ENHANCEMENT AND ARTIFACT REMOVAL FOR TRANSFORM CODED DOCUMENT IMAGES

A Dissertation

Submitted to the Faculty

of

Purdue University

by

Tak Shing Wong

In Partial Fulfillment of the

Requirements for the Degree

of

Doctor of Philosophy

May 2011

Purdue University

West Lafayette, Indiana

ACKNOWLEDGMENTS

I would like to convey my deepest and sincere gratitude to my advisers, Professor Charles A. Bouman and Professor Ilya Pollak, for their invaluable guidance, support, and encouragement throughout the course of my doctoral studies. Professor Bouman is enthusiastic, insightful, meticulous, and has an uncompromising attitude towards perfection. All these characters of him have deep and positive impacts on me. Professor Pollak has been following my progress closely in the past few years. I sincerely thank for the end-less hours he had spent on reviewing my work, and the many important sugguestions and assistance he provided me. From both of them, I have learnt many important lessons which continuously push me to improve myself everyday. I am also thankful to Professor Jan P. Allebach and Professor George T.-C. Chiu for dedicating their precious time to serve on my advisory committee.

I like to sincerely thank all my friends and colleagues in EISL and at Purdue, especially Eri Haneda, Guangzhi Cao, Zhou Yu, Maria Vlachopoulou, Landis Huffman, Hasib Siddiqui, Buyue Zhang, Maribel Figuera Alegre, Jianing Wei, Guotong Feng, Haolin Feng, Chun Kin Au Yeung, and Jianming Lin. They have been providing me help in many different ways, and make my life at Purdue memorable and worthwhile.

I am grateful to Xerox Corporation for the generous financial support, and Zhigang Fan at Xerox Webster Research Center for the insightful discussions and remarks on my research. My internship at the USRC, Sony Electronic Inc. was an extremely wonderful and fruitful experience, which was mainly attributed to my mentor, Farhan A. Baqai. During the same period, Xiaogang Dong provided me with assistance of all kinds, to which I am also indebted.

I am always thankful to my beloved parents and my sister for their love and providing me with a warm family to grow up in. My wholehearted thanks go to my wife, Dewen Lai, who has been very supportive to my undertaking postgraduate

studies from the beginning. She has always been a special and important one in my life, and I cherish everything she has done for me.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Figure Page

ABSTRACT

Wong, Tak Shing Ph.D., Purdue University, May 2011. Enhancement and Artifact Removal for Transform Coded Document Images . Major Professors: Charles A. Bouman and Ilya Pollak.

Despite several more advanced image compression algorithms have been proposed, JPEG is still widely used because of its advantage in simplicity. However, images encoded by the JPEG algorithm exhibit undesirable blocking and ringing artifacts. In particular, for document images, ringing artifacts reduce the shapeness and clarity of the text, and affect the readability of the documents. This dissertation presents two different approaches to improve the decoding quality for document images encoded with JPEG.

In the first approach, we pose the JPEG decoding problem as an inverse problem and decode the image with Bayesian reconstruction. The scheme works by first segmenting the image into blocks of three classes corresponding to background, text, and picture. For each class of image blocks, we design a specific prior model to capture the characteristics of the class. The class-specific prior models are then used to compute the maximum *a posteriori* (MAP) estimate of the original image. The scheme substantially improves the quality of decoded images both visually and as measured by PSNR. Also, the decoded text regions are essentially free from ringing artifacts even when the images are compressed at a low bit rate.

In the second approach, we introduce the Hypothesis Selection Filter (HSF) as a generic approach for image quality enhancement. The HSF provides a systematic method for combining the advantages of multiple linear or nonlinear image filters into a single general framework. Our major contributions include the basic architecture of the HSF and a novel unsupervised training procedure for the design of an optimal pixel classifier. The resulting classifier distinguishes the different types of image content and appropriately adjusts the weighting factors of the image filters so that each filter is applied to the regions for which it is most appropriate. We demonstrate the effectiveness of the HSF by applying it as a post-processing step for JPEG decoding so as to reduce the JPEG artifacts in the decoded document image. In our scheme, we incorporated 4 different image filters for reducing the JPEG artifacts in the different types of image content that are common in document images, like text, graphics, and natural images. Based on several evaluation methods, including visual inspection of a variety of image patches with different types of content, global PSNR, and a global blockiness measure, our method outperforms state-of-the-art JPEG decoding methods. In addition, the generic structure and the training basis of HSF makes the scheme potentially applicable to many other quality enhancement and image reconstruction tasks.

