A Document Image Model and Estimation Algorithm for Optimized JPEG Decompression

Tak-Shing Wong, Charles A. Bouman, *Fellow, IEEE*, Ilya Pollak, and Zhigang Fan

***Abstract*—The JPEG standard is one of the most prevalent image compression schemes in use today. While JPEG was designed for use with natural images, it is also widely used for the encoding of raster documents. Unfortunately, JPEG's characteristic blocking and ringing artifacts can severely degrade the quality of text and graphics in complex documents. We propose a JPEG decompression algorithm which is designed to produce substantially higher quality images from the same standard JPEG encodings. The method works by incorporating a document image model into the decoding process which accounts for the wide variety of content in modern complex color documents. The method works by first segmenting the JPEG encoded document into regions corresponding to background, text, and picture content. The regions corresponding to text and background are then decoded using *maximum a posteriori* (MAP) estimation. Most importantly, the MAP reconstruction of the text regions uses a model which accounts for the spatial characteristics of text and graphics. Our experimental comparisons to the baseline JPEG decoding as well as to three other decoding schemes, demonstrate that our method substantially improves the quality of decoded images, both visually and as measured by PSNR.**

***Index Terms*—Decoding, document image processing, image enhancement, image reconstruction, image segmentation, JPEG.**

I. Introduction

BASELINE JPEG [1], [2] is still perhaps the most widely used lossy image compression algorithm. It has a simple structure, and efficient hardware and software implementations of JPEG are widely available. Although JPEG was first developed for natural image compression, in practice, it is also commonly used for encoding document images. However, document images encoded by the JPEG algorithm exhibit undesirable blocking and ringing artifacts [3]. In particular, ringing artifacts significantly reduce the sharpness and clarity of the text and graphics in the decoded image.

In recent years, several more advanced schemes have been developed for document image compression. For examples, DjVu [4] and approaches based on the mixed raster content (MRC) model [5] are designed specifically for the compression of compound documents containing text, graphics and natural images. These multilayer schemes can dramatically improve on the trade-off between the quality and bit-rate of baseline JPEG compression. However, the encoding processes of these advanced schemes are also substantially more complicated than the JPEG algorithm. The simplicity of the JPEG algorithm allows many high performance and memory efficient JPEG encoders to be implemented. Such encoders enable JPEG to remain as a preferred encoding scheme in many document compression applications, especially in certain firmware based systems.

Many schemes have been proposed to improve on the quality of JPEG encoded images. One approach is to adjust the bit usage of the image blocks during encoding [6]–[8]. In this approach, the bit rate is adjusted in accordance to the content of the blocks so as to achieve better rate-distortion characteristics. However, although this approach usually improves the PSNR of the decoded image, it does not address the JPEG artifacts directly. Also, images which have been compressed cannot take advantage of these schemes. Alternatively, another approach applies postprocessing steps in the decoding process to suppress JPEG artifacts [9]–[15]. The schemes in [9], [10] reduce blocking artifacts by methods derived from projections onto convex sets (POCS). In [11], [12], prior knowledge of the original image is introduced in the decoding process with a Markov random field (MRF). The decoded image is then formed by computing the *maximum a posteriori* (MAP) estimate of the original image given the JPEG compressed image. Adaptive postfiltering techniques are suggested in [13]–[15] to reduce blocking and/or ringing artifacts in the decoded image. Filter kernels are chosen based on the amount of detail in the neighborhood of the targeted pixel to suppress JPEG artifacts without over-blurring details. A review of postprocessing techniques can be found in [16]. Still another approach requires modifications to both the encoder and the decoder. An example is given by the scheme in [17] which applies the local cosine transform to reduce blocking artifacts. Despite much work that has been done to improve the JPEG decoding quality, however, most of the schemes proposed are designed primarily for natural images rather than documents.

In this paper, we propose a JPEG decompression scheme which substantially improves the decoded image quality for document images compressed by a conventional JPEG encoder. Our scheme works by first segmenting the image into blocks of three classes: background, text, and picture. Image blocks of each class are then decompressed by an algorithm designed

Manuscript received February 24, 2009; revised June 21, 2009. First published July 24, 2009; current version published October 16, 2009. This research was supported in part by a grant from the Xerox Foundation. Part of this work has been presented at the 2007 IEEE Workshop on Statistical Signal Processing. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Mark (Hong-Yuan) Liao.

T.-S. Wong, C. A. Bouman, and I. Pollak are with the school of Electrical and Computer Engineering, Purdue University, West Lafayette, IN 47907 USA (e-mail: wong17@ecn.purdue.edu; bouman@ecn.purdue.edu; ipollak@ecn.purdue.edu).

Z. Fan is with the Xerox Research and Technology, Xerox Corporation, Webster, NY 14580 USA (e-mail: zfan@xeroxlabs.com).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIP.2009.2028252

1057-7149/$26.00 © 2009 IEEE

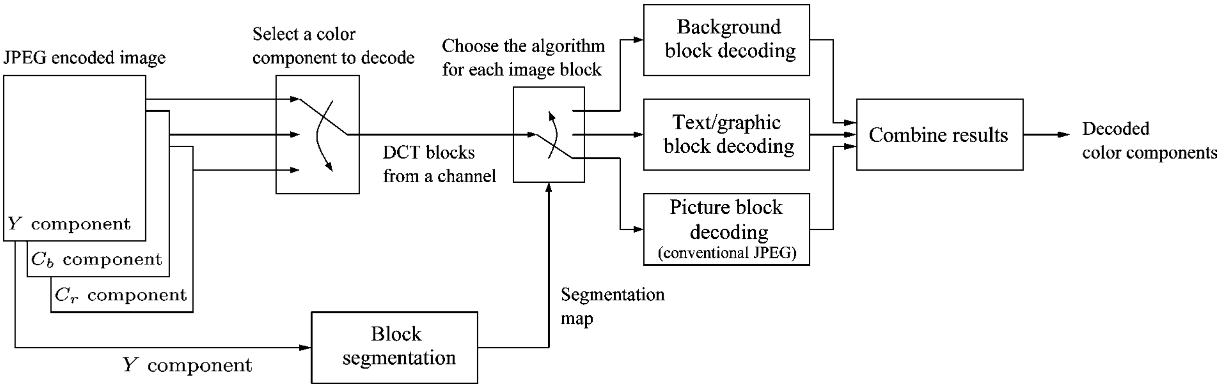

JPEG encoded image
Select a color component to decode
Choose the algorithm for each image block
Background block decoding
Text/graphic block decoding
Picture block decoding (conventional JPEG)
Combine results
Decoded color components
DCT blocks from a channel
Y component
C_b component
C_r component
Segmentation map
Block segmentation
Y component

Fig. 1. Overview of the proposed scheme. The luminance component is used to segment the JPEG compressed image into three classes of image blocks. The segmentation map is then used to determine the class of each block and to select the algorithm used to decode the block.

specifically for that class, in order to achieve a high quality decoded image. In particular, one important contribution of our work is the introduction of a novel text model that is used to decode the text blocks. Our text model captures the bimodal distribution of text pixels by representing each pixel as a continuous combination of a foreground color and a background color. During the decoding process, the foreground and background colors are adaptively estimated for each block. As demonstrated in Section VII, the text regions decoded with this text model are essentially free from ringing artifacts even when images are compressed at a relatively low bit rate.

The three classes of blocks used in our scheme have different characteristics and they suffer differently from JPEG artifacts. The background blocks correspond to the background of the document and smooth regions of natural images. Due to the smoothness of the background blocks, they are susceptible to the blocking artifacts. The text blocks are comprised of the text and graphic regions of the image. These blocks contain many sharp edges and they suffer most severely from the ringing artifacts. The remaining picture blocks consist of irregular regions of natural images. They suffer from both ringing and blocking artifacts. As noted in [18], the high-frequency content in these highly textured blocks makes the JPEG artifacts less noticeable. Thus, we simply use the conventional JPEG decoding to decode the picture blocks.

We describe the structure of our decoding scheme in Section II. For the luminance component, we then present the prior models used to decode the background blocks and the text blocks in Section III, and the MAP reconstruction algorithms in Section IV. We introduce our block based segmentation algorithm in Section V. Following this, in Section VI, we extend the decoding scheme to the chrominance components to address the low signal-to-noise ratio and low resolution commonly seen in the encoded chrominance components. Finally in Section VII, we present the experimental results and compare our scheme with three other existing JPEG decoding algorithms.

II. Overview of the Proposed Scheme

Under the JPEG encoding scheme, a color image is first converted to the $\mathrm{YC_bC_r}$ color space [19], [20], and the chrominance components are optionally subsampled. After this preprocessing, each color component is partitioned into nonoverlapping 8 × 8 blocks, and each block from the components undergoes the three steps of forward discrete cosine transform (DCT) [21], quantization, and entropy encoding. For an achromatic image, the preprocessing stage is omitted. The problem of JPEG decoding is to reconstruct the original image from the encoded DCT coefficients.

Fig. 1 shows the block diagram of our approach to JPEG decoding. First, the segmentation algorithm classifies the image blocks from the luminance component into three classes corresponding to background, text, and picture. Next, the color components of the JPEG image are decoded. For each color component, the segmentation map is used to determine the class of each block contained in the color component. Each block is then decoded with an algorithm designed to achieve the best quality for the given block class. After decoding the color components, the chrominance components are interpolated to the original resolution if they have been subsampled. Finally, the image in $\mathrm{YC_bC_r}$ color space is transformed to the desired output color space, usually sRGB [22].

We introduce our notation by briefly reviewing the achromatic JPEG codec. We denote random variables and vectors by uppercase letters, and their realizations by lowercase letters. Let X_s be a column vector containing the 64 intensity values of the block s. Then the DCT coefficients for this block are given by $Y_s = DX_s$, where D is the 64 × 64 orthogonal DCT transformation matrix. The JPEG encoder computes the quantized DCT coefficients as $\tilde{Y}_{s,i} = Q_i$ round $[Y_{s,i}/Q_i]$, where Q_i is a set of quantization step sizes. A typical JPEG decoder takes the inverse DCT of the quantized coefficients to form an 8 × 8 block of pixels $\tilde{X}_s = D^{-1}\tilde{Y}_s$. We also use $T(\cdot)$ to denote the quantization operation so that $\tilde{Y}_s = T(Y_s) = T(DX_s)$.

In our scheme, JPEG decoding is posed as an inverse problem in a Bayesian framework. This inverse problem is ill-posed because JPEG quantization is a many-to-one transform, i.e., many possible blocks X_s can produce the same quantized DCT coefficients $\tilde{Y}_s$. We regularize the decoding problem by developing a prior model for the original image and computing the MAP estimate [23] of the original image from the decoded DCT coefficients.

Specifically, for a particular preprocessed color component, the conditional probability mass function[1] of $\tilde{Y}_s$ given X_s is determined from the structure of the JPEG encoder as

$$p(\tilde{y}_s|x_s) = \begin{cases} 1, & \text{if } T(Dx_s) = \tilde{y}_s \\ 0, & \text{otherwise.} \end{cases} \tag{1}$$

Let X be the vector concatenating X_s of every block s from the color component, and let $\tilde{Y}$ be the vector of the corresponding quantized DCT coefficients. Then the probability of $\tilde{Y}$ given X is given by

$$\begin{aligned} p(\tilde{y}|x) &= \prod_s p(\tilde{y}_s|x_s) \\ &= \begin{cases} 1, & \text{if } T(Dx_s) = \tilde{y}_s \quad \text{for all} \quad s \\ 0, & \text{otherwise.} \end{cases} \end{aligned} \tag{2}$$

This forward model simply reflects the fact that for every block s, the quantized DCT coefficients $\tilde{Y}_s = \tilde{y}_s$ can be calculated deterministically given a specific set of pixel values $X_s = x_s$. If, moreover, X has the prior probability density $p(x)$, the MAP estimate for X based on observing $\tilde{Y} = \tilde{y}$ is then given by

$$\hat{x} = \arg\min_x \{-\log p(\tilde{y}|x) - \log p(x)\}.$$

Referring to (2), we see that the first term in the function we are minimizing, $-\log p(\tilde{y}|x)$, is either zero or ∞. Thus, we must ensure that the first term is zero in order to obtain a minimum. According to (2), this is accomplished by enforcing the constraints $T(Dx_s) = \tilde{y}_s$ for all s. In other words, our MAP solution must be consistent with the observed quantized coefficients. Therefore, the MAP estimate of X given $\tilde{Y}$ is the solution to the constrained optimization problem

$$\hat{x} = \arg\min_x [-\log p(x)] \quad \text{subject to} \quad T(Dx_s) = \tilde{y}_s \quad \text{for all} \quad s. \tag{3}$$

In practice, we solve the optimization problem (3) separately for the three classes of blocks. Let X^b, X^t, and X^p be the vectors of all pixels from the background, text, and picture blocks, respectively. The optimization problem for each class uses a prior model specific to the class. For the text blocks, we use a prior distribution $p(x^t|\phi)$ parameterized by a vector of hyperparameters ϕ, and compute the joint MAP estimate for X^t and ϕ by maximizing their joint probability density $p(x^t, \phi) = p(x^t|\phi)p(\phi)$. The optimization sub-problems for the background and text blocks are respectively given by

$$\hat{x}^b = \arg\min_{x^b} [-\log p(x^b)] \tag{4}$$

subject to $T(Dx_s) = \tilde{y}_s$ for all background blocks s, and

$$(\hat{x}^t, \hat{\phi}) = \arg\min_{x^t, \phi} [-\log p(x^t, \phi)] \tag{5}$$

subject to $T(Dx_s) = \tilde{y}_s$ for all text blocks s. For the picture blocks, we simply adopt the conventional JPEG decoding algorithm.

