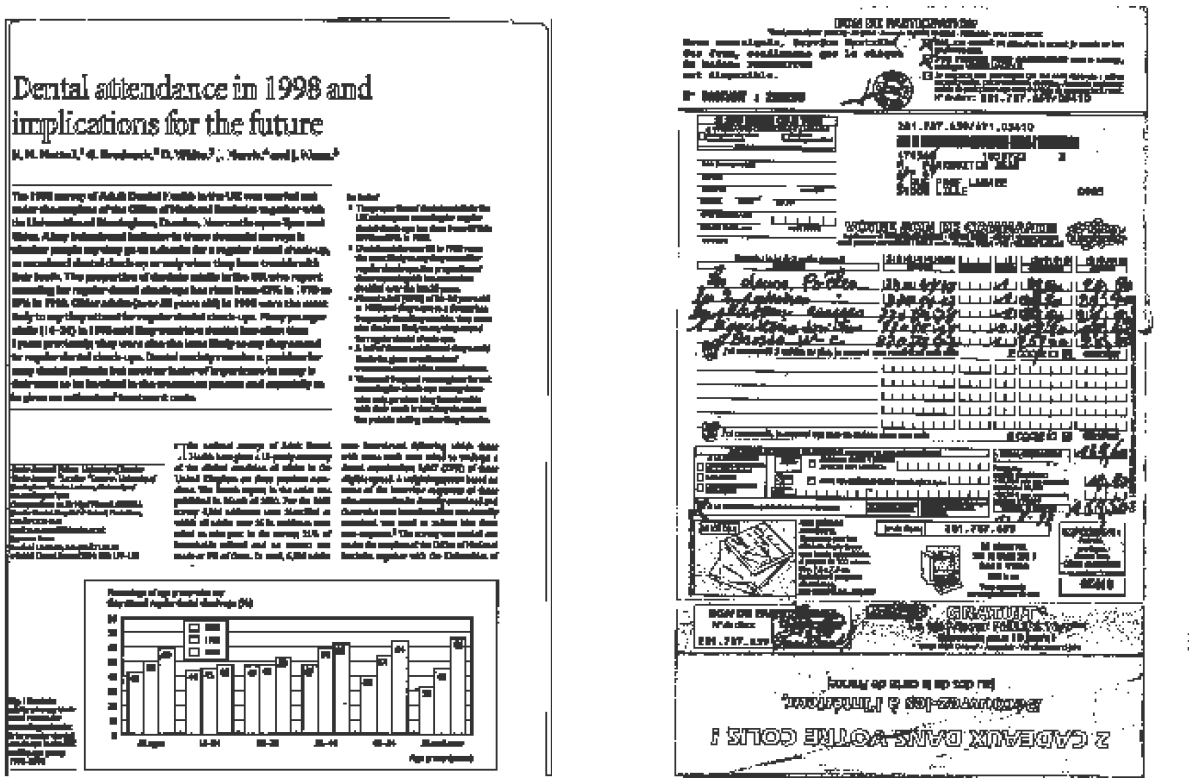


Fig. 10. Illustration result for a grayscale and noisy image (quality = 9): original image (top row), the result decoded by JPEG scheme for the clipped parts (middle row), and the result of the proposed approach (bottom row).

plots the results of our decoder. Here, the original images are encoded at the quality of 3. It can be seen that a severe degree of ringing and blocking artifacts is added to the JPEG decoding results, which really makes the visual perception of documents become annoying. In contrast, these artifacts are greatly removed in the results decoded by our scheme. Most of the foreground and background values are correctly reconstructed by the proposed approach. The obtained results are very close to the original texts, even though the considered compression rate is quite high.

Figure 10 demonstrates the decoding results for a grayscale and noisy image for visual inspection. The original image is depicted in the top row, and the decompression results for the two clipped portions are shown in the middle and bottom rows with respect to the results of the JPEG decoder and our proposed approach. The coding quality is set to 9 in this experiment. It is easily seen that the text decoded by the JPEG algorithm is heavily distorted by blocking artifacts for all of the regions inside each character. In addition, the ringing artifacts appear around the edges of the text and of the asterisk symbol. When compared with the decoding result of our approach, both ringing and blocking artifacts are nicely treated. The intensity values inside the characters and the asterisk are homogeneously dominated, making them insensitive to blocking artifacts. The spurious details around the text's edges are also smoothed out so that they are homogeneous with the background. It is worth mentioning that the proposed approach does not incur the blurring effect for the decoded results. It still produces sharp transitions between the text and background.

We also provided additional results in Figures 11-12 to highlight the robustness of the proposed method when dealing with heavy distortion of blocking and ringing artifacts. In Figures 11(a) and 12(a), blockiness is clearly visible in the results of the JPEG algorithm, making the image inadmissible



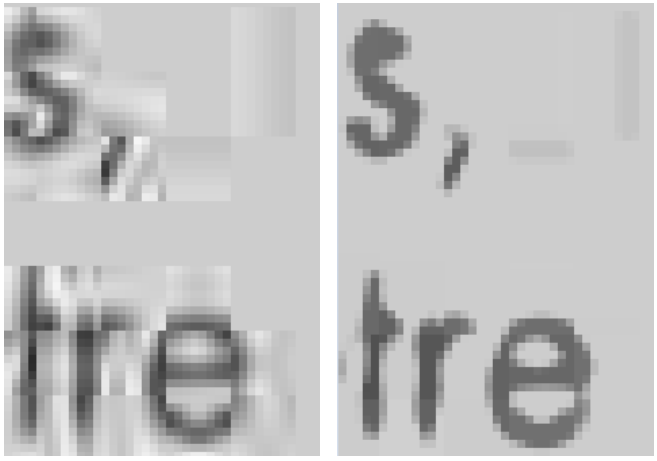
(a) (b)