1. A DOCUMENT IMAGE MODEL AND ESTIMATION ALGORITHM FOR OPTIMIZED JPEG DECOMPRESSION

1.1 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 post-processing 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 post-filtering 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 post-processing 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 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 1.7, 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 1.2. For the luminance component, we then present the prior models used to decode the background blocks and the text blocks in Section 1.3, and the MAP reconstruction algorithms in Section 1.4. We introduce our block based segmentation algorithm in Section 1.5. Following this, in Section 1.6, 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 1.7, we present the experimental results and compare our scheme with three other existing JPEG decoding algorithms.

1.2 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 non-overlapping 8×8 blocks, and each block from the components undergoes the three steps of forward

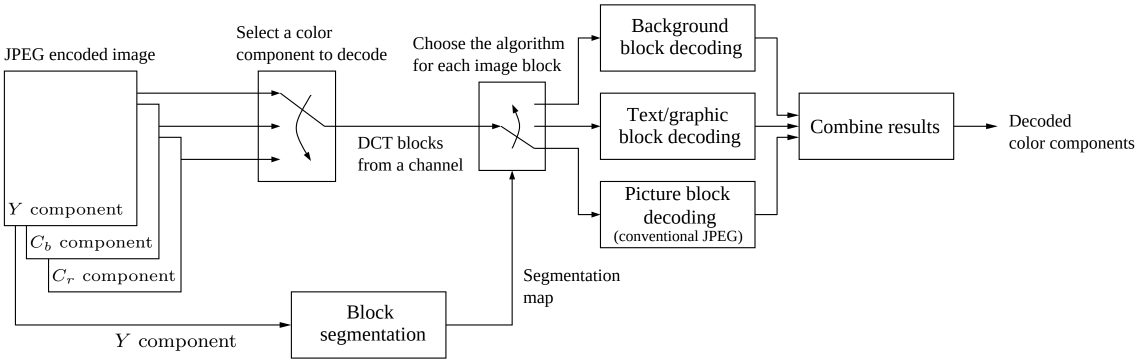

JPEG encoded image
Y component
C_b component
C_r component
Select a color component to decode
DCT blocks from a channel
Choose the algorithm for each image block
Background block decoding
Text/graphic block decoding
Picture block decoding (conventional JPEG)
Combine results
Decoded color components
Segmentation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

Fig. 1.1. 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.

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. 1.1 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 s-th block. 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 $\tilde{Y}_{s,i} = Q_i \text{ round}\left[\frac{Y_{s,i}}{Q_i}\right]$, 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 maximum *a posteriori* probability (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

$$p(\tilde{y}_s|x_s) = \begin{cases} 1, & \text{if } T(Dx_s) = \tilde{y}_s \\ 0, & \text{otherwise.} \end{cases} \tag{1.1}$$

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

$$p(\tilde{y}|x) = \prod_s p(\tilde{y}_s|x_s) = \begin{cases} 1, & \text{if } T(Dx_s) = \tilde{y}_s \text{ for all } s \\ 0, & \text{otherwise.} \end{cases} \tag{1.2}$$

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

[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.

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

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

Referring to (1.2), 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 (1.2), 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

$$\hat{x} = \arg\min_{x} \left[-\log p(x) \right] \quad \text{subject to } T(Dx_s) = \tilde{y}_s \text{ for all } s. \tag{1.3}$$

In practice, we solve the optimization problem (1.3) 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

$$\hat{x}^b = \arg\min_{x^b} \left[-\log p(x^b) \right] \tag{1.4}$$

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

$$(\hat{x}^t, \hat{\phi}) = \arg\min_{x^t, \phi} \left[-\log p(x^t, \phi) \right] \tag{1.5}$$

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.

1.3 Prior Models for the Luminance Blocks

1.3.1 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 $K_{bb} = \big\{\{r,s\} : r \text{ and } s \text{ are neighbor background blocks}\big\}$. 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

$$p(x^b) = \frac{1}{\text{const}} \exp\left\{ -\frac{1}{2\sigma_{_B}^2} \sum_{\{r,s\}\in K_{bb}} h_{r,s}(\mu_r - \mu_s)^2 \right\}, \tag{1.6}$$

where $\sigma_{_B}^2$ 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 $h_{r,s} = \frac{1}{6}$ if r and s are horizontal or vertical neighbors, and $h_{r,s} = \frac{1}{12}$ if r and s are diagonal neighbors.