III. Prior Models for the Luminance Blocks

A. Prior Model for the Luminance Background Blocks

To enforce smoothness across the boundaries of neighboring background blocks, we model the average intensities of the background blocks as a Gaussian Markov random field (GMRF) [24], [25]. We use an eight-point neighborhood system and assume only pairwise interactions between neighboring background blocks specified by the set of cliques $K_{bb} = \{\{r,s\} : r \text{ and } s \text{ are neighbor background blocks}\}$. Let X^b be the vector of all pixels from the background blocks of the luminance component. The Gibbs distribution of the GMRF is then given by

$$p(x^b) = \frac{1}{\text{const}} \exp\left\{-\frac{1}{2\sigma_B^2} \sum_{\{r,s\}\in K_{bb}} h_{r,s}(\mu_r - \mu_s)^2\right\} \tag{6}$$

where σ_B^2 and $h_{r,s}$ are the parameters of the distribution, and $\mu_s = \frac{1}{64}\sum_{i=0}^{63} x_{s,i}$ is the average intensity of the block s. The parameters $h_{r,s}$ are chosen as $h_{r,s} = 1/6$ if r and s are horizontal or vertical neighbors, and $h_{r,s} = 1/12$ if r and s are diagonal neighbors.

B. Prior Model for the Luminance Text Blocks

We choose the prior model for the text blocks of the luminance component to reflect the observation that text blocks are typically two-color blocks, i.e., most pixel values in such a block are concentrated around the foreground intensity and the background intensity. For each text block s, we model its two predominant intensities as independent random variables $C_{1,s}$ and $C_{2,s}$. To accommodate smooth transitions between the two intensities and other variations, we model each pixel within block s as a convex combination of $C_{1,s}$ and $C_{2,s}$ plus additive white Gaussian noise denoted by $W_{s,i}$. With this model, the ith pixel in block s is given by

$$X_{s,i} = \alpha_{s,i}C_{1,s} + (1-\alpha_{s,i})C_{2,s} + W_{s,i} \tag{7}$$

where the two gray levels, $C_{1,s}$ and $C_{2,s}$, are mixed together by $\alpha_{s,i}$ which plays a role similar to the alpha channel [26] in computer graphics. The random variables $W_{s,i}$ are mutually independent, zero-mean Gaussian random variables with a common variance σ_W^2.

Let α_s be the vector containing the alpha values of the pixels in the text block s, and let α be the vector concatenating α_s for all the text blocks. Further, let C_1 and C_2 be the vectors of all $C_{1,s}$ and $C_{2,s}$ random variables for all text blocks, respectively. We assume that the following three objects are mutually independent: the additive Gaussian noise, α, and the pair $\{C_1, C_2\}$.

[1] Here, and in the rest of the paper, we simplify notation by denoting all probability mass and density functions by p, whenever the random variables that they describe can be inferred from their arguments. Whenever an ambiguity may arise, we denote the probability mass or density function of the random variable V by p_V.

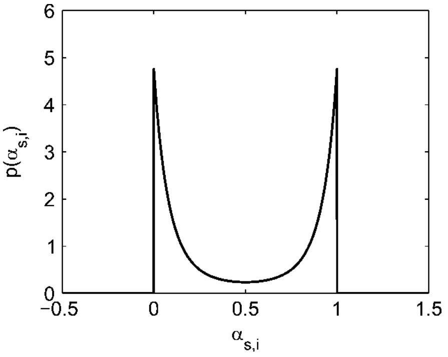


Fig. 2. Marginal probability density function of an alpha value $\alpha_{s,i}$, for $\nu = 12$. As the alpha value controls the proportion of the two intensities $C_{1,s}$ and $C_{2,s}$ present in a text pixel value, the density function's support is $[0,\ 1]$. The bimodal nature of the density function with peaks at 0 and 1 models the clustering of the text pixel values around $C_{1,s}$ and $C_{2,s}$.

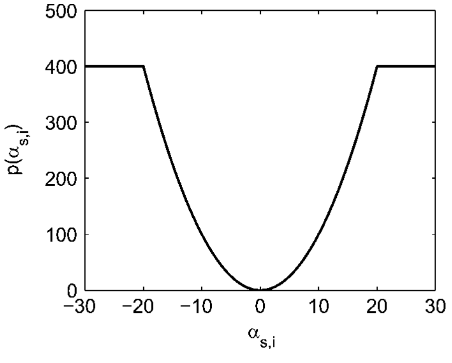


Fig. 3. Potential function $\rho(x) = \min(x^2, \tau^2)$, $\tau = 20$, of the Markov random fields used to characterize the spatial variation of the predominant colors $C_{1,s}$ and $C_{2,s}$. The threshold parameter τ ensures that we avoid excessively penalizing large intensity difference between the two corresponding predominant colors of two neighboring blocks.

It then follows from (7) that the conditional probability density function of the vector X^t of all the pixel values of the text blocks, given C_1, C_2 and α, is given by the Gaussian density

$$p(x^t|c_1, c_2, \alpha) = \frac{1}{\text{const}} \times \exp\left\{-\frac{1}{2\sigma_W^2} \sum_{s \text{ text block}} \|x_s - \alpha_s c_{1,s} - (\mathbf{1} - \alpha_s) c_{2,s}\|^2\right\} \quad (8)$$

where $\mathbf{1}$ is a 64-dimensional column vector with all entries equal to 1.

Since $\alpha_{s,i}$ models the proportion of the two intensities $C_{1,s}$ and $C_{2,s}$ present in $X_{s,i}$, we impose that $0 \leq \alpha_{s,i} \leq 1$ with probability one. The fact that most pixel values in a text block tend to cluster around the two predominant intensities is captured by modeling $\alpha_{s,i}$ with a bimodal distribution having peaks at 0 and 1. We model the components of α as independent and identically distributed random variables, with the joint probability density function (9), shown at the bottom of the page. As shown in Fig. 2, the marginal density for each $\alpha_{s,i}$ has support on $[0,\ 1]$ and peaks at 0 and 1. The parameter $\nu > 0$ controls the sharpness of the peaks, and, therefore, affects the smoothness of the foreground/background transition in the decoded text.

To enforce smoothness of colors in nearby blocks, we model spatial variation of the two predominant intensities of text blocks as two Markov random fields (MRF's) [24], [25]. We use an eight-point neighborhood system and assume only pairwise interactions between neighboring blocks for the MRF's. In addition, in the case of a text block, s, neighboring to a background block, r, one of the two predominant intensities of the text block is typically similar to the predominant intensity of the background block. Therefore, the MRF's also capture the pairwise interaction of every such pair $\{s, r\}$. For a background block r, we estimate its predominant intensity by $\hat{\mu}_r$ obtained from the background block decoding algorithm described in Section IV-A. Then, our model for C_1 and C_2 is expressed by the Gibbs distribution

$$p(c_1, c_2) = \frac{1}{\text{const}} \times \exp\left\{-\frac{1}{2\sigma_C^2} \sum_{\{s,r\} \in K_{tt}} \Big(\rho(c_{1,s} - c_{1,r}) + \rho(c_{2,s} - c_{2,r})\Big)\right\} \times \exp\left\{-\frac{1}{2\sigma_C^2} \sum_{\{s,r\} \in K_{tb}} \rho\big(\min(|c_{1,s} - \hat{\mu}_r|, |c_{2,s} - \hat{\mu}_r|)\big)\right\} \quad (10)$$

where $K_{tt} = \big\{\{s, r\} : s \text{ and } r \text{ are neighboring text blocks}\big\}$, $K_{tb} = \big\{\{s,r\} : s \text{ is a text block, } r \text{ is a background block, } s \text{ and } r \text{ are neighbors}\big\}$, and $\rho(x) = \min(x^2,\ \tau^2)$, where τ is a threshold parameter, as depicted in Fig. 3. The first exponential function of (10) describes the pairwise interactions between every pair $\{s,\ r\}$ of neighboring text blocks in the clique set K_{tt}. For each such pair, the potential function ρ encourages the similarity of $c_{1,r}$, and $c_{1,s}$ and the similarity of $c_{2,r}$ and $c_{2,s}$. The second exponential function of (10) captures the pairwise interactions of every pair $\{s, r\}$ of neighboring blocks such that s is a text block and r is a background block. For each such pair, the value of $c_{1,s}$ or $c_{2,s}$ which is closer to $\hat{\mu}_r$ is driven toward $\hat{\mu}_r$ by the potential function ρ. In the potential function ρ, the threshold τ is used to avoid excessively penalizing large intensity differences which may arise when two neighboring blocks are from two different text regions with distinct background and/or foreground intensities.

From (8), (9), and (10), the prior model for text blocks of the luminance component is given by (11), shown at the bottom of the next page.

$$p(\alpha) = \begin{cases} \frac{1}{\text{const}} \exp\left\{\nu \sum_{s \text{ text block}} \left\|\alpha_s - \frac{1}{2}\mathbf{1}\right\|^2\right\}, & \text{if } 0 \leq \alpha_{s,i} \leq 1 \quad \text{for all } s, i \\ 0, & \text{otherwise} \end{cases} \quad (9)$$

IV. Optimization for Decoding the Luminance Component

To decode the luminance component, we need to solve the optimization problems (4) and (5) with the specific prior models (6) for the background blocks and (11) for the text blocks. We use iterative optimization algorithms to solve the two problems. For each problem, we minimize the cost function iteratively through a series of simple local updates. Each update minimizes the cost function with respect to one or a few variables, while the remaining variables remain unchanged. One full iteration of the algorithm consists of updating every variable of the cost function once. These iterations are repeated until the change in the cost between two successive iterations is smaller than a predetermined threshold.

A. Optimization for Decoding the Luminance Background Blocks

To decode the luminance background blocks, we minimize $-\log p(x^b)$ of (6) subject to the constraints $T(Dx_s) = \tilde{y}_s$ for every background block s. We solve this minimization problem in the frequency domain. For the vector y_s containing the DCT coefficients of the block s, we adopt the convention that the first element $y_{s,0}$ is the DC coefficient of the block. Then, we can express the average intensity of the block s as $\mu_s = y_{s,0}/8$, and the original cost function, $-\log p(x^b)$, becomes

$$\mathcal{C}(y^b) = \frac{1}{128\sigma_B^2} \sum_{\{r,s\}\in K_{bb}} h_{r,s}(y_{r,0} - y_{s,0})^2 \qquad (12)$$

where y^b is the vector containing the DCT coefficients y_s of all the background blocks. We minimize the cost function (12) subject to the transformed constraints $T(y_s) = \tilde{y}_s$ for every background block s.

To perform the minimization, we first initialize y_s by the quantized DCT coefficients $\tilde{y}_s$ for each background block s. The algorithm then iteratively minimizes the cost function $\mathcal{C}(y^b)$ with respect to one variable at a time. We first obtain the unconstrained minimizer for $y_{s,0}$ by setting the partial derivative of the cost function with respect to $y_{s,0}$ to zero. Then, we clip the unconstrained minimizer to the quantization range which $y_{s,0}$ must fall in, and update $y_{s,0}$ by (13), shown at the bottom of the page, where $\mathrm{clip}(\cdot, [\min, \max])$ is the clipping operator which clips the first argument to the range $[\min, \max]$. Because the cost function is independent of the AC coefficients, the AC coefficients remain unchanged.

B. Optimization for Decoding the Luminance Text Blocks

In order to decode the luminance text blocks, we must minimize the cost function of (11) subject to the constraint that $T(Dx_s) = \tilde{y}_s$ for every text block s. We perform this task using iterative optimization, where each full iteration consists of a single update of each block, s. The update of each block s is performed in three steps: 1) First, we minimize the cost with respect to the alpha channel, α_s; 2) we then minimize with respect to the two colors, $(c_{1,s}, c_{2,s})$; 3) and finally we minimize with respect to the pixel values, x_s. These full iterations are repeated until the desired level of convergence is reached. We now describe the procedures used for each of these three required updates for a particular block s.