Fig. 7. Block classification results for the images in Figure 6 (a,b) with quality= 9: white for smooth blocks and dark gray for non-smooth blocks.

for visual perception. Here, the artificial blocks are appearing densely around the smooth regions as well as the text regions, globally creating an annoying chessboard view. In contrary, the results of the proposed method, as presented in Figures 11(b) and 12(b), show that these unpleasant effects have been greatly removed, while adequately preserving finer details of the characters. Although there are still some subtle remaining trails of blockiness (e.g., small bar lines in the top-right of Figure 12(b)), the visual quality of the image has been substantially improved.

The proposed approach works effectively not only for text images but also for graphical content. Figure 13 experimentally justifies this point. The original image is plotted in Figure 13(a). After compressing the image at the quality of 9, part of the decoding result is enlarged in Figure 13(b), which corresponds to the JPEG decoder, and (c), which corresponds to the proposed approach. Again, blocking and ringing artifacts are clearly visible in the JPEG decoding result. In contrast, the proposed approach successfully separates the (dark) bar lines from the background. The bar lines are uniformly reconstructed while maintaining finer details at the contour locations. Hence, the artifacts are mostly invisible in the decoding result.

Lastly, we experimentally investigate the impact of decoding a mixed picture and text document. Figure 14 gives an exemplified image where the characters are embedded in the background picture. As for block segmentation, most of the image blocks in this image are classified as text blocks, although they can be actually considered as picture blocks or



(a) (b)

Fig. 12. Deblocking result (quality = 9): (a) JPEG decoder, (b) the proposed approach.

smooth blocks. The decoding result of the proposed method is presented in Figure 14(b) where we can observe that the image details tend to be more sharpening. The ringing artifact occurring in proximity to the characters is eliminated, while the blockiness is lessened in the picture region. However, the finer details appearing on the face and the texture cloth of the lady are not completely reconstructed. This shows the limitation of the proposed method when working on texture regions or multi-model image content. In that case, specific deringing

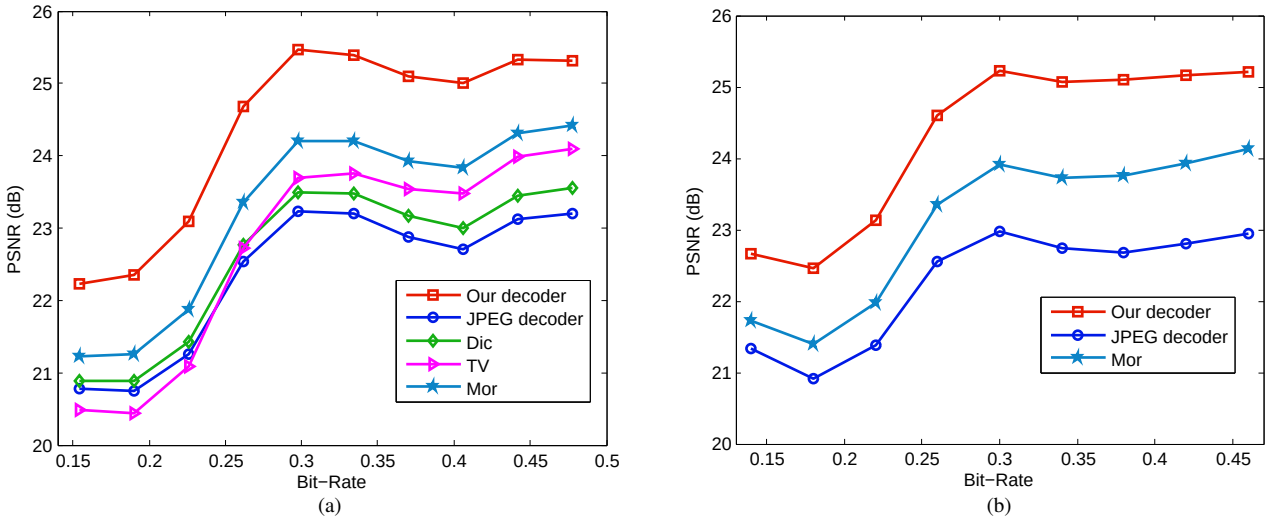


Fig. 8. The PSNR results of all the studied methods: (a) the results for 10 randomly selected images, (b) the results for the full dataset.