1.3.2 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 i-th pixel in block s is given by

$$X_{s,i} = \alpha_{s,i}C_{1,s} + (1-\alpha_{s,i})C_{2,s} + W_{s,i}\,, \tag{1.7}$$

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 $\{C_1, C_2\}$. It then follows from (1.7) 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

$$p(x^t|c_1,c_2,\alpha) = \frac{1}{\text{const}} \exp\left\{ -\frac{1}{2\sigma_W^2} \sum_{s \text{ text block}} \|x_s - \alpha_s c_{1,s} - (\mathbf{1} - \alpha_s)c_{2,s}\|^2 \right\}, \qquad (1.8)$$

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 $0 \le \alpha_{s,i} \le 1$ 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

$$p(\alpha) = \begin{cases} \dfrac{1}{\text{const}} \exp\left\{ \nu \displaystyle\sum_{s \text{ text block}} \|\alpha_s - \tfrac{1}{2}\mathbf{1}\|^2 \right\}, & 0 \le \alpha_{s,i} \le 1 \text{ for all } s, i \\ 0, & \text{otherwise.} \end{cases} \qquad (1.9)$$

As shown in Fig. 1.2, 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.

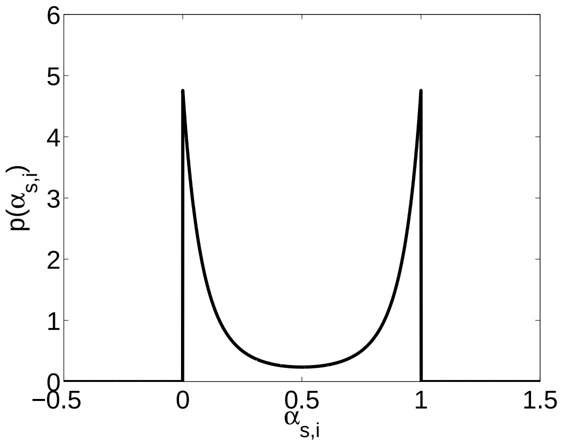


Fig. 1.2. The marginal probability density function of an alpha value $\alpha_{s,i}$, for $\nu = 12$. As the alpha value controls the proportion of the two intensities $C_{1,s}$ and $C_{2,s}$ present in a text pixel value, the density function's support is $[0, 1]$. The bimodal nature of the density function with peaks at 0 and 1 models the clustering of the text pixel values around $C_{1,s}$ and $C_{2,s}$.

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 1.4.1. Then, our model for C_1 and C_2 is expressed by the Gibbs distribution

$$\begin{aligned} p(c_1, c_2) = \frac{1}{\text{const}} \exp\left\{ -\frac{1}{2\sigma_C^2} \sum_{\{s,r\} \in K_{tt}} \Big(\rho(c_{1,s} - c_{1,r}) + \rho(c_{2,s} - c_{2,r}) \Big) \right\} \\ \times \exp\left\{ -\frac{1}{2\sigma_C^2} \sum_{\{s,r\} \in K_{tb}} \rho\Big(\min(|c_{1,s} - \hat{\mu}_r|, |c_{2,s} - \hat{\mu}_r|) \Big) \right\}, \end{aligned} \tag{1.10}$$

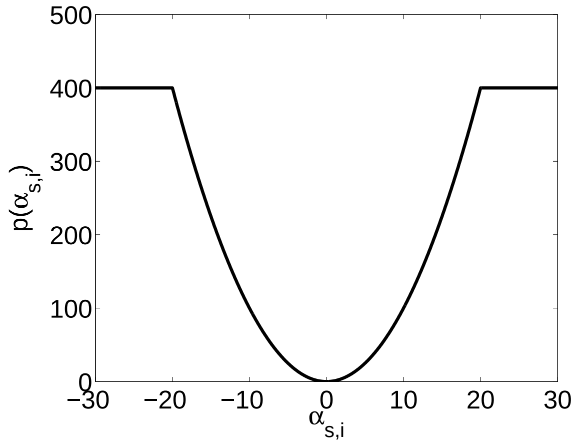


Fig. 1.3. The potential function $\rho(x) = \min(x^2, \tau^2)$, $\tau = 20$, of the Markov random fields used to characterize the spatial variation of the predominant colors $C_{1,s}$ and $C_{2,s}$. The threshold parameter τ ensures that we avoid excessively penalizing large intensity difference between the two corresponding predominant colors of two neighboring blocks.

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. 1.3. The first exponential function of (1.10) 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 (1.10) 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 (1.8), (1.9) and (1.10), the prior model for text blocks of the luminance component is given by

$$\begin{aligned}-\log p(x^t, c_1, c_2, \alpha) = & \frac{1}{2\sigma_W^2} \sum_{s \text{ text block}} \|x_s - \alpha_s c_{1,s} - (\mathbf{1} - \alpha_s) c_{2,s}\|^2 \\ & + \frac{1}{2\sigma_C^2} \sum_{\{s,r\} \in K_{tt}} \big(\rho(c_{1,s} - c_{1,r}) + \rho(c_{2,s} - c_{2,r})\big) \\ & + \frac{1}{2\sigma_C^2} \sum_{\{s,r\} \in K_{tb}} \rho(\min(|c_{1,s} - \hat{\mu}_r|, |c_{2,s} - \hat{\mu}_r|)) \\ & - \nu \sum_{s \text{ text block}} \|\alpha_s - \tfrac{1}{2}\mathbf{1}\|^2 + \text{const.} \qquad (1.11)\end{aligned}$$

1.4 Optimization for Decoding the Luminance Component

To decode the luminance component, we need to solve the optimization problems (1.4) and (1.5) with the specific prior models (1.6) for the background blocks and (1.11) 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.