The block update of α_s is computed by successively minimizing the cost with respect to $\alpha_{s,i}$ at each pixel location i. For a particular $\alpha_{s,i}$, we can rewrite the cost function as a quadratic function of $\alpha_{s,i}$ in the form $a\alpha_{s,i}^2 + b\alpha_{s,i} + d$, where

$$a = \frac{(c_{2,s} - c_{1,s})^2}{2\sigma_W^2} - \nu, \qquad (14)$$

$$b = \frac{(c_{2,s} - c_{1,s})(x_{s,i} - c_{2,s})}{\sigma_W^2} + \nu. \qquad (15)$$

$$\begin{aligned} -\log p(x^t, c_1, c_2, \alpha) = &\frac{1}{2\sigma_W^2} \sum_{s \text{ text block}} \|x_s - \alpha_s c_{1,s} - (\mathbf{1} - \alpha_s)c_{2,s}\|^2 \\ &+ \frac{1}{2\sigma_C^2} \sum_{\{s,r\}\in K_{tt}} \big(\rho(c_{1,s} - c_{1,r}) + \rho(c_{2,s} - c_{2,r})\big) \\ &+ \frac{1}{2\sigma_C^2} \sum_{\{s,r\}\in K_{tb}} \rho(\min(|c_{1,s} - \hat{\mu}_r|, |c_{2,s} - \hat{\mu}_r|)) \\ &- \nu \sum_{s \text{ text block}} \left\|\alpha_s - \frac{1}{2}\mathbf{1}\right\|^2 + \text{const} \end{aligned} \qquad (11)$$

$$y_{s,0} \leftarrow \mathrm{clip}\left(\frac{\sum_{r:\{r,s\}\in K_{bb}} h_{r,s} y_{r,0}}{\sum_{r:\{r,s\}\in K_{bb}} h_{r,s}}, \quad \left[\tilde{y}_{s,0} - \frac{Q_0}{2}, \tilde{y}_{s,0} + \frac{Q_0}{2}\right]\right) \qquad (13)$$

If $a \neq 0$, this quadratic function has the unique unconstrained extremum at

$$\alpha^*_{s,i} = -\frac{b}{2a} = \frac{\nu\sigma^2_W + (c_{2,s} - c_{1,s})(x_{s,i} - c_{2,s})}{2\nu\sigma^2_W - (c_{2,s} - c_{1,s})^2}. \tag{16}$$

If $a > 0$, the quadratic function is convex and the constrained minimizer for $\alpha_{s,i}$ is $\alpha^*_{s,i}$ clipped to the interval $[0, 1]$. If $a < 0$, the quadratic function is concave and the constrained minimizer for $\alpha_{s,i}$ is either 0 or 1, depending on whether $\alpha^*_{s,i} > 1/2$ or $\alpha^*_{s,i} \leq 1/2$. In the case when $a = 0$, the quadratic function reduces to a linear function of $\alpha_{s,i}$ with slope b, and the constrained minimizer for $\alpha_{s,i}$ is either 0 or 1, depending on the sign of b. Thus, the update formula for this particular $\alpha_{s,i}$ is

$$\alpha_{s,i} \leftarrow \begin{cases} \text{clip}(\alpha^*_{s,i}, [0,1]), & \text{if } a > 0 \\ \text{step}(\frac{1}{2} - \alpha^*_{s,i}), & \text{if } a < 0 \\ \text{step}(-b), & \text{if } a = 0 \end{cases} \tag{17}$$

where $\text{step}(\cdot)$ is the unit step function.

The block update of the two colors, $(c_{1,s}, c_{2,s})$ requires the minimization of the cost function

$$F(c_{1,s}, c_{2,s}) = \frac{1}{2\sigma^2_W} \|x_s - \alpha_s c_{1,s} - (\mathbf{1} - \alpha_s) c_{2,s}\|^2 + \frac{1}{2\sigma^2_C} \sum_{r \in \partial s} f_r(c_{1,s}, c_{2,s}) \tag{18}$$

where ∂s is the set of the nonpicture neighbor blocks of s, and $f_r(c_{1,s}, c_{2,s})$ is given by (19), shown at the bottom of the page.

Unfortunately, $f_r(c_{1,s}, c_{2,s})$ is a nonconvex function of $(c_{1,s}, c_{2,s})$; however, the optimization problem can be simplified by using functional substitution methods to compute an approximate solution to the original problem [27], [28]. Using functional substitution, we replace the $f_r(c_{1,s}, c_{2,s})$ by

$$\tilde{f}_r(c_{1,s}, c_{2,s}) = a_{1,r}|c_{1,s} - b_{1,r}|^2 + a_{2,r}|c_{2,s} - b_{2,r}|^2 \tag{20}$$

where $b_{1,r} = c_{1,r}$ and $b_{2,r} = c_{2,r}$ if r is a text block, and $b_{1,r} = b_{2,r} = \hat{\mu}_r$ if r is a background block. The coefficients $a_{1,r}$ and $a_{2,r}$ are chosen as (21) and (22), shown at the bottom of the page, where the primed quantities, $c'_{1,s}$ and $c'_{2,s}$, denote the values of the colors before updating. Each step function of the form $\text{step}(A - B)$ simply captures the inequality test $A > B$.

Using this substitute function results in the quadratic cost function given by

$$\tilde{F}(c_{1,s}, c_{2,s}) = \frac{1}{2\sigma^2_W} \|x_s - \alpha_s c_{1,s} - (\mathbf{1} - \alpha_s) c_{2,s}\|^2 + \frac{1}{2\sigma^2_C} \sum_{r \in \partial s} \tilde{f}_r(c_{1,s}, c_{2,s}). \tag{23}$$

Since this cost is quadratic, the update can be computed in closed form as the solution to

$$(c_{1,s}, c_{2,s}) \leftarrow \arg \min_{c_{1,s}, c_{2,s}} \tilde{F}(c_{1,s}, c_{2,s}). \tag{24}$$

The block update of the pixels x_s requires that the cost function $\|x_s - \alpha_s c_{1,s} - (\mathbf{1} - \alpha_s) c_{2,s}\|^2$ be minimized subject to the constraint that $T(Dx_s) = \tilde{y}_s$. The solution to this constrained minimization problem can be computed using the three steps given by (25)–(27) at the bottom of the page. The quantity $\alpha_s c_{1,s} + (\mathbf{1} - \alpha_s) c_{s,2}$ is first transformed to the DCT domain in (25). Then (26) clips these DCT coefficients to the respective ranges they are known to be within. Finally in (27), these clipped DCT coefficients are transformed back to the space domain to form the updated pixels, x_s. Because the DCT is orthogonal, these three steps compute the correct constrained minimizer for x_s. Since we need to estimate $c_{1,s}$ and $c_{2,s}$ in the spatial domain and enforce the forward model constraint in the DCT domain, each block update must include a forward DCT and a backward DCT.

$$f_r(c_{1,s}, c_{2,s}) = \begin{cases} \rho(c_{1,s} - c_{1,r}) + \rho(c_{2,s} - c_{2,r}), & \text{if } r \text{ is a text block} \\ \rho(\min(|c_{1,s} - \hat{\mu}_r|, |c_{2,s} - \hat{\mu}_r|)), & \text{if } r \text{ is a background block} \end{cases} \tag{19}$$

$$a_{1,r} = \begin{cases} \text{step}(\tau - |c'_{1,s} - c_{1,r}|), & \text{if } r \text{ is a text block} \\ \text{step}(\tau - |c'_{1,s} - \hat{\mu}_r|)\text{step}(|c'_{2,s} - \hat{\mu}_r| - |c'_{1,s} - \hat{\mu}_r|), & \text{if } r \text{ is a background block} \end{cases} \tag{21}$$

$$a_{2,r} = \begin{cases} \text{step}(\tau - |c'_{2,s} - c_{2,r}|), & \text{if } r \text{ is a text block} \\ \text{step}(\tau - |c'_{2,s} - \hat{\mu}_r|)\text{step}(|c'_{1,s} - \hat{\mu}_r| - |c'_{2,s} - \hat{\mu}_r|), & \text{if } r \ \text{ is a background block} \end{cases} \tag{22}$$

$$y_s \leftarrow D(\alpha_s c_{1,s} + (\mathbf{1} - \alpha_s) c_{2,s}) \tag{25}$$

$$y_{s,i} \leftarrow \text{clip}\left(y_{s,i}, \left[\tilde{y}_{s,i} - \frac{Q_i}{2}, \tilde{y}_{s,i} + \frac{Q_i}{2}\right]\right), \quad \text{for} \quad i = 0, \ldots, 63 \tag{26}$$

$$x_s \leftarrow D^{-1} y_s \tag{27}$$

Update iterations for text block decoding

```
do {
    for each text block $s$ {
        /* update alpha values $\alpha_s$ */
        for $i = 0, \ldots, 63$
            update $\alpha_{s,i}$ by (17)

        /* update $c_{1,s}$ and $c_{2,s}$ */
        for each $r \in \partial s$
            determine $\tilde{f}_r(c_{1,s}, c_{2,s})$ by (20)-(22)
        $(c_{1,s}, c_{2,s}) \leftarrow \arg\min_{c_{1,s}, c_{2,s}} \tilde{F}(c_{1,s}, c_{2,s})$

        /* update pixels $x_s$ */
        $y_s \leftarrow D(\alpha_s c_{1,s} + (\mathbf{1} - \alpha_s) c_{2,s})$
        for $i = 0, \ldots, 63$
            $y_{s,i} \leftarrow \mathrm{clip}(y_{s,i}, [\tilde{y}_{s,i} - \frac{Q_i}{2}, \tilde{y}_{s,i} + \frac{Q_i}{2}])$
        $x_s \leftarrow D^{-1} y_s$
    }
} while change in cost function > threshold
```

Fig. 4. Pseudo-code of the update iterations for text block decoding. One full iteration consists of updating every text block once. Each text block s is updated in three steps which minimize the cost with respect to: 1) the alpha values in α_s; 2) the predominant intensities $(c_{1,s}, c_{2,s})$; and 3) the pixel intensities in x_s.

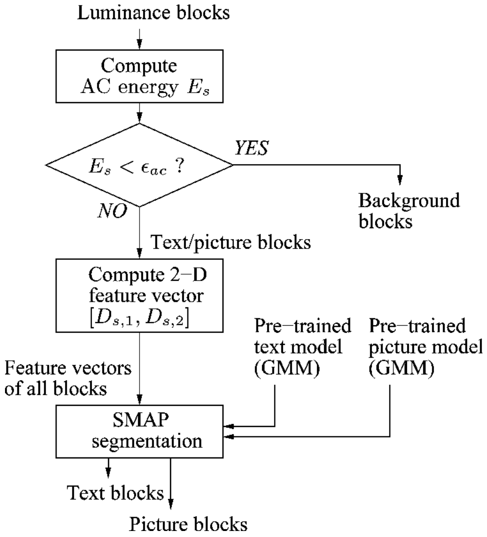


Fig. 5. Block-based segmentation. The background blocks are first identified by AC energy thresholding. A 2-D feature vector is then computed for each block. Two Gaussian mixture models are obtained from supervised training: one for the text class and one for the picture class. With these two models, the feature vector image is segmented using the SMAP segmentation algorithm. The result is combined with the detected background blocks to form the final segmentation map.

Fig. 4 gives the pseudo-code for the update iterations of the text blocks. Since all the update formulas reduce the cost function monotonically, convergence of the algorithm is ensured.

Lastly, we briefly describe the initialization of the algorithm. For each text block s, we initialize the intensity values x_s by the values $\tilde{x}_s$ decoded by conventional JPEG. For $c_{1,s}$ and $c_{2,s}$, we first identify the pixels decoded by conventional JPEG and located within the 16×16 window centered at the block s, and we cluster the pixels into two groups using k-means clustering [29]. We then initialize $c_{1,s}$ by the smaller of the two cluster means, and initialize $c_{2,s}$ by the larger mean. The alpha values require no initialization.

V. Block-Based Segmentation

Our segmentation algorithm classifies each luminance block as one of three classes: background, text, and picture. Fig. 5 shows the block diagram of the segmentation algorithm.

We first compute the AC energy of each block s by $E_s = \sum_{i=1}^{63} \tilde{y}_{s,i}^2$, where $\tilde{y}_{s,i}$ is the ith quantized DCT coefficient of the block. If E_s is smaller than the threshold ϵ_{ac}, the block s is classified as a background block.

Next, we compute a 2-D feature vector for each block in order to classify the remaining blocks into the text and picture classes. The first feature component is based on the encoding length proposed in [8], [30]. The encoding length of a block is defined as the number of bits in the JPEG stream used to encode the block. Typically, the encoding lengths for text blocks are longer than for nontext blocks due to the presence of high contrast edges in the text blocks. However, the encoding length also depends on the quantization matrix: the larger the quantization steps, the smaller the encoding length. To make the feature component more robust to different quantization matrices, we multiply the encoding length by a factor determined from the quantization matrix. Suppose Q_i^* are the default luminance quantization step sizes as defined in Table K.1 in [2], and Q_i are the quantization step sizes used to encode the luminance component. We use the quantity $\lambda = \sum_i Q_i^* Q_i / \sum_i Q_i^* Q_i^*$ as a measure of the coarseness of the quantization step sizes Q_i as compared to the default. Larger quantization step sizes Q_i correspond to larger values of λ. We define the first feature component of the block s by

$$D_{s,1} = \lambda^{\gamma} \times \text{encoding length of block } s \tag{28}$$

where the parameter $\gamma = 0.5$ is determined from training. The second feature component, $D_{s,2}$, measures how close a block is to being a two-color block: the smaller $D_{s,2}$, the closer the block s is to being a two-color block. We take the luminance component decoded by the convectional JPEG decoder and use k-means clustering to separate the pixels in a 16×16 window centered at the block s into two groups. Let $\theta_{1,s}$ and $\theta_{2,s}$ denote the two cluster means. If $\theta_{1,s} \neq \theta_{2,s}$, the second feature component is computed by

$$D_{s,2} = \frac{\sum_{i=0}^{63} \min\{|\tilde{x}_{s,i} - \theta_{1,s}|^2, |\tilde{x}_{s,i} - \theta_{2,s}|^2\}}{|\theta_{1,s} - \theta_{2,s}|^2}. \tag{29}$$

If $\theta_{1,s} = \theta_{2,s}$, we define $D_{s,2} = 0$.