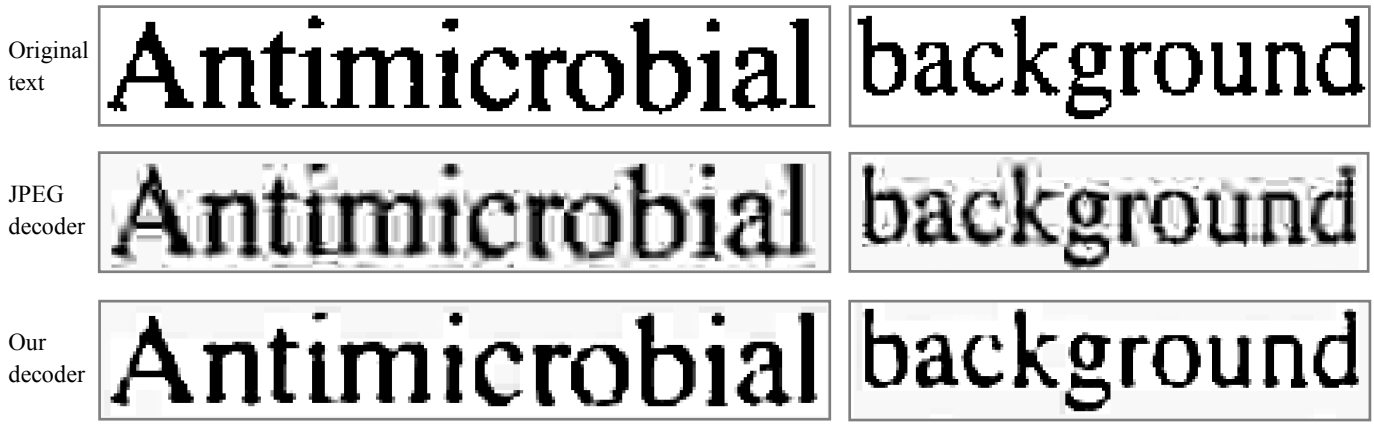


Fig. 9. Illustration results for several clipped binary text images: original text (top row), the results of the JPEG decoder (middle row) and the results of the proposed approach (bottom row).

methods for natural images (e.g., total variation regularization, sparse representation) would be more appropriate to handle such issues.

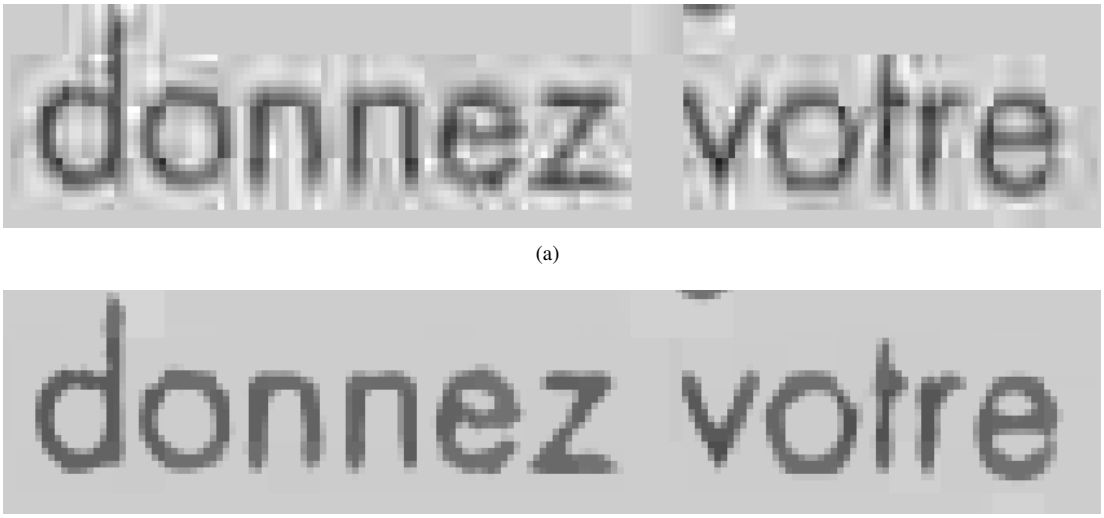
C. Running time analysis

This section provides an evaluation of decoding time for four methods, including the TV method, the proposed method, the *Mor* method and the baseline JPEG scheme. Specifically, the processing time is computed under the assumption that each method takes as an input the quantized DCT coefficients of the compressed image. Hence, the JPEG's decoding time concerns the computational cost of the inverse DCT transform only. This serves as a baseline benchmark for the other methods. All of the methods are run on a CPU machine without parallel implementation. Because the TV and *Mor* methods are implemented in Matlab, it is difficult to compare directly these methods with the others. Fortunately, the C++ version of the TV method is available in the original paper, accompanied with the processing time. Hence, we have reproduced the decompression time of the TV method reported in the original paper [10].

TABLE V
A REPORT ON DECOMPRESSION TIME (MS)

Image size	TV method	Our method	*Mor* method	JPEG
512×512	588	20	12	4
1600×1200	4459	130	71	10
4272×2848	28257	690	424	20