1.4.1 Optimization for Decoding the Luminance Background Blocks

To decode the luminance background blocks, we minimize $-\log p(x^b)$ of (1.6) 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

$$\mathcal{C}(y^b) = \frac{1}{128\sigma_B^2} \sum_{\{r,s\}\in K_{bb}} h_{r,s}(y_{r,0} - y_{s,0})^2, \tag{1.12}$$

where y^b is the vector containing the DCT coefficients y_s of all the background blocks. We minimize the cost function (1.12) 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

$$y_{s,0} \leftarrow \text{clip}\left(\frac{\sum_{r:\{r,s\}\in K_{bb}} h_{r,s}\, y_{r,0}}{\sum_{r:\{r,s\}\in K_{bb}} h_{r,s}},\ \left[\tilde{y}_{s,0} - \frac{Q_0}{2}, \tilde{y}_{s,0} + \frac{Q_0}{2}\right]\right), \tag{1.13}$$

where 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.

1.4.2 Optimization for Decoding the Luminance Text Blocks

In order to decode the luminance text blocks, we must minimize the cost function of (1.11) 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

$$a = \frac{(c_{2,s} - c_{1,s})^2}{2\sigma_W^2} - \nu, \tag{1.14}$$

$$b = \frac{(c_{2,s} - c_{1,s})(x_{s,i} - c_{2,s})}{\sigma_W^2} + \nu. \tag{1.15}$$

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

$$\alpha_{s,i}^* = -\frac{b}{2a} = \frac{\nu\sigma_W^2 + (c_{2,s} - c_{1,s})(x_{s,i} - c_{2,s})}{2\nu\sigma_W^2 - (c_{2,s} - c_{1,s})^2}. \tag{1.16}$$

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 $\alpha_{s,i}^* > \frac{1}{2}$ 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

$$\alpha_{s,i} \leftarrow \begin{cases} \text{clip}(\alpha_{s,i}^*, [0,1]), & \text{if } a > 0 \\ \text{step}(\frac{1}{2} - \alpha_{s,i}^*), & \text{if } a < 0 \\ \text{step}(-b), & \text{if } a = 0 \end{cases} \tag{1.17}$$

where 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

$$F(c_{1,s}, c_{2,s}) = \frac{1}{2\sigma_W^2} \|x_s - \alpha_s c_{1,s} - (\mathbf{1} - \alpha_s)c_{2,s}\|^2 + \frac{1}{2\sigma_C^2} \sum_{r \in \partial s} f_r(c_{1,s}, c_{2,s}), \tag{1.18}$$

where ∂s is the set of the non-picture neighbor blocks of s, and $f_r(c_{1,s}, c_{2,s})$ is given by

$$f_r(c_{1,s}, c_{2,s}) = \begin{cases} \rho(c_{1,s} - c_{1,r}) + \rho(c_{2,s} - c_{2,r}), & \text{if } r \text{ is a text block} \\ \rho(\min(|c_{1,s} - \hat{\mu}_r|, |c_{2,s} - \hat{\mu}_r|)), & \text{if } r \text{ is a background block.} \end{cases} \tag{1.19}$$

Unfortunately, $f_r(c_{1,s}, c_{2,s})$ is a non-convex 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

$$\tilde{f}_r(c_{1,s}, c_{2,s}) = a_{1,r}|c_{1,s} - b_{1,r}|^2 + a_{2,r}|c_{2,s} - b_{2,r}|^2, \tag{1.20}$$

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

$$a_{1,r} = \begin{cases} \text{step}(\tau - |c'_{1,s} - c_{1,r}|), & \text{if } r \text{ is a text block} \\ \text{step}(\tau - |c'_{1,s} - \hat{\mu}_r|)\ \text{step}(|c'_{2,s} - \hat{\mu}_r| - |c'_{1,s} - \hat{\mu}_r|), & \text{if } r \text{ is a background block} \end{cases} \tag{1.21}$$

$$a_{2,r} = \begin{cases} \text{step}(\tau - |c'_{2,s} - c_{2,r}|), & \text{if } r \text{ is a text block} \\ \text{step}(\tau - |c'_{2,s} - \hat{\mu}_r|)\ \text{step}(|c'_{1,s} - \hat{\mu}_r| - |c'_{2,s} - \hat{\mu}_r|), & \text{if } r \text{ is a background block} \end{cases} \tag{1.22}$$

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 step$(A - B)$ simply captures the inequality test $A > B$.