We characterize the feature vectors of the text blocks and those of the picture blocks by two Gaussian mixture models. We use these two Gaussian mixture models with the SMAP segmentation algorithm [31] to segment the feature vector image. The result is combined with the background blocks detected by AC thresholding to produce the final segmentation map.

Last, we describe the training process which determines the parameter γ in (28) and the two Gaussian mixture models of the text and picture classes. In the training process, we use a set of training images consisting of 54 digital and scanned images. Each image is manually segmented and JPEG encoded

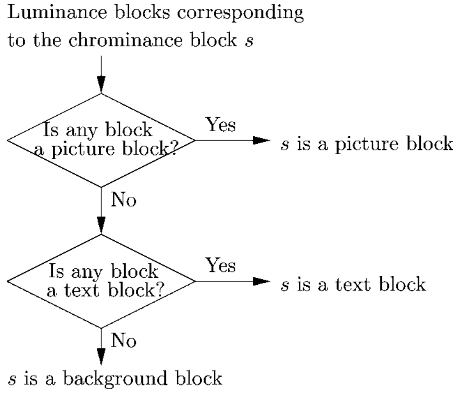

Luminance blocks corresponding
to the chrominance block s
Is any block a picture block?
Yes
s is a picture block
No
Is any block a text block?
Yes
s is a text block
No
s is a background block

Fig. 6. Classification rule for a chrominance block in a subsampled chrominance component. Each chrominance block s corresponds to several luminance blocks which cover the same area of the image. If these luminance blocks contain a picture block, block s is labeled as a picture block. Otherwise, if the luminance blocks contain a text block, block s is labeled as a text block. If all the corresponding luminance blocks are background blocks, block s is labeled as a background block.

with 9 different quantization matrices, corresponding to λ_j with $j = 1, \ldots, 9$. For the ith image encoded by the jth quantization matrix, we first compute the average encoding lengths of the text blocks and the picture blocks, denoted by $u_{i,j}$ and $v_{i,j}$ respectively. The parameter γ is then determined from the following optimization problem:

$$\hat{\gamma} = \arg\min_{\gamma} \min_{u,v} \sum_{i=1}^{54} \sum_{j=1}^{9} \left[(\lambda_j^{\gamma} u_{i,j} - u)^2 + (\lambda_j^{\gamma} v_{i,j} - v)^2 \right]. \quad (30)$$

Next, we obtain the Gaussian mixture model for the text class by applying the EM algorithm to the feature vectors of the text blocks of the JPEG encoded images, using the implementation in [32]. To reduce computation, only 2% of the text blocks from each JPEG encoded image are used to perform training. By the same procedure, we obtain the Gaussian mixture model for the picture class using the feature vectors of the picture blocks.

VI. Decoding of the Chrominance Components

In this section, we explain how to extend the luminance decoding scheme to the chrominance components. To decode a particular chrominance component, we first segment the chrominance blocks into the background, text, and picture classes based on the classification of the luminance blocks. If the chrominance and luminance components have the same resolution, we label each chrominance block by the class of the corresponding luminance block. However, if the chrominance component has been subsampled, then each chrominance block corresponds to several luminance blocks. In this case, we determine the class of each chrominance block based on the classification of the corresponding luminance blocks according to the procedure in Fig. 6.

The background and picture blocks of the chrominance component are decoded using the same methods as are used for their luminance counterparts. However, chrominance text blocks are decoded using the alpha channel calculated from the corresponding luminance blocks. If the chrominance component and the luminance component have the same resolution, the luminance alpha channel is used as the chrominance alpha channel. However, if the chrominance component has been subsampled, then the chrominance alpha channel is obtained by decimating the luminance alpha channel using block averaging. The only problem when the chrominance component has been subsampled is that the corresponding luminance blocks may include background blocks. For these luminance background blocks, we must determine the alpha channel in order to perform the decimation. For such a luminance background block r, we can create the missing alpha channel by comparing its average intensity $\hat{\mu}_r$ to the average values of the two predominant intensities of its neighboring text blocks. If $\hat{\mu}_r$ is closer to the average value of $\hat{c}_{1,s}$, the alpha values of the pixels in the block r are set to 1. Otherwise, the alpha values of the background pixels are set to 0.

TABLE I
Parameter Values Selected for the Proposed Algorithm

Parameter	Value	Defined in
$h_{r,s}$	$1/6$ if r, s are immediate neighbor blocks, $1/12$ if r, s are diagonal neighbor blocks	Eq. (6)
σ_W	5	Eq. (8)
ν	12	Eq. (9)
σ_C	3.5	Eq. (10)
ϵ_{ac}	200	Section V
τ	20	ρ in Section III-B

The optimization for decoding the chrominance text blocks is similar to the algorithm described in Section IV-B except for the following changes. First, we initialize the two predominant intensities $c_{1,s}$ and $c_{2,s}$ for each chrominance text block s using their MMSE estimates

$$(c_{1,s}, c_{2,s}) = \arg\min_{c_{1,s}, c_{2,s}} \left\| \tilde{x}_s - \hat{\alpha}_s c_{1,s} - (\mathbf{1} - \hat{\alpha}_s) c_{2,s} \right\|^2 \quad (31)$$

where $\tilde{x}_s$ contains the pixel values of the block decoded by the conventional JPEG decoder, and $\hat{\alpha}_s$ is the alpha channel of the block computed from the luminance alpha channel. Second, since the value of the alpha channel is computed from the luminance component, the step of updating the alpha channel is skipped in the algorithm of Fig. 4.

Lastly, for a subsampled chrominance component, we need to interpolate the component to restore its original resolution. We apply linear interpolation to the background blocks and the picture blocks. For the text blocks, we perform the interpolation by combining the decoded chrominance component with the high resolution luminance alpha channel. We explain this interpolation scheme in Fig. 7 for the case when the chrominance component has been subsampled by 2 in both vertical and horizontal directions. For each of the interpolated chrominance pixels, we use the corresponding luminance alpha value as its alpha value, and offset the decoded pixel value x_k by the difference in alpha values $\alpha_k - \alpha_{k,i}$ scaled by the range $c_2 - c_1$. The scheme can easily be generalized to other subsampling factors. Using this interpolation scheme, the resulting text regions are sharper than they are when using linear interpolation.

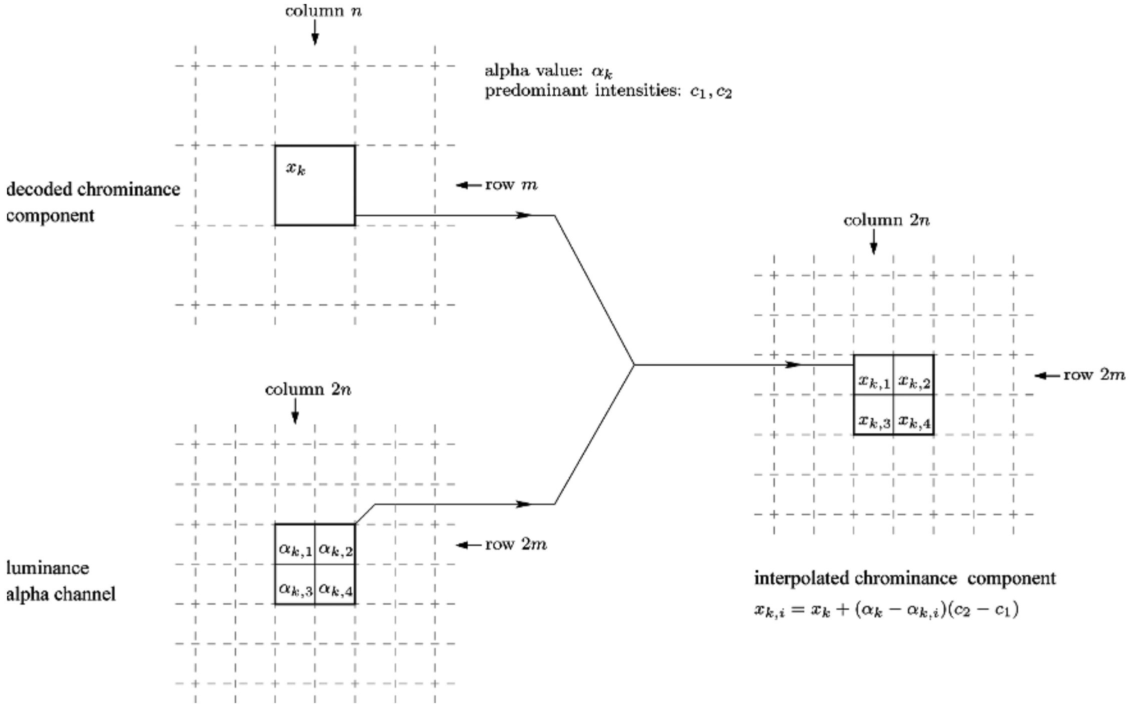

column n
alpha value: α_k
predominant intensities: c_1, c_2
decoded chrominance component
x_k
row m
column $2n$
$x_{k,1}$ $x_{k,2}$
$x_{k,3}$ $x_{k,4}$
row $2m$
column $2n$
luminance alpha channel
$\alpha_{k,1}$ $\alpha_{k,2}$
$\alpha_{k,3}$ $\alpha_{k,4}$
row $2m$
interpolated chrominance component
$x_{k,i} = x_k + (\alpha_k - \alpha_{k,i})(c_2 - c_1)$

Fig. 7. Interpolation of chrominance text pixels when the chrominance component has been subsampled by 2 in both vertical and horizontal directions. For the text pixel at position (m, n) of the decoded chrominance component, suppose its decoded value is x_k, its alpha value is α_k, and the two predominant intensities are c_1 and c_2. We first identify the corresponding luminance pixels at positions $(2m, 2n)$, $(2m, 2n+1)$, $(2m+1, 2n)$, and $(2m+1, 2n+1)$. Using the alpha values of these luminance pixels, we then compute the corresponding pixels of the interpolated chrominance component by $x_{k,i} = x_k + (\alpha_k - \alpha_{k,i})(c_2 - c_1)$, where $\alpha_{k,i}$ is the estimated luminance alpha value.

Biomedical Engineering Highlights
Building the Biomed Knowledge Base
Assembly
Helvetica Text 27
E=MPAC²T
Wise Words
QUIZ
How Much Ash Could a Nuke Plant Save if a Nuke Plant Replaced Coal?
impact interact

(a) Image 1
2550×3300 pixels, 300dpi

(b) Image 2
3193×4174 pixels, 400dpi

(c) Image 3
2336×3215 pixels, 300dpi

Fig. 8. Thumbnails of the original test images. The corresponding JPEG encoded images have bit rates 0.43 bits per pixel (bpp), 0.53 bpp, and 0.32 bpp, respectively. All the three images were compressed with 2:1 chrominance subsampling in both vertical and horizontal directions.

VII. Experimental Results

We now present the results of several image decoding experiments. We demonstrate that our proposed algorithm significantly outperforms the conventional JPEG decoding algorithm and three other existing JPEG decoding schemes. Table I summarizes the parameter values chosen for the proposed algorithm. In decoding the background blocks, the parameter σ_B^2 in the cost function (12) is a positive multiplicative constant whose value is irrelevant in determining the minimizer. Therefore, it is omitted from Table I.