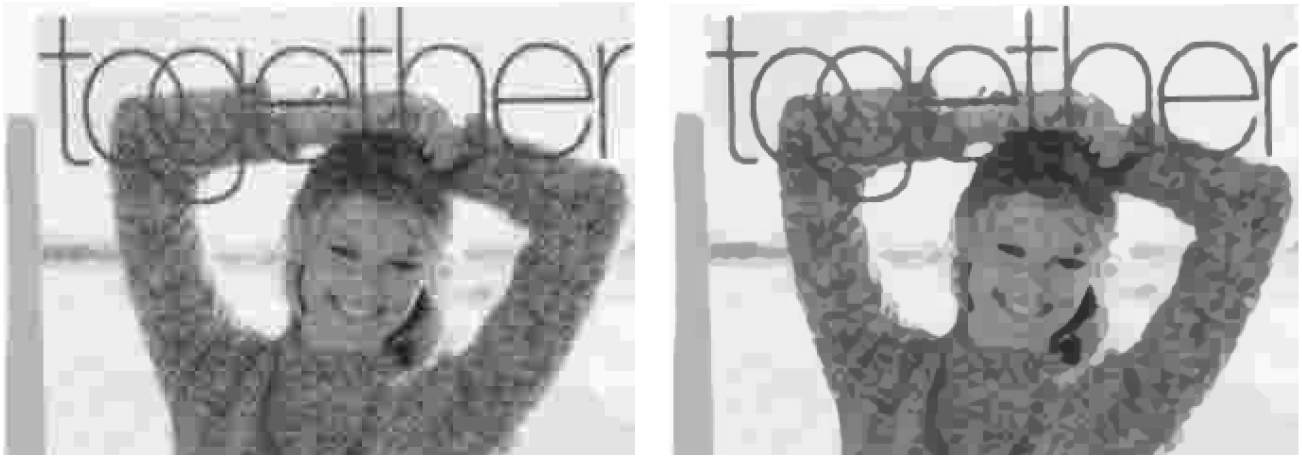
Table V gives a comparative result of the processing time of the studied methods for three different image sizes. As expected, the *Mor* decoder works very efficiently because it was developed as a non-iterative and simple computation method. The proposed approach is more computationally intensive than the *Mor* method, but it still has a very low computational cost. For instance, the decoding time of the proposed method for a quite large grayscale image (e.g., 4272×2848) is 690 milliseconds (ms) when compared with 424 (ms) of the *Mor* method. This obtained result is very



(a)

(b)

Fig. 11. Robustness of the proposed method for dealing with both blocking and riniging artifacts (quality = 9): (a) JPEG decoder, (b) the proposed approach.



(a) (b)

Fig. 14. Result of decoding a mixed text and picture image (quality = 9): (a) JPEG decoder, (b) the proposed approach.

promising for the proposed approach when considering the fact that our PSNR improvement is two times higher than that of the *Mor* method (see Figure 8(b)). The TV decoder is undoubtedly the most computationally expensive method due to its iterative process of switching back and forth between the compression and spatial domains. Its processing time, for instance, is approximately 28.2 (seconds) for decoding a 4272×2848 image.

D. Impact of parameter setting

We are now studying the robustness of the proposed approach according to different configurations of parameter setting. The studied parameters are reported in Table IV, including T_{seg}, λ, α, k_1 and k_2.

The first parameter, T_{seg}, served as the thresholding value to classify an image block into either smooth block or non-smooth block. To justify the sensitivity of this parameter, we have varied T_{seg} in a wide range of $[10, 100]$ while the other parameters are left unchanged. Consequently, it was found that the obtained results are very stable, i.e., the maximum variation of PSNR is 0.0001 (dB). In addition, we also study the impact of T_{seg} when varying the quantization step size (i.e., the quality parameter). To this aim, the proposed algorithm is applied to the full MAR dataset at two very different qualities $q \in \{3, 30\}$ and with $T_{seg} \in \{0.01E_m, 0.1E_m, 0.5E_m\}$ where E_m is the maximum AC energy of each image. The results presented in Table VI show that at very low quality compression (e.g., $q = 3$), the variation of PSNR scores for the two lower settings of T_{seg} is negligible (e.g., 0.034 (db)). However, at higher value $T_{seg} = 0.5E_m$, PSNR score is remarkably reduced by 1.079 (db) because many text blocks are mis-classified as smooth blocks and hence not processed for deringing. Table VI also reveals that the impact of T_{seg} seems to be quite subtle when considering small quantization step size (i.e., $q = 30$). These results again verify our proposition that it is advised to choose a low value for T_{seg} in order to ensure a stable performance of the system. In that case, the system operates at low precision and strong recall

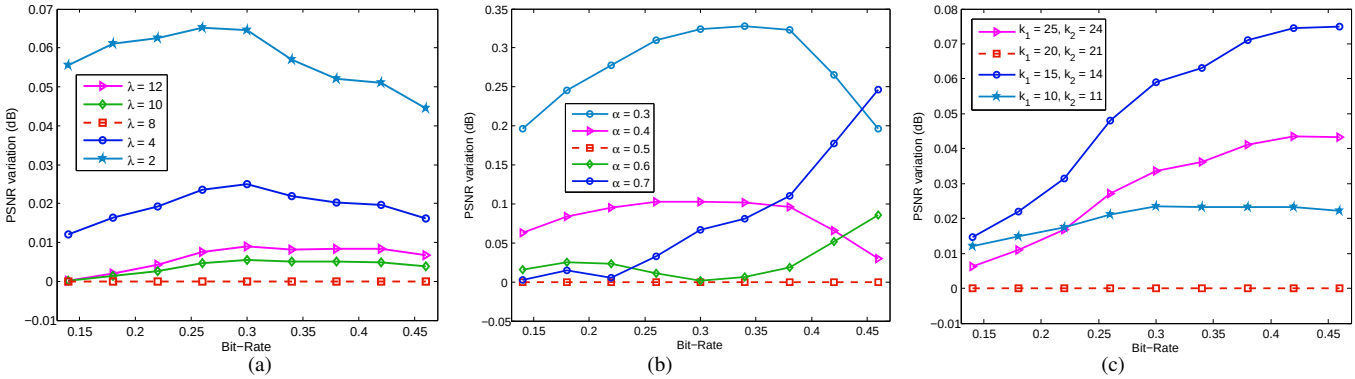


Fig. 15. Evaluation of parameter impact of the proposed approach: (a) parameter λ, (b) parameter α, and (c) parameters (k_1, k_2). The dash lines correspond to the default setting which is detailed in Table IV.