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

$$\tilde{F}(c_{1,s}, c_{2,s}) = \frac{1}{2\sigma_W^2}\|x_s - \alpha_s c_{1,s} - (\mathbf{1} - \alpha_s)c_{2,s}\|^2 + \frac{1}{2\sigma_C^2}\sum_{r \in \partial s} \tilde{f}_r(c_{1,s}, c_{2,s}) \ . \tag{1.23}$$

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

$$(c_{1,s}, c_{2,s}) \leftarrow \arg\min_{c_{1,s}, c_{2,s}} \tilde{F}(c_{1,s}, c_{2,s}). \tag{1.24}$$

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 following three steps.

$$y_s \leftarrow D(\alpha_s c_{1,s} + (\mathbf{1} - \alpha_s) c_{2,s}) \tag{1.25}$$

$$y_{s,i} \leftarrow \text{clip}\left(y_{s,i}, \left[\tilde{y}_{s,i} - \frac{Q_i}{2}, \tilde{y}_{s,i} + \frac{Q_i}{2} \right] \right) \qquad \text{for } i = 0, \ldots, 63 \tag{1.26}$$

$$x_s \leftarrow D^{-1} y_s. \tag{1.27}$$

The quantity $\alpha_s c_{1,s} + (\mathbf{1} - \alpha_s) c_{s,2}$ is first transformed to the DCT domain in (1.25). Then (1.26) clips these DCT coefficients to the respective ranges they are known to be within. Finally in (1.27), 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.

Fig. 1.4 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.

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 (1.17)
        /* 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 (1.20)-(1.22)
        $(c_{1,s}, c_{2,s}) \leftarrow \arg\min_{c_{1,s}, c_{2,s}} \tilde{F}(c_{1,s}, c_{2,s})$

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

Fig. 1.4. 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.

1.5 Block-Based Segmentation

Our segmentation algorithm classifies each luminance block as one of three classes: background, text, and picture. Fig. 1.5 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 i-th 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.

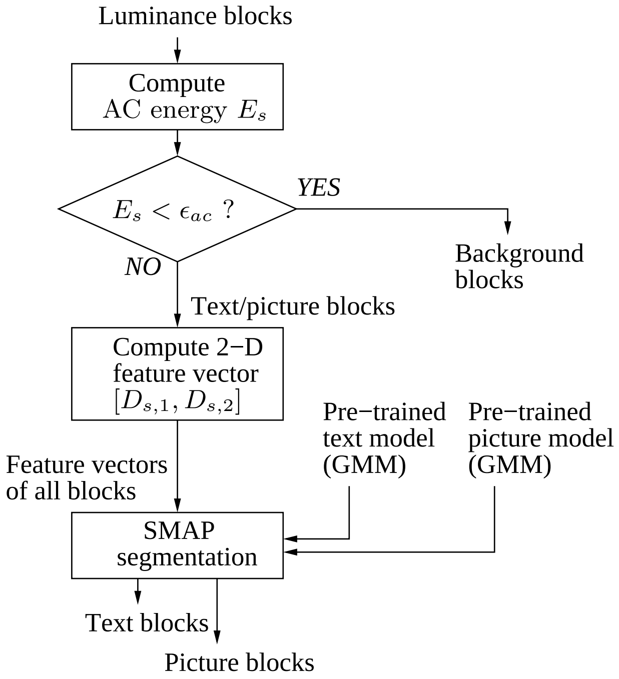


Fig. 1.5. Block-based segmentation. The background blocks are first identified by AC energy thresholding. A 2-D feature vector is then computed for each block. Two Gaussian mixture models are obtained from supervised training: one for the text class and one for the picture class. With these two models, 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.

Next, we compute a two-dimensional 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 non-text 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

$$D_{s,1} = \lambda^{\gamma} \times \text{encoding length of block } s, \tag{1.28}$$

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

$$D_{s,2} = \frac{\sum_{i=0}^{63} \min\left\{ |\tilde{x}_{s,i} - \theta_{1,s}|^2, |\tilde{x}_{s,i} - \theta_{2,s}|^2 \right\}}{|\theta_{1,s} - \theta_{2,s}|^2}. \tag{1.29}$$

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.