To evaluate the performance of the proposed algorithm, we use 60 test document images: 30 digital images converted from soft copies, and 30 scanned images obtained using an Epson Expression 10000XL scanner and descreened by [33]. Each of the

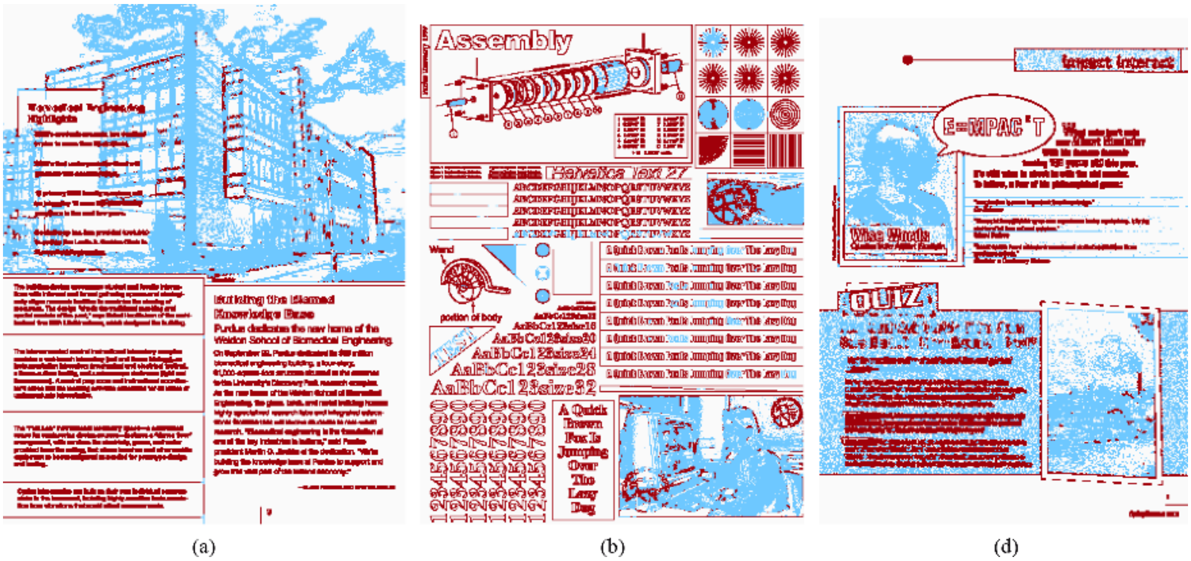
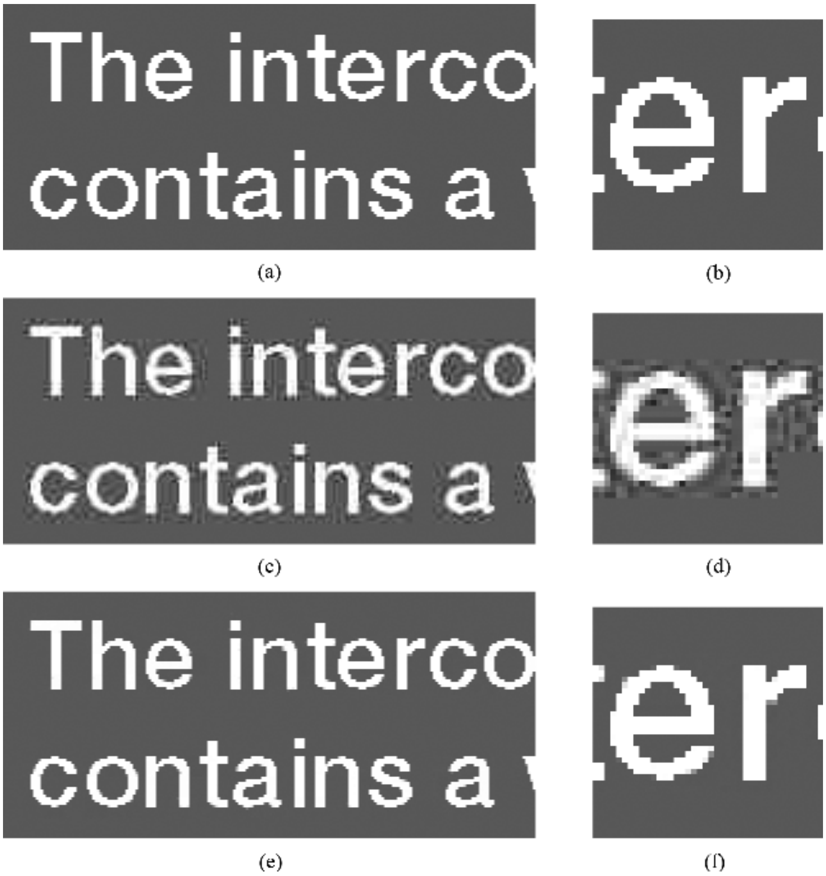

(a) (b) (d)

Fig. 9. Segmentation maps of (a) Image 1, (b) Image 2, and (c) Image 3. White: background blocks; red: text blocks; blue: picture blocks.

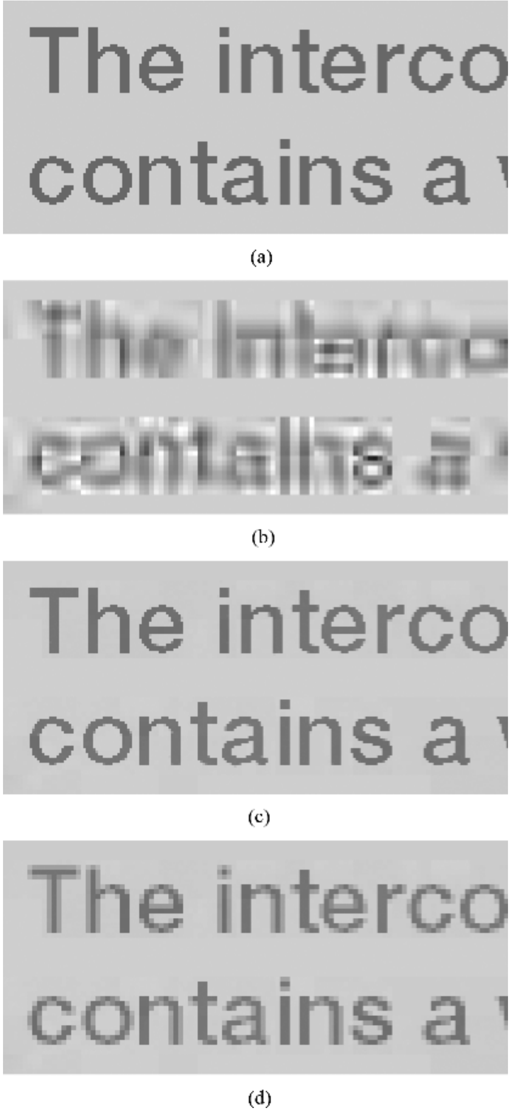


(a) (b)
(c) (d)
(e) (f)

Fig. 10. Luminance component of a text region of Image 1. (a), (b) Original. (c), (d) Conventional JPEG decoding. (e), (f) The proposed scheme. (b), (d), and (f) are enlargements of a small region of (a), (c), and (e) respectively.

60 images contains some text and/or graphics. Since our focus is document images, we do not consider images that are purely pictures. Six of the 30 digital images and 11 of the 30 scanned images are purely text/graphics with no pictures. None of the test images were used for training our segmentation algorithm. We discuss and demonstrate the visual quality of the decoded images using three example images shown in Fig. 8. Both Image 1 and Image 2 are digital images, and Image 3 is a scanned image. They are all JPEG encoded with 2:1 chrominance subsampling in both vertical and horizontal directions. We use high compression ratios to compress the images in order to show the improvement in the decoded images more clearly.

We apply our segmentation algorithm, described in Section V, to the JPEG encoded images. Fig. 9 shows that the



(a)

(b)

(c)

(d)

Fig. 11. Chrominance component (C_r) of the region shown in Fig. 10. (a) Original. (b) Decoded by conventional JPEG decoding and interpolated by pixel replication. (c) Decoded by our scheme. (d) Decoded by our scheme but interpolated by pixel replication.

corresponding segmentation results are generally accurate. It should be noted that in the smooth regions of natural images, many image blocks are classified as background blocks. This classification is appropriate since it then allows our decoding algorithm to reduce the blocking artifacts in these regions.

Figs. 10 and 11 demonstrate the improvement in text block decoding using the proposed algorithm. Fig. 10(a) shows the luminance component of a small text region computed from Image 1. A small region within Fig. 10(a) is further enlarged in Fig. 10(b) to show the fine details. Fig. 10(c) and (d) shows the region of the JPEG encoded image decoded by the conventional JPEG decoder. The decoded region contains obvious ringing artifacts around the text. Fig. 10(e) and (f) shows the same region decoded by our scheme. Compared to Fig. 10(c) and (d), the region decoded by our scheme is essentially free from ringing artifacts and has a much more uniform foreground and background. In addition, the foreground and background intensities are also faithfully recovered.

Fig. 11(a) shows the chrominance component C_r for the region in Fig. 10(a). The result decoded by the conventional JPEG decoder and interpolated by pixel replication is shown in Fig. 11(b). The decoded region is highly distorted due to chrominance subsampling. Fig. 11(c) shows the region decoded by the proposed scheme. Since the decoding is aided by the luminance alpha channel, the visual quality of the decoded region is much higher than that decoded by the conventional JPEG decoder. To demonstrate the effect of interpolation of the chrominance components, Fig. 11(d) shows the result decoded by our scheme but interpolated by pixel replication. The text region decoded by our scheme in Fig. 11(c) is much clearer and sharper as compared to Fig. 11(d).

Fig. 12(c) shows the region completely decoded using our scheme. A comparison with the same region decoded by the conventional JPEG decoder in Fig. 12(b) reveals that the proposed algorithm significantly improves the quality of the decoded regions. Additional results for text regions in Fig. 13(c)–15(c) shows that the proposed algorithm consistently decodes the text regions at high quality.

We also compare our results with three existing JPEG decoding algorithms: Algorithm I proposed in [11], Algorithm II proposed in [34], and Algorithm III proposed in [3]. Algorithm I is a MAP reconstruction scheme. Both Algorithm II and Algorithm III are segmentation based decoding schemes.

Algorithm I uses a Markov random field as the prior model for the whole image. The scheme employs the Huber function as the potential function of the MRF. Using gradient descent optimization, the scheme performs JPEG decoding by computing the MAP estimate of the original image given the encoded DCT coefficients. Figs. 12(d)–15(d) show the decoding results for the text regions. Algorithm I significantly reduces the ringing artifacts in the text regions. However, because the prior model was not designed specifically for text, the decoded regions are generally not as sharp as those decoded by our scheme. Also, because the color components are decoded independently, the chrominance components decoded by Algorithm I are of low quality.

Algorithm II uses the segmentation algorithm of [8] to classify each image block as background, text or picture. However, in principle, Algorithm II can be used in conjunction with any preprocessing segmentation procedure that labels each block as background, text, or picture. Since our main objective is to evaluate the decoding methods rather than the preprocessing methods, we use our segmentation maps with Algorithm II. Algorithm II uses stochastic models for the DCT coefficients of the text blocks and of the picture blocks, and replaces each DCT coefficient with its Bayes least-squares estimate. The algorithm estimates the model parameters from the encoded DCT coefficients. The conventional JPEG decoded background blocks are left unchanged by Algorithm II.

The text decoding results of Algorithm II, shown in Figs. 12(e)–15(e), are only marginally improved over the conventional JPEG decoding. During JPEG encoding, many of the high-frequency DCT coefficients are quantized to zero, which is a main cause of the ringing artifacts in the decoded text blocks. However, due to the symmetry of the Gaussian distributions assumed for the text blocks by Algorithm II, the zero DCT coefficients are not altered at all by Algorithm II.

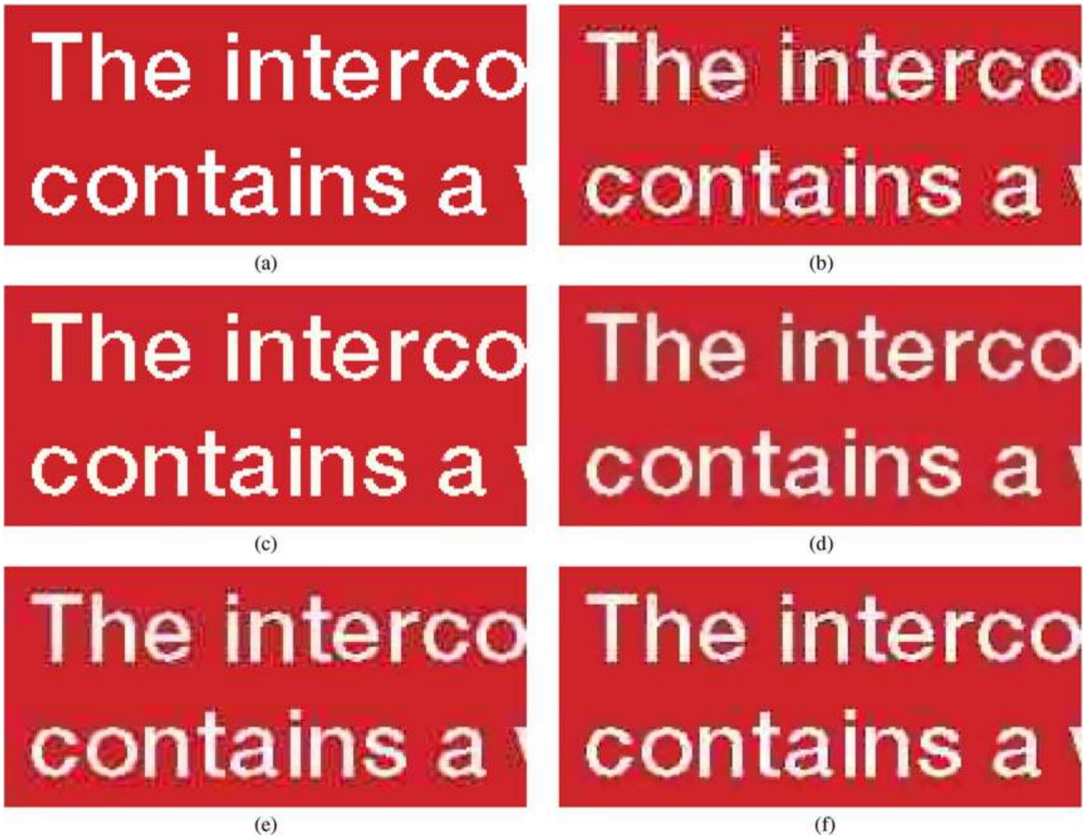


Fig. 12. Text region from Image 1. (a) Original. (b) Conventional JPEG decoding. (c) The proposed algorithm. (d) Algorithm I [11]. (e) Algorithm II [34]. (f) Algorithm III [3].