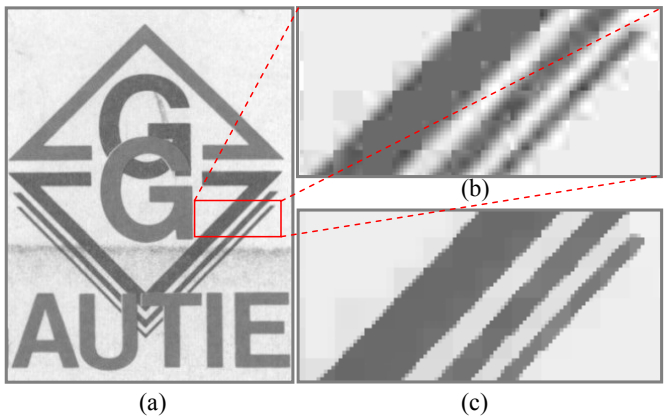


Fig. 13. Example results for graphical image (quality = 9): (a) original image, (b) magnified version of JPEG decoding result for the clipped part, (c) decoding result for the clipped part by the proposed approach.

for text blocks with little impact on PSNR results.

TABLE VI
AVERAGE PSNR SCORES FOR DIFFERENT SETTINGS OF T_{seg} AND q

	$T_{seg} = 0.01E_m$	$T_{seg} = 0.1E_m$	$T_{seg} = 0.5E_m$
$q = 3$	22.838	22.804	21.725
$q = 30$	28.662	28.644	28.623

Next, we discuss about the effect of setting the parameter λ. The choice of the parameter λ is a more challenging task for the TV-based methods. Generally, setting λ to a higher value would result in a fewer number of iterations and vice versa. This is because of the fast convergence of Newton's method, which is reliant on how close the optimal solution is to the observed data. Hence, we have studied the impact of λ by computing the average PSNR for the full dataset for each $\lambda \in \{2, 4, 8, 10, 12\}$. The maximum number of iterations is set to six. As shown in Figure 15(a), the maximum PSNR variation is 0.063 (dB) with respect to $\lambda = 2$ and at the bit-rate of approximately 0.26 (bbp). This small variation shows the robustness of the proposed algorithm to λ.

The last three parameters α, k_1 and k_2 are used in the text decoding process. We vary the parameter α in the range of $[0.3, 0.7]$ with the incremental step of 0.1 and the following combinations are set to the parameters (k_1, k_2): (10, 11), (15, 14), (20, 21), (25, 24). Figure 15(b) illustrates the impact of the parameter α in which we can observe that setting $\alpha = 0.3$ will result in the maximum PSNR variation of 0.33 (dB). Consequently, decent compromise of α would be in the range of $[0.4, 0.6]$ where the highest variation is approximately 0.11 (dB). Concerning the parameters k_1 and k_2, Figure 15(c) clearly shows that the proposed algorithm is quite robust to different settings of these two parameters, with the maximum variation of 0.075 (dB).

E. Comparison with advanced codecs

All the baseline methods presented afore-mentioned fall into the same class of post-processing DCT data in which they are designed to work only in the decoder's side without any information about the original signal. We are now going to verify how efficient are these methods when compared with novel coding technologies which are expected to be potential successors of JPEG standard. Among many advanced codecs (e.g., WebP[3], Mozjpeg[4], JPEG-XR[5]), High Efficiency Video Coding (HEVC) [35] is a very promising technology for video compression. HEVC has been designed to exploit effectively the spatial redundancy for both inter mode (i.e., between successive frames) and intra mode (i.e., within a single frame). Due to its excellent coding quality, HEVC has been adapted to still picture coding by using its intra mode. This is exactly what the emerging codec BPG (Better Portable Graphics)[6] does. A recent study of Mozilla[7] showed that HEVC (intra mode) or BPG is the best performer among many state-of-the-art codecs in use today.

In this part, we are comparing the performance of the proposed method over the BPG codec. The evaluation protocol must be driven in an objective way to highlight the difference in characteristics (e.g., post-processing versus pre-processing) of each method. To be more specific, the proposed method is

[3]https://developers.google.com/speed/webp/?csw=1
[4]https://github.com/mozilla/mozjpeg
[5]http://jpeg.org/jpegxr/index.html
[6]http://bellard.org/bpg/
[7]http://people.mozilla.org/∼josh/

a post-processing technique which is applied at the decoder's side and thus does not have any information in advance about the original image. In contrast, BPG falls into a pre-processing scheme and hence has complete prior knowledge about the clean signal, enabling it to perform both rate-distortion optimization at the encoder's side as well as post-processing at the decoder's side.

To address this difference, we propose a three-scenery evaluation protocol as detailed in Figure 16. In the first case, BPG is applied to original images and hence is different from our method in the input. For the two latter cases, we pass the same input to BPG and our method. This means, BPG takes as input the DCT data in JPEG files, encodes them with the best quality, and performs decompression to get back the results. This scheme is referred to as BPG-JPEG. Our method is applied, as usual, to the DCT data for post-processing the JPEG files.