Lastly, we describe the training process which determines the parameter γ in (1.28) 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 with 9 different quantization matrices, corresponding to λ_j with $j = 1, \ldots, 9$. For the i-th image encoded by the j-th 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:

$$\hat{\gamma} = \arg\min_{\gamma} \min_{u,v} \sum_{i=1}^{54} \sum_{j=1}^{9} \left[(\lambda_j^{\gamma}\, u_{i,j} - u)^2 + (\lambda_j^{\gamma}\, v_{i,j} - v)^2 \right]. \qquad (1.30)$$

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.

1.6 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. 1.6.

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

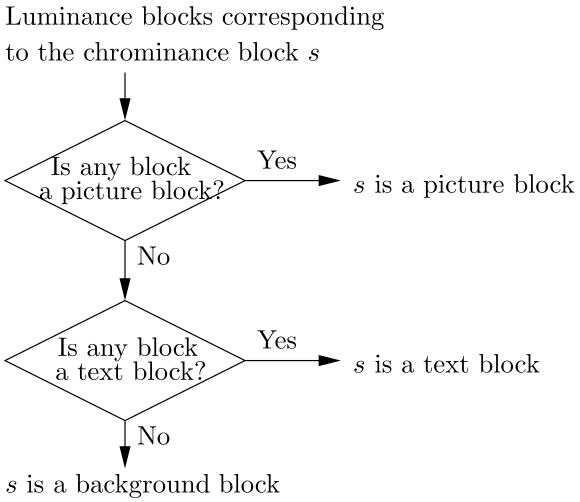

Luminance blocks corresponding
to the chrominance block s
Is any block a picture block?
Yes
s is a picture block
No
Is any block a text block?
Yes
s is a text block
No
s is a background block

Fig. 1.6. 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.

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.

The optimization for decoding the chrominance text blocks is similar to the algorithm described in Section 1.4.2 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

$$(c_{1,s}, c_{2,s}) = \arg\min_{c_{1,s}, c_{2,s}} \|\tilde{x}_s - \hat{\alpha}_s c_{1,s} - (\mathbf{1} - \hat{\alpha}_s) c_{2,s}\|^2, \tag{1.31}$$

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. 1.4.

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. 1.7 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.

1.7 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 1.1 summarizes the parameter values chosen for the proposed algorithm. In decoding the background blocks, the parameter σ_B^2 in the cost function (1.12) is a positive multi-

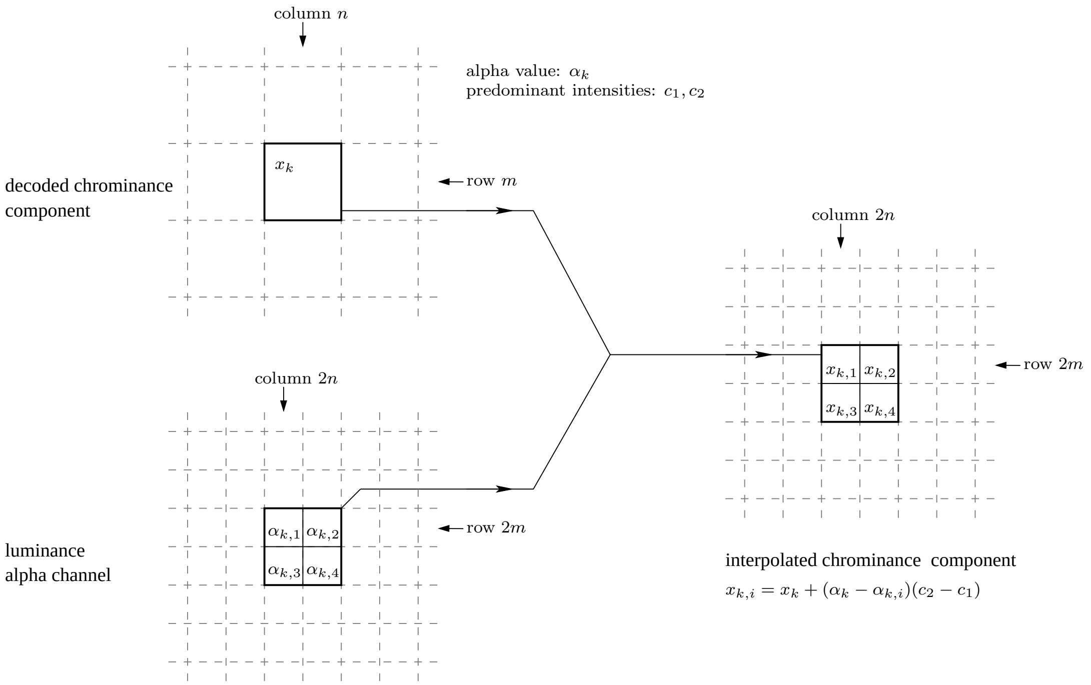

column n
alpha value: α_k
predominant intensities: c_1, c_2
decoded chrominance component
x_k
row m
column $2n$
row $2m$
$x_{k,1}$ $x_{k,2}$
$x_{k,3}$ $x_{k,4}$
column $2n$
luminance alpha channel
$\alpha_{k,1}$ $\alpha_{k,2}$
$\alpha_{k,3}$ $\alpha_{k,4}$
row $2m$
interpolated chrominance component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

Fig. 1.7. 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.

plicative constant whose value is irrelevant in determining the minimizer. Therefore, it is omitted from Table 1.1.