Therefore, the prior model imposed by Algorithm II is insufficient to effectively restore the characteristics of the text.

Algorithm III assumes that the image has been segmented into text blocks and picture blocks. It furthermore assumes that the text parts have been segmented into regions each of which has a uniform background and a uniform foreground. For each text region, Algorithm III first uses the intensity histogram to estimate the background color, and applies a simple thresholding scheme followed by morphological erosion to identify the background pixels. The scheme then replaces the intensity of each background pixel with the estimated background color. Finally, if any DCT coefficient falls outside the original quantization interval as a result of this processing, it is changed to the closest quantization cut-off value of its correct quantization interval. For the picture blocks, Algorithm III smooths out blocking artifacts by applying a sigma filter to the nonedge pixels on the boundaries of picture blocks, as identified by an edge detection algorithm.

There is a difficulty that prevents a direct comparison of our algorithm to Algorithm III. The difficulty stems from the assumption that the text portions of the image have been presegmented into regions with uniform background and uniform foreground. Without such a segmentation procedure, the scheme is not directly applicable to images in which text regions have varying background and/or foreground colors, such as our three test images. Therefore, in order to compare our algorithm to Algorithm III, we manually select from Image 1 a single text region which has a uniform foreground color and a uniform background color—specifically, the entire rectangular region with red background. We then process the entire Image 1 with Algorithm III: the blocks in the manually selected text region are processed as text blocks, and the rest of the image is processed as picture blocks. We show a portion of the selected text region in Fig. 12(a), and the result of decoding it with Algorithm III in Fig. 12(f). Since Algorithm III only smoothes out the background pixels, ringing artifacts are still strong in the foreground and near the background/foreground transition areas. In addition, due to the low resolution and low signal-to-noise ratio in the chrominance components, the computed chrominance background masks have low accuracy. This leads to color bleeding in the decoded text. In Fig. 15(f), similar results are obtained for Image 3 in which we select the region with red text on white background in the upper right portion of the document as the only text region to apply Algorithm III.

Fig. 16 compares the decoding results for a region containing mostly background blocks. In this region, most of the image blocks corresponding to the blue sky are classified as background, while most of the remaining blocks corresponding to the clouds are classified as picture blocks. Fig. 16(b) shows the region decoded by the conventional JPEG decoder. The decoded region exhibits obvious contouring as a result of quantization. Algorithm I, Fig. 16(d), significantly reduces the blocking artifacts, but contouring in the blue sky is still apparent. Algorithm II uses the conventional JPEG decoded blocks for the background blocks, so contouring in the blue sky is not improved at all. As Algorithm III applies the sigma filter only to the block boundary pixels, contouring is only slightly improved in Fig. 16(f). With our scheme, Fig. 16(c), contouring

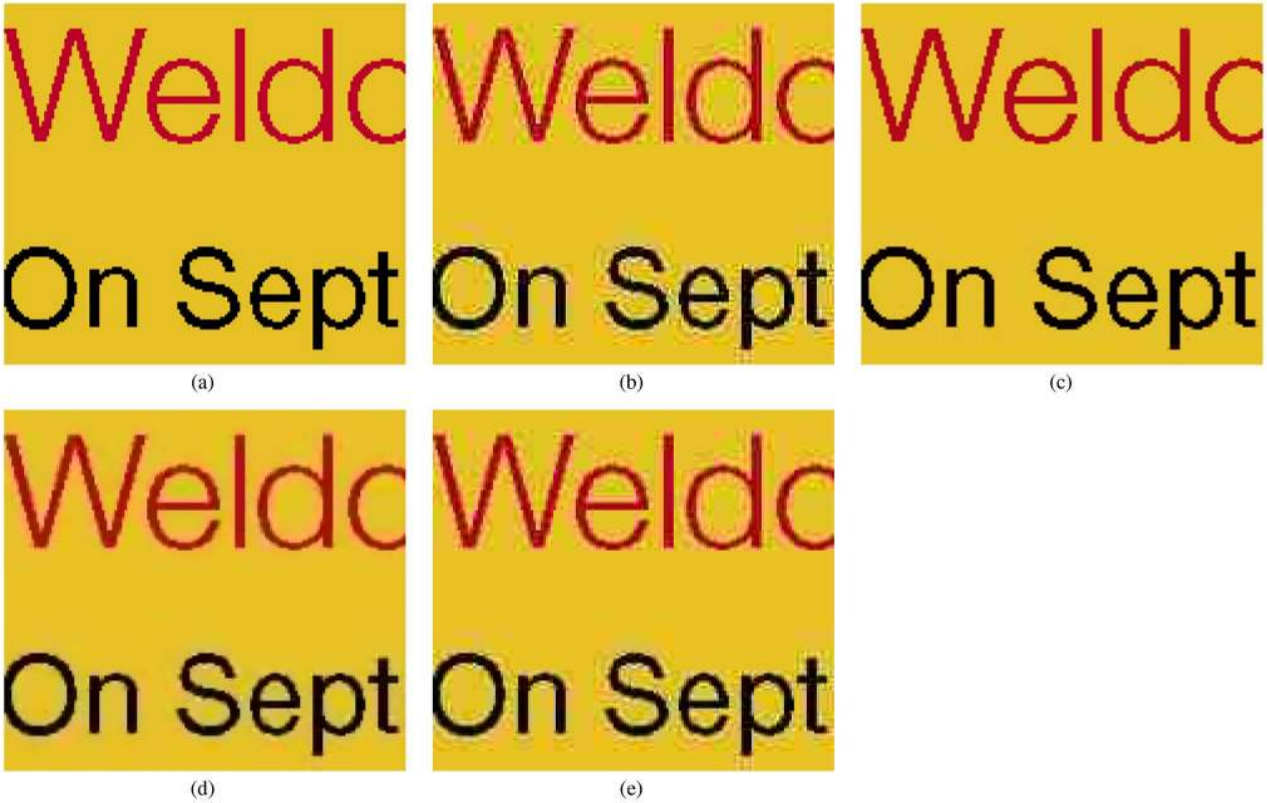


Fig. 13. Another text region from Image 1. (a) Original. (b) Conventional JPEG decoding. (c) The proposed algorithm. (d) Algorithm I [11]. (e) Algorithm II [34].

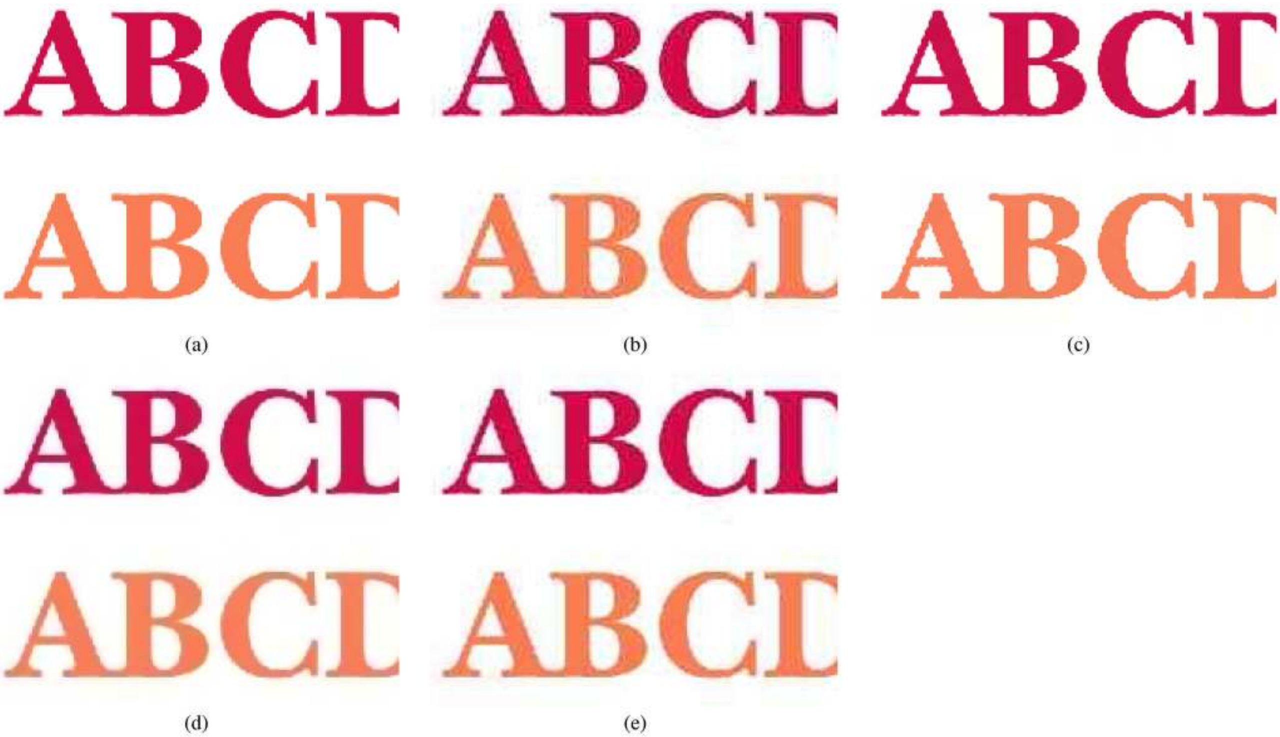


Fig. 14. Text region from Image 2. (a) Original. (b) Conventional JPEG decoding. (c) The proposed algorithm. (d) Algorithm I [11]. (e) Algorithm II [34].

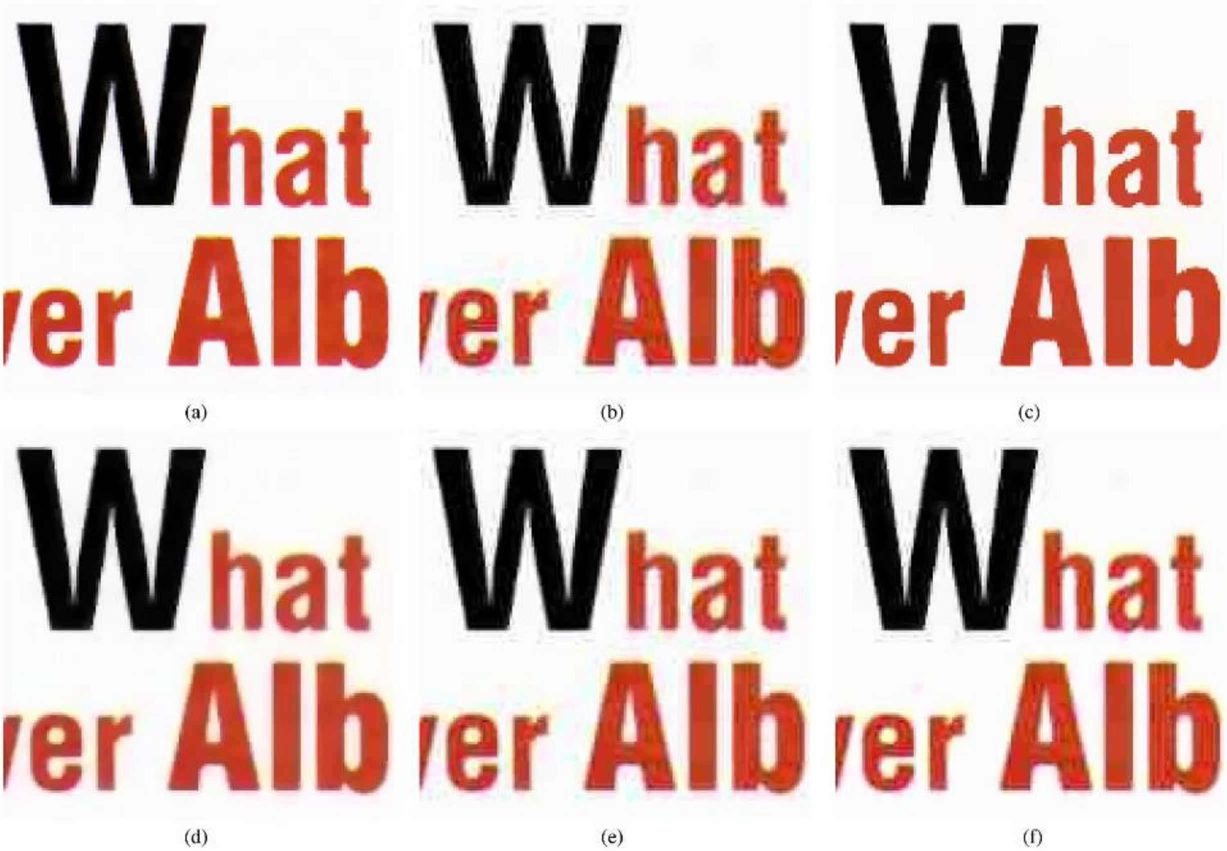


Fig. 15. Text region from Image 3. (a) Original. (b) Conventional JPEG decoding (c) The proposed algorithm. (d) Algorithm I [11]. (e) Algorithm II [34]. (f) Algorithm III [3]. For (f), only the text in red is decoded by the text decoding scheme of Algorithm III. The portion of the document corresponding to the letter "W" is decoded as picture by Algorithm III.

and blocking artifacts are largely eliminated. The blue sky in the decoded image looks smooth and natural. Although our scheme decodes the picture blocks with the conventional JPEG decoder, JPEG artifacts in these blocks are less revealing due to the significant presence of high-frequency components in these blocks. We should also point out that the original image in Fig. 16(a), if examined closely, also exhibits a small amount of blocking artifacts. This is typical in all the real world test images we collected, and is likely due to the lossy compression commonly employed by image capture devices. Because we used a high compression ratio to JPEG encode the original image in our experiment, none of the decoding schemes in Fig. 16 can accurately restore the artifacts.