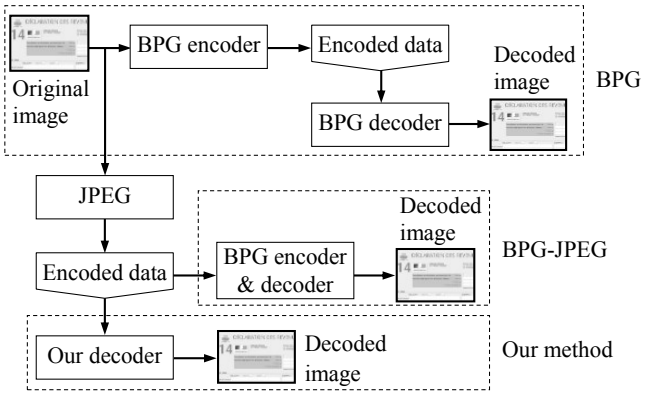


Fig. 16. Evaluation workflow for BPG and the proposed approach.

Figure 17 shows the PSNR scores of four methods, including BPG, BPG-JPEG, JPEG and the proposed method when applied to text documents of the MAR dataset. As seen, BPG gives very interesting results, while BPG-JPEG's performance is very close to that of the JPEG decoder. On average, BPG produces an improvement of up to 5.9 (db) and 4.2 (db) over the JPEG and the proposed method, respectively. However, the superiority of BPG comes mainly from the pre-processing phase (i.e., at the encoder's side) by its intensive rate-distortion optimization process and thus is subject to intensively computational cost (see Table VII)[8]. For instance, it takes BPG 3430 (ms) and 1190 (ms) for encoding and decoding a 300dpi binary document image (e.g., 2544×3296), respectively. When compared with JPEG[9], the processing times are 60 (ms) for compression and 110 (ms) for decompression. In other words, BPG makes a deceleration factor of around 57x and 11x over JPEG for compression and decompression, respectively. Although the proposed method takes a slight more overhead of computation for decompression, it is still very efficient compared with BPG. Figure 17 also verifies that BPG-JPEG gives almost no benefit when working on the decoder's side of JPEG data. This implies that a huge amount of JPEG files being in use widely today would not be benefited by using BPG. Lastly, some parts of HEVC codec used in BPG are patented. Therefore, to replace the current JPEG standard, it is needed to wait for the complete standardization of BPG or HEVC (intra mode). In the meanwhile, JPEG standard and well-optimized JPEG schemes shall continue to be used widely as argued by a recent study conducted by Mozilla[10].

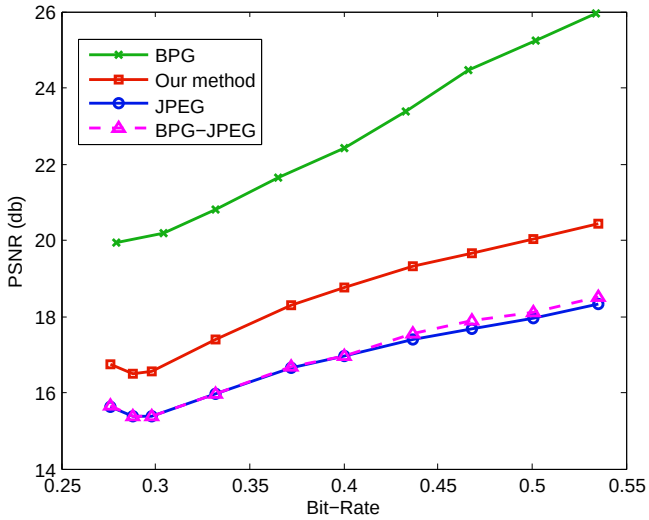


Fig. 17. PSNR results of BPG, BPG-JPEG, JPEG and the proposed method.

TABLE VII
PROCESSING TIME (MS) OF BPG, JPEG AND OUR METHOD

Task	BPG	JPEG	Our method
Compression	3430	60	60
Decompression	1190	110	430

VIII. CONCLUSIONS

In this paper, we have presented a novel approach for optimally decompressing the JPEG document images. First, the DCT blocks are classified into either smooth blocks or non-smooth blocks. Specific decoding algorithms are then developed for each type of block. For smooth blocks, we first introduce a fast technique to compute the total block boundary variation (TBBV) in the DCT domain. This efficient TBBV measure is used as an objective function to recover the smooth blocks. Reconstruction of non-smooth blocks is performed by incorporating a novel text model that accounts for the characteristics of the document content. The proposed approach has been validated through extensive experiments in comparison with other methods. Experimental results showed that the proposed approach gives a substantial improvement of visual quality while incurring a relative low computational cost.

Although the proposed approach gives quite interesting results, there is still room for further improvements. In this work, it is assumed that the picture blocks are not handled. Therefore,

[8]The fast algorithm of BPG is used, known as x265, version 0.9.6.
[9]JPEG's source code is available at: http://libjpeg.sourceforge.net/

[10]https://blog.mozilla.org/research/2014/03/05/introducing-the-mozjpeg-project/

exploiting well-established approaches dedicated to natural images (e.g., sparse representation, total variation regularization) for dealing with these blocks would be a potential direction. Additionally, fast implementation of the proposed algorithm with the inclusion of parallelized and vectorized computing will be investigated. Furthermore, extension of this work to color images should be considered as well. At last, the proposed text model assumes a bimodal function for the intensity distribution of each image block. Although this assumption is reasonable for a wide range of document content, it may not fit with multi-modal distribution (e.g., gradient texts, texture content). Therefore, novel post-processing methods to deal with coding artifacts for such specific document content would be a challenging and attractive problem for researchers.