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 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

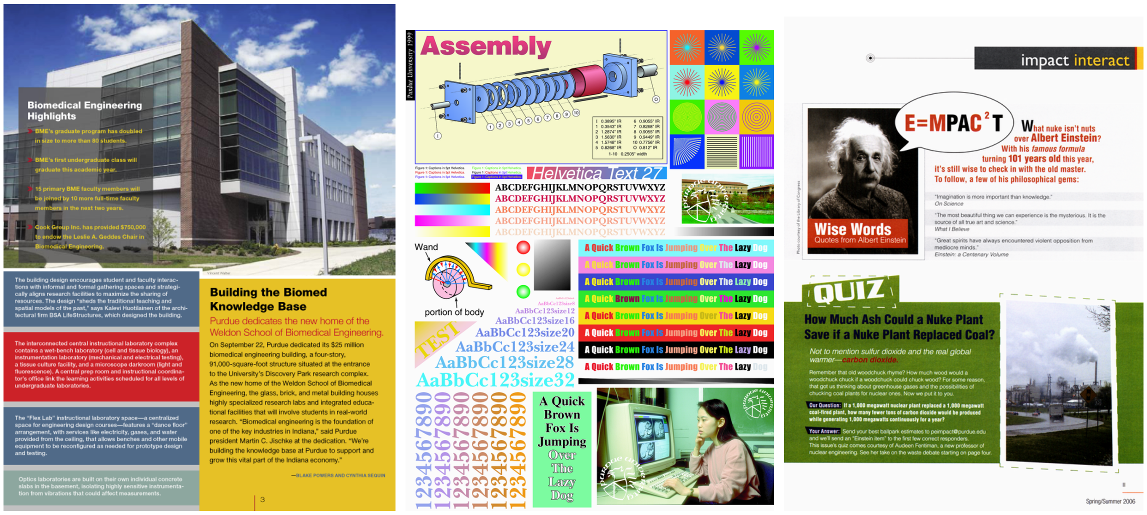
Table 1.1
Parameter values selected for the proposed algorithm.

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. (1.6)
σ_W	5	Eq. (1.8)
ν	12	Eq. (1.9)
σ_C	3.5	Eq. (1.10)
ϵ_{ac}	200	Section 1.5
τ	20	ρ in Section 1.3.2

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. 1.8. 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 1.5, to the JPEG encoded images. Fig. 1.9 shows that the 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.

Fig. 1.10 and Fig. 1.11 demonstrate the improvement in text block decoding using the proposed algorithm. Fig. 1.10(a) shows the luminance component of a small text region computed from Image 1. A small region within Fig. 1.10(a) is further enlarged in Fig. 1.10(b) to show the fine details. Fig. 1.10(c) and (d) show the region of the JPEG encoded image decoded by the conventional JPEG decoder. The decoded