Fig. 17 shows a region from Image 3 with most blocks classified as picture blocks. Among the five decoding schemes, Algorithm I in Fig. 17(d) has the best performance as far as reducing blocking artifacts is concerned. However, the smoothing due to the use of the MRF in Algorithm I also causes loss of detail in the decoded image. The problem is more pronounced in the highly textured picture blocks like those in the hair, moustache, and shoulder. The region decoded by Algorithm II in Fig. 17(e) looks very similar to that decoded by the conventional JPEG decoder in Fig. 17(b). In Fig. 17(f), Algorithm III reduces the blocking artifacts in the picture blocks without significant loss of detail. However, the sigma filter employed by Algorithm III is insufficient to reduce the blocking artifacts in the dark background. The region decoded by our scheme in Fig. 17(c) smooths out the blocking artifacts in the dark background blocks only, while the remaining picture blocks are decoded by the conventional JPEG decoder.

We now discuss the robustness of our algorithm with respect to various model assumptions and parameters. First, for some text blocks, the bi-level assumption of our text model may be violated, as in Fig. 18(a) and (b). In this case, the forward model [formulated in (2) and implemented through (25)–(27)] ensures that the decoded block is consistent with the encoded DCT coefficients. Because of this, we avoid decoding such an image block as a two-color block. This is demonstrated in Fig. 18(b).

Additionally, our algorithm is robust to segmentation errors. First, misclassification of image blocks to the background class does not cause significant artifacts. This is because processing of background blocks is unlikely to introduce artifacts since only the DC coefficient of background blocks is adjusted. Moreover, Figs. 18(c) and 18(d) show that even the misclassification of picture blocks to the text class does not typically result in significant artifacts. This is because such misclassified picture blocks typically contain image details with sharp edge transitions, so the decoded image still accurately represents the original image.

We also verify the robustness of the proposed algorithm to the variation of the parameters. In this experiment, we use a subset of four images from the 60 test images. Each image is JPEG encoded at four different bit rates, resulting in a total of 16 encoded images. In each test, we vary one of the parame-

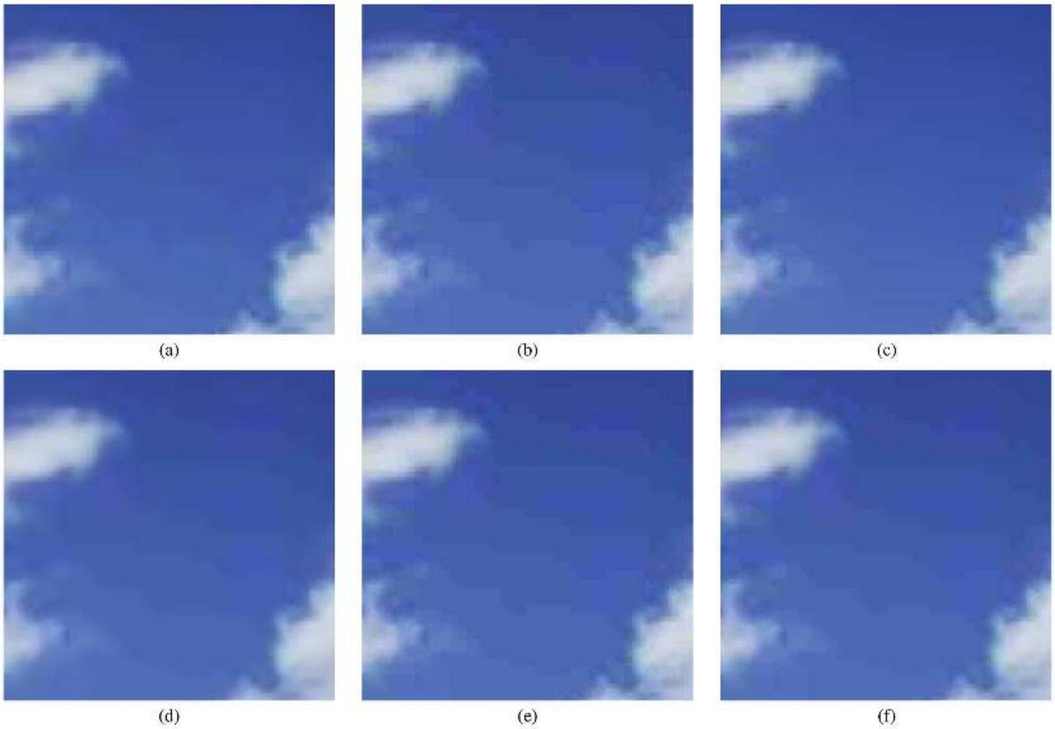


Fig. 16. Smooth region from Image 1. The image blocks corresponding to the blue sky are mostly labeled as background blocks by our segmentation algorithm, and the remaining blocks are labeled as picture blocks. (a) Original. (b) Conventional JPEG decoder. (c) The proposed algorithm. (d) Algorithm I [11]. (e) Algorithm II [34]. (f) Algorithm III [3].

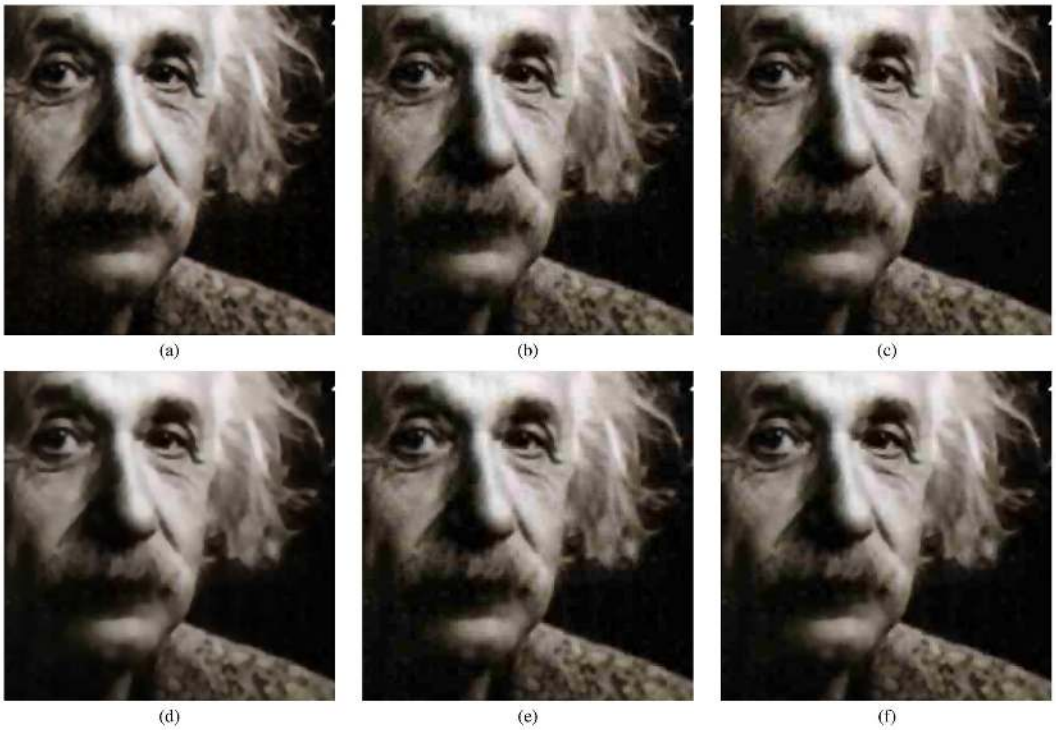


Fig. 17. Region from Image 3 containing mostly picture blocks. The image blocks corresponding to the face and shoulder are mostly labeled as picture blocks, and the remaining blocks are labeled as background blocks. (a) Original. (b) Conventional JPEG decoder. (c) The proposed algorithm. (d) Algorithm I [11]. (e) Algorithm II [34]. (f) Algorithm III [3].

ters in Table I (except $h_{r,s}$) over a $\pm 10\%$ interval and compute the average PSNR for the 16 decoded images. The maximum variation in the average PSNR, tabulated in Table II, shows that the algorithm is not sensitive to the choices of parameter values. Additionally, we have found no visually noticeable differences in the decoded images.

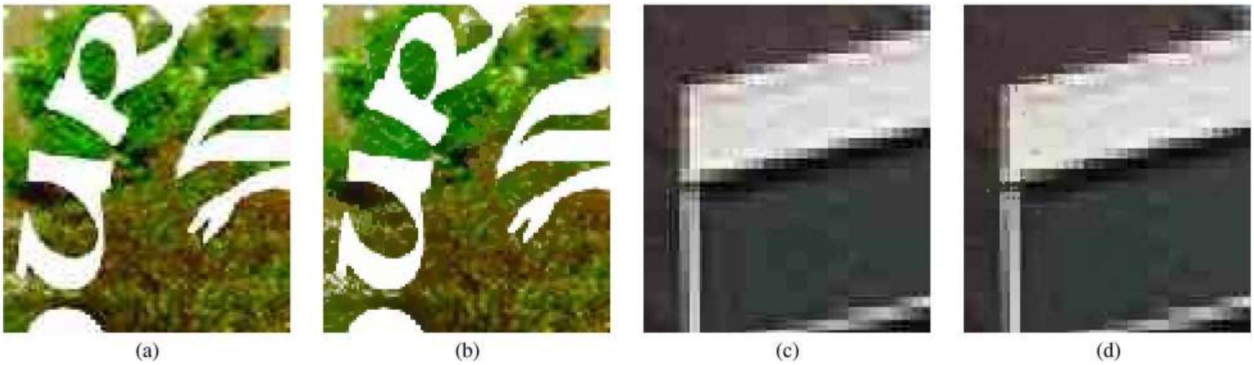


Fig. 18. Robustness of the proposed algorithm. (a), (b) Image patch where text blocks contain nonuniform background: (a) conventional JPEG decoder; (b) the proposed algorithm. (c), (d) Image patch where our segmentation algorithm misclassifies some of the picture blocks as text blocks: (c) conventional JPEG decoder; (d) the proposed algorithm.

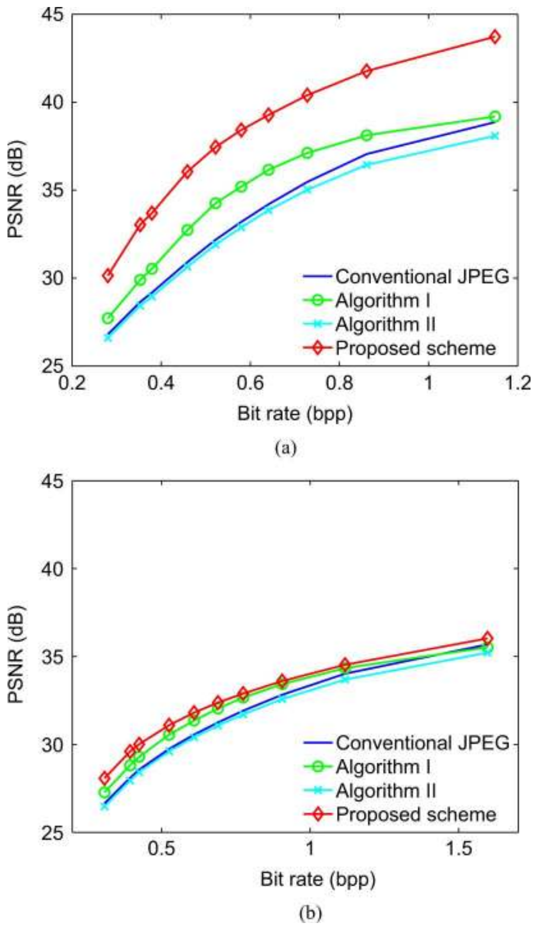


Fig. 19. Average PSNR versus average bit rate computed for 30 digital images in (a), and another 30 scanned images in (b).