APPENDIX
COMPUTING AVERAGED BIT-RATE AND PSNR

Given a set of N input images, namely I_1, I_2, ..., I_k, ..., I_N, we compress each image I_k at 5 different encoding qualities (i.e., $q \in \{2, 4, 6, 8, 10\}$). Let $C_{k,q}$ be the image obtained by compressing I_k using the quality q, the bit-rate and PSNR score computed from the original image I_k and its compressed version $C_{k,q}$ are denoted by $BR_{k,q}$ and $PSNR_{k,q}$, respectively. From all the pairs $(BR_{k,q}, PSNR_{k,q})$ obtained previously, we create a histogram of PSNR scores as follows:

- Initializating the minimum bit-rate ($minBR$), the uniform interval of bit-rate ($stepBR$), the number of bit-rate points ($numBR$), the occurrences of bit-rate ($count[]$) and the histogram of PSNR ($hist[]$):

$$\begin{aligned} minBR &\leftarrow 0.154 \\ stepBR &\leftarrow 0.036 \\ numBR &\leftarrow 9 \\ hist[i] &\leftarrow 0 \\ count[i] &\leftarrow 0 \end{aligned}$$

  where $i = 0, 1, 2, \ldots, 9$.
- For each pair $(BR_{k,q}, PSNR_{k,q})$ with $k = 1, 2, \ldots, N$ and $q \in \{2, 4, 6, 8, 10\}$, we update $hist$ and $count$ as follows:

$$\begin{aligned} index &\leftarrow \text{round}((BR_{k,q} - minBR)/stepBR) \\ hist[index] &\leftarrow hist[index] + PSNR_{k,q} \\ count[index] &\leftarrow count[index] + 1 \end{aligned}$$

- Finally, compute the uniformly spaced data points of bit-rate and the averaged PSNR scores:

$$\begin{aligned} dataPoints[i] &\leftarrow minBR + i * stepBR \\ hist[i] &\leftarrow hist[i]/count[i] \end{aligned}$$

  where $i = 0, 1, 2, \ldots, 9$.
- For display purpose, make a plot of $hist[i]$ over $dataPoints[i]$.

REFERENCES

[1] I. R. 2301, *File Format for Internet Fax*. L. McIntyre, S. Zilles, R. Buckley, D. Venable, G. Parsons, and J. Rafferty, March 1998. ftp://ftp.isi.edu/in-notes/rfc2301.txt.

[2] L. Bottou, P. Haffner, P. G. Howard, P. Simard, Y. Bengio, and Y. LeCun, "High quality document image compression with djvu," *Journal of Electronic Imaging*, vol. 7, no. 3, pp. 410–425, 1998.

[3] D. Huttenlocher, P. Felzenszwalb, and W. Rucklidge, "Digipaper: a versatile color document image representation," in *1999 Proceedings International Conference on Image Processing*, vol. 1, 1999, pp. 219–223.

[4] H. Cheng and C. A. Bouman, "Document compression using rate-distortion optimized segmentation," *Journal of Electronic Imaging*, vol. 10, no. 2, pp. 460–474, 2001.

[5] G. K. Wallace, "The jpeg still picture compression standard," *Communications of the ACM*, vol. 34, no. 4, pp. 30–44, 1991.

[6] Y. Yang, N. Galatsanos, and A. Katsaggelos, "Projection-based spatially adaptive reconstruction of block-transform compressed images," *IEEE Transactions on Image Processing*, vol. 4, no. 7, pp. 896–908, 1995.

[7] S. Yang, Y.-H. Hu, and D. Tull, "Blocking effect removal using robust statistics and line process," in *Multimedia Signal Processing, 1999 IEEE 3rd Workshop on*, 1999, pp. 315–320.

[8] J. J. Zou and H. Yan, "A deblocking method for bdct compressed images based on adaptive projections," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 15, no. 3, pp. 430–435, 2005.

[9] T. Kartalov, Z. Ivanovski, L. Panovski, and L. Karam, "An adaptive pocs algorithm for compression artifacts removal," in *Signal Processing and Its Applications, 2007. ISSPA 2007. 9th International Symposium on*, 2007, pp. 1–4.

[10] K. Bredies and M. Holler, "A total variation-based jpeg decompression model," *SIAM Journal on Scientific Computing*, vol. 5, no. 1, pp. 366–393, 2012.

[11] M. Elad and M. Aharon, "Image denoising via sparse and redundant representations over learned dictionaries," *IEEE Transactions on Image Processing*, vol. 15, no. 12, pp. 3736–3745, 2006.

[12] M. Protter and M. Elad, "Image sequence denoising via sparse and redundant representations," *IEEE Transactions on Image Processing*, vol. 18, no. 1, pp. 27–35, 2009.

[13] C. Jung, L. Jiao, H. Qi, and T. Sun, "Image deblocking via sparse representation," *Signal Processing: Image Communication*, vol. 27, no. 6, pp. 663–677, 2012.

[14] H. Chang, M. Ng, and T. Zeng, "Reducing artifact in jpeg decompression via a learned dictionary," *Transactions on Signal Processing (TSP)*, vol. 62, no. 3, pp. 718–728, 2013.