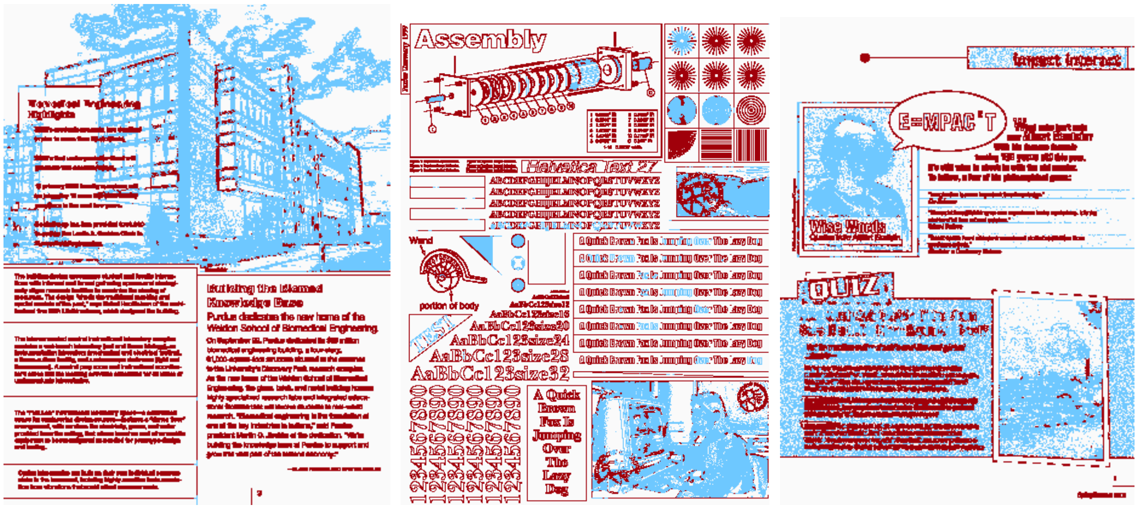

Biomedical Engineering Highlights
Building the Biomed Knowledge Base
Assembly
Helvetica Text 27
impact interact
E=MPACT
Wise Words
QUIZ
How Much Ash Could a Nuke Plant Save if a Nuke Plant Replaced Coal?

(a) Image 1
2550×3300 pixels, 300dpi

(b) Image 2
3193×4174 pixels, 400dpi

(c) Image 3
2336×3215 pixels, 300dpi

Fig. 1.8. 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.



(a)

(b)

(c)

Fig. 1.9. Segmentation maps of (a) Image 1, (b) Image 2, and (c) Image 3. White: background blocks; red: text blocks; blue: picture blocks.

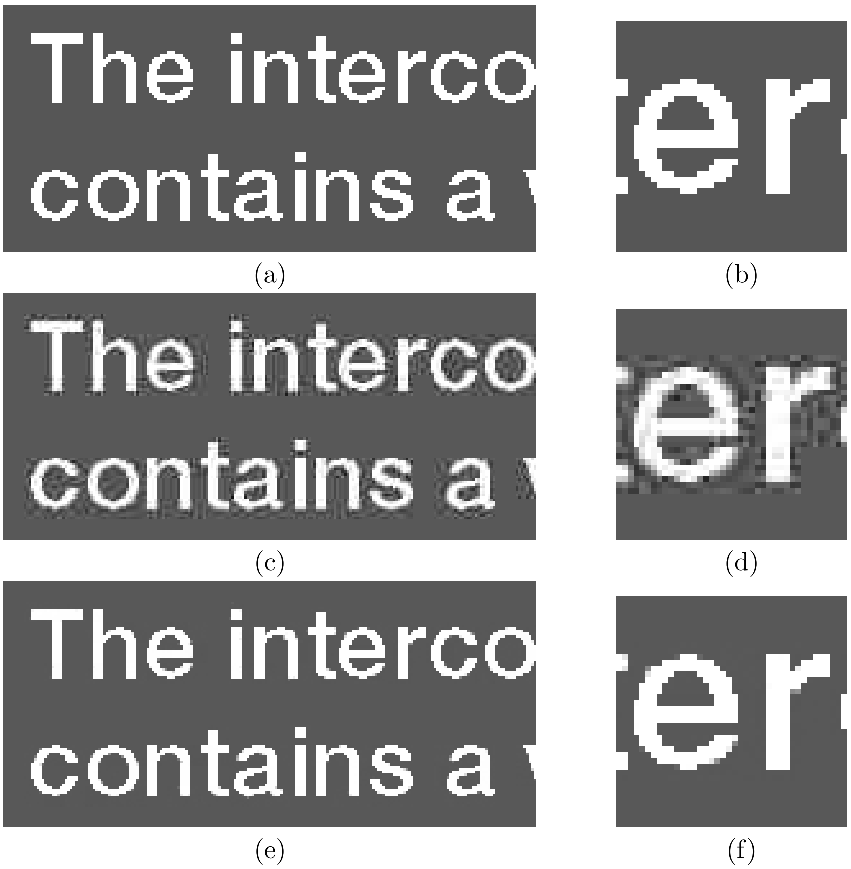


Fig. 1.10. Luminance component of a text region of Image 1. (a), (b) Original. (c), (d) Conventional JPEG decoding. (e), (f) The proposed scheme. (b), (d), and (f) are enlargements of a small region of (a), (c), and (e) respectively.

region contains obvious ringing artifacts around the text. Fig. 1.10(e) and (f) show the same region decoded by our scheme. Compared to Fig. 1.10(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. 1.11(a) shows the chrominance component C_r for the region in Fig. 1.10(a). The result decoded by the conventional JPEG decoder and interpolated by pixel replication is shown in Fig. 1.11(b). The decoded region is highly distorted due to chrominance subsampling. Fig. 1.11(c) shows the region decoded by the proposed