TABLE II
MAXIMUM VARIATION IN PSNR WHEN EACH PARAMETER IS VARIED OVER A ±10% INTERVAL

Parameter	Range of values	Max. variation in PSNR
σ_W	4.5 – 5.5, increment of 0.2	0.08 dB
ν	10.8 – 13.2, increment of 0.4	0.03 dB
σ_C	3.0 – 4.0, increment of 0.2	0.01 dB
ϵ_{ac}	180 – 220, increment of 10	0.00 dB
τ	18 – 22, increment of 1	0.001 dB

Fig. 19 shows the rate-distortion curves for our algorithm and compares them to the Algorithms I and II and the conventional JPEG. For a range of different compression ratios, the figure shows average peak signal-to-noise ratio (PSNR) versus the average bit rates computed for our test set of 30 digital images in (a), and for the test set of 30 scanned images in (b). For the digital images, the proposed algorithm has a much better rate-distortion performance than the other three algorithms. Based on the segmentation results of the images encoded at the highest bit rate, 69%, 16%, and 15% of the image blocks are respectively labeled as background, text, and picture. For the set of scanned images, the rate-distortion performance of the proposed scheme is still better than that of the other three algorithms; however, the differences are less significant. In these images, the text regions contain scanning noise and other distortions. The removal of the scanning image noise by the proposed scheme can actually increase the mean squared error, despite of the improved visual quality. In the set of scanned images, 53%, 23%, and 24% of the blocks are respectively labeled as background, text, and picture.

VIII. CONCLUSION

We focused on the class of document images, and proposed a JPEG decoding scheme based on image segmentation. A major contribution of our research is on the use of a novel text model to improve the decoding quality of the text regions. From the results presented in Section VII, images decoded by our scheme are significantly improved, both visually and quantitatively, over the baseline JPEG decoding as well as three other approaches. In particular, the text regions decoded by our scheme are essentially free from ringing artifacts even when images are compressed with relatively low bit rate. The adaptive nature of the text model allows the foreground color and the background color to be estimated accurately without obvious color shift. Blocking artifacts in smooth regions are also largely eliminated.

REFERENCES

[1] G. K. Wallace, “The JPEG still picture compression standard,” *Commun. ACM*, vol. 34, no. 4, pp. 30–44, 1991.

[2] ISO/IEC 10918-1: Digital Compression and Coding of Continuous-Tone Still Images, Part 1, Requirements and Guidelines, International Organization for Standardization 1994.

[3] B. Oztan, A. Malik, Z. Fan, and R. Eschbach, "Removal of artifacts from JPEG compressed document images," presented at the SPIE Color Imaging XII: Processing, Hardcopy, and Applications, Jan. 2007.

[4] L. Bottou, P. Haffner, P. G. Howard, P. Simard, Y. Bengio, and Y. Lecun, "High quality document image compression with DjVu," *J. Electron. Imag.*, vol. 7, pp. 410–425, 1998.

[5] Mixed Raster Content (MRC), ITU-T Recommendation T.44, ITU, 2005.

[6] K. Ramchandran and M. Vetterli, "Rate-distortion optimal fast thresholding with complete JPEG/MPEG decoder compatibility," *IEEE Trans. Image Process.*, vol. 3, pp. 700–704, 1994.

[7] M. G. Ramos and S. S. Hemami, "Edge-adaptive JPEG image compression," *Vis. Commun. Image Process.*, vol. 2727, no. 1, pp. 1082–1093, 1996.

[8] K. Konstantinides and D. Tretter, "A JPEG variable quantization method for compound documents," *IEEE Trans. Image Process.*, vol. 9, no. 7, pp. 1282–1287, Jul. 2000.

[9] A. Zakhor, "Iterative procedures for reduction of blocking effects in transform image coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 2, no. 3, pp. 91–95, Mar. 1992.

[10] Y. Yang, N. Galatsanos, and A. Katsaggelos, "Regularized reconstruction to reduce blocking artifacts of block discrete cosine transform compressed images," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 3, no. 12, pp. 421–432, Dec. 1993.

[11] T. O'Rourke and R. Stevenson, "Improved image decompression for reduced transform coding artifacts," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 5, no. 12, pp. 490–499, Dec. 1995.

[12] T. Meier, K. Ngan, and G. Crebbin, "Reduction of blocking artifacts in image and video coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 9, no. 4, pp. 490–500, Apr. 1999.

[13] T. Chen, H. Wu, and B. Qiu, "Adaptive postfiltering of transform coefficients for the reduction of blocking artifacts," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 11, no. 5, pp. 594–602, May 2001.

[14] A. Averbuch, A. Schclar, and D. Donoho, "Deblocking of block-transform compressed images using weighted sums of symmetrically aligned pixels," *IEEE Trans. Image Process.*, vol. 14, no. 2, pp. 200–212, Feb. 2005.

[15] Z. Fan and R. Eschbach, "JPEG decompression with reduced artifacts," in *Proc. SPIE & IS&T Symp. Electronic Imaging: Image and Video Compression*, Jan. 1994, vol. 2186, pp. 50–55.

[16] M.-Y. Shen and C.C.-J. Kuo, "Review of postprocessing techniques for compression artifact removal," *J. Vis. Commun. Image Represent.*, vol. 9, no. 1, pp. 2–14, Mar. 1998.

[17] G. Aharoni, A. Averbuch, R. Coifman, and M. Israeli, "Local cosine transform—A method for the reduction of the blocking effect in JPEG," *J. Math. Imag. Vis.*, vol. 3, no. 1, pp. 7–38, Mar. 1993.

[18] T. Meier, K. N. Ngan, and G. Crebbin, "Reduction of blocking artifacts in image and video coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 9, no. 4, pp. 490–500, Apr. 1999.

[19] E. Hamilton, *JPEG File Interchange Format* 1992, C-Cube Microsystems.

[20] Recommendation ITU-R BT.601, Encoding Parameters of Digital Television for Studios Geneva, Switzerland, ITU, 1992.

[21] A. K. Jain*, Fundamentals of Digital Image Processing*, 1st ed. Upper Saddle River, NJ: Prentice-Hall, 1989, ch. 5, pp. 150–154.

[22] M. Anderson, R. Motta, S. Chandrasekar, and M. Stokes, "Proposal for a standard default color space for the internet-sRGB," in *Proc. IS&T/SID 4th Color Imaging Conf.*, Scottsdale, AZ, Nov. 1996, pp. 238–246.

[23] H. L. Van Trees*, Detection, Estimation, and Modulation Theory, Part I*, 1st ed. New York: Wiley, 1968, pp. 54–63.

[24] J. Besag, "On the statistical analysis of dirty pictures," *J. Roy. Statist. Soc.*, vol. 48, no. 3, pp. 259–302, 1986.

[25] J. Besag, "Spatial interaction and the statistical analysis of lattice systems," *J. Roy. Statist. Soc.*, vol. 36, no. 2, pp. 192–236, 1974.

[26] T. Porter and T. Duff, "Compositing digital images," *SIGGRAPH Comput. Graph.*, vol. 18, no. 3, pp. 253–259, 1984.

[27] J. Zheng, S. S. Saquib, K. Sauer, and C. A. Bouman, "Parallelizable Bayesian tomography algorithms with rapid, guaranteed convergence," *IEEE Trans. Image Process.*, vol. 9, no. 10, pp. 1745–1759, Oct. 2000.

[28] D. R. Hunter and K. Lange, "A tutorial on MM algorithms," *Amer. Statist.*, vol. 58, no. 1, pp. 30–37, Feb. 2004.

[29] J. McQueen, "Some methods for classification and analysis of multivariate observations," in *Proc. 5th Berkeley Symp. Mathematical Statistics and Probability*, 1967, pp. 281–297.

[30] R. L. de Queiroz, "Processing JPEG-compressed images and documents," *IEEE Trans. Image Process.*, vol. 7, no. 12, pp. 1661–1672, Dec. 1998.

[31] C. A. Bouman and M. Shapiro, "A multiscale random field model for Bayesian image segmentation," *IEEE Trans. Image Process.*, vol. 3, no. 2, pp. 162–177, Mar. 1994.

[32] C. A. Bouman, Cluster: An Unsupervised Algorithm for Modeling Gaussian Mixtures Apr. 1997 [Online]. Available: http://www.ece.purdue.edu/bouman/software/cluster

[33] H. Siddiqui and C. A. Bouman, "Training-based descreening," *IEEE Trans. Image Process.*, vol. 16, no. 3, pp. 789–802, Mar. 2007.

[34] E. Y. Lam, "Compound document compression with model-based biased reconstruction," *J. Electron. Imag.*, vol. 13, pp. 191–197, 2004.

Tak-Shing Wong received the B.Eng. degree in computer engineering and the M.Phil. degree in electrical and electronic engineering from the Hong Kong University of Science and Technology in 1997 and 2000, respectively. He is currently pursuing the Ph.D. degree at the School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN.

His research interests are in image segmentation, document image analysis, and processing.

Charles A. Bouman (S'86–M'89–SM'97–F'01) received the B.S.E.E. degree from the University of Pennsylvania, Philadelphia, in 1981, the M.S. degree from the University of California, Berkeley, in 1982, and the Ph.D. degree in electrical engineering from Princeton University, Princeton, NJ, in 1989.

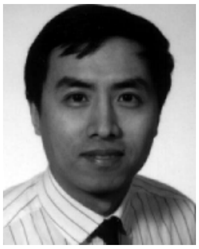
From 1982 to 1985, he was a full staff member at the Massachusetts Institute of Technology Lincoln Laboratory. In 1989, he joined the faculty of Purdue University, West Lafayette, IN, where he is a Professor with a primary appointment in the School of Electrical and Computer Engineering and a secondary appointment in the School of Biomedical Engineering. Currently, he is Co-Director of Purdue's Magnetic Resonance Imaging Facility located in Purdue's Research Park. His research focuses on the use of statistical image models, multiscale techniques, and fast algorithms in applications including tomographic reconstruction, medical imaging, and document rendering and acquisition.

Prof. Bouman is a Fellow of the IEEE, a Fellow of the American Institute for Medical and Biological Engineering (AIMBE), a Fellow of the society for Imaging Science and Technology (IS&T), a Fellow of the SPIE professional society, a recipient of IS&T's Raymond C. Bowman Award for outstanding contributions to digital imaging education and research, and a University Faculty Scholar of Purdue University. He is currently the Editor-in-Chief of the IEEE TRANSACTIONS ON IMAGE PROCESSING and a member of the IEEE Biomedical Image and Signal Processing Technical Committee. He has been a member of the Steering Committee for the IEEE TRANSACTIONS ON MEDICAL IMAGING and an Associate Editor for the IEEE TRANSACTIONS ON IMAGE PROCESSING and the IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE. He has also been Co-Chair of the 2006 SPIE/IS&T Symposium on Electronic Imaging, Co-Chair of the SPIE/IS&T Conferences on Visual Communications and Image Processing 2000 (VCIP), a Vice President of Publications and a member of the Board of Directors for the IS&T Society, and he is the founder and Co-Chair of the SPIE/IS&T Conference on Computational Imaging.

Ilya Pollak received the B.S. and M.Eng. degrees in 1995 and the Ph.D. degree in 1999 from the Massachusetts Institute of Technology, Cambridge, all in electrical engineering.

From 1999–2000, he was a postdoctoral researcher at the Division of Applied Mathematics, Brown University, Providence, RI. Since 2000, he has been with Purdue University, West Lafayette, IN, where he is currently an Associate Professor of Electrical and Computer Engineering. He has held visiting positions at INRIA (The French National Institute for Research in Computer Science and Control), Sophia Antipolis, France; Tampere University of Technology, Finland; and Jefferies, Inc., New York. His research interests are in image and signal processing, specifically hierarchical statistical models, fast estimation algorithms, nonlinear scale spaces, adaptive representations with applications to image and video compression, segmentation, classification, and financial time series analysis.

Prof. Pollak received a CAREER award from the National Science Foundation in 2001. He received an Eta Kappa Nu Outstanding Faculty Award in 2002 and in 2007 and a Chicago-Area Alumni Young Faculty Award in 2003. He is an Associate Editor of the IEEE Transactions on Image Processing.He is a Co-Chair of the SPIE/IS&T Conference on Computational Imaging.



Zhigang Fan received the M.S. and Ph.D. degrees in electrical engineering from the University of Rhode Island, Kingston, in 1986 and 1988, respectively.

He joined Xerox Corporation in 1988 where he is currently a principal scientist in Xerox Corporate Research and Technology. His research interests include various aspects of image processing and recognition, in particular, color imaging, document image segmentation and analysis, anti-counterfeit, and security printing. He has authored and coauthored more than 70 technical papers, as well as over 150 patents and pending applications.

Dr. Fan is an Associate Editor for the IEEE TRANSACTIONS ON IMAGE PROCESSING.