[15] E. Y. Lam, "Compound document compression with model-based biased reconstruction," *Journal of Electronic Imaging*, vol. 13, no. 1, pp. 191–197, 2004.

[16] B. Oztan, A. Malik, Z. Fan, and R. Eschbach, "Removal of artifacts from jpeg compressed document images," in *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, vol. 6493, Jan 2007, pp. 1–9.

[17] T. Wong, C. Bouman, I. Pollak, and Z. Fan, "A document image model and estimation algorithm for optimized jpeg decompression," *Transactions on Image Processing (TIP)*, vol. 18, no. 11, pp. 2518–2535, 2009.

[18] T. O'Rourke and R. Stevenson, "Improved image decompression for reduced transform coding artifacts," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 5, no. 6, pp. 490–499, 1995.

[19] X. Zhang, R. Xiong, X. Fan, S. Ma, and W. Gao, "Compression artifact reduction by overlapped-block transform coefficient estimation with block similarity," *Image Processing, IEEE Transactions on*, vol. 22, no. 12, pp. 4613–4626, 2013.

[20] A. Chambolle, "An algorithm for total variation minimization and applications," *Journal of Mathematical Imaging and Vision*, vol. 20, no. 1-2, pp. 89–97, 2004.

[21] K. Bredies, K. Kunisch, and T. Pock, "Total generalized variation," *SIAM Journal on Imaging Sciences*, vol. 3, no. 3, pp. 492–526, 2010.

[22] M. Nikolova, "Local strong homogeneity of a regularized estimator," *SIAM Journal on Applied Mathematics*, vol. 61, no. 2, pp. 633–658, 2000.

[23] M. Aharon, M. Elad, , and A. Bruckstein, "The k-svd: An algorithm for designing of overcomplete dictionaries for sparse representation," *IEEE Transactions on Signal Processing (TSP)*, vol. 54, no. 11, pp. 4311–4322, 2006.

[24] Y. C. Pati, R. Rezaiifar, and P. S. Krishnaprasad, "Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition," vol. 1, 1993, pp. 40–44.
[25] I. Popovici and W. Withers, "Locating edges and removing ringing artifacts in jpeg images by frequency-domain analysis," *Image Processing, IEEE Transactions on*, vol. 16, no. 5, pp. 1470–1474, 2007.
[26] S. Liu and A. Bovik, "Efficient dct-domain blind measurement and reduction of blocking artifacts," *Transactions on Circuits and Systems for Video Technology (TCSVT)*, vol. 12, no. 12, pp. 1139–1149, 2002.
[27] C. Park, J. Kim, and S. Ko, "Fast blind measurement of blocking artifacts in both pixel and dct domains," *Journal of Mathematical Imaging and Vision*, vol. 28, no. 3, pp. 279–284, 2007.
[28] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE Trans. on Systems, Man and Cybernetics*, vol. 9, no. 1, pp. 62–66, 1979.
[29] R. de Queiroz, "Processing jpeg-compressed images and documents," *Transactions on Image Processing (TIP)*, vol. 8, no. 12, pp. 1661–1672, 1998.
[30] D. M. Strong and T. F. Chan, "Edge-preserving and scale-dependent properties of total variation regularization," in *Inverse Problems*, 2000, pp. 165–187.
[31] H. S. Chang and K. Kang, "A compressed domain scheme for classifying block edge patterns," *Image Processing, IEEE Transactions on*, vol. 14, no. 2, pp. 145–151, Feb 2005.
[32] E. Feig and S. Winograd, "Fast algorithms for the discrete cosine transform," *IEEE Transactions on Signal Processing*, vol. 40, no. 9, pp. 2174–2193, 1992.
[33] N. Brummer and J. du Preez, "Application-independent evaluation of speaker detection," *Computer Speech and Language*, vol. 20, no. 2, pp. 230–275, 2006.
[34] A. Alaei, D. Conte, and R. Raveaux, "Document image quality assessment based on improved gradient magnitude similarity deviation," in *13th International Conference on Document Analysis and Recognition (ICDAR 2015)*, 2015, pp. 176–180.
[35] B. Bross, W.-J. Han, J.-R. Ohm, G. J. Sullivan, and T. Wiegand, "High efficiency video coding (hevc) text specification draft 9," in *Joint Collaborative Team on Video Coding (JCT-VC) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11, document JCTVC-K1003, Shanghai, China, Oct. 2012.*, 2012.

The-Anh Pham The-Anh Pham received a Masters (Research) with specialization in Computer Science from Vietnam National University, Hanoi in 2006 (Vietnam). He has then worked as a lecturer at Hong Duc University (Vietnam) since 2007. From 2010 to 2013, he fulfilled his PhD thesis in France. Since June 2014, he has been working in a full research fellow position at Polytech's Tours, France. His research interests include document image analysis, image compression, feature extraction and indexing, shape analysis and representation.

Mathieu Delalandre Mathieu Delalandre obtained his PhD Thesis in 2005 from the Rouen University (Rouen, France). Then, starting from 2006 until 2009, he has worked in different Research Fellow positions in laboratories and institutes Europe-wide: (Nottingham, UK), (Barcelona, Spain). Since September 2009, he has been an Assistant Professor at the LI laboratory (Tours, France) in the RFAI group. His ongoing research activities deal with the image processing field including object detection and template matching, local detectors and descriptors and processing in the transform domain. His application domains are related to Document Image Analysis including symbol and logo recognition, comics copyright protection, image networking and wine scanner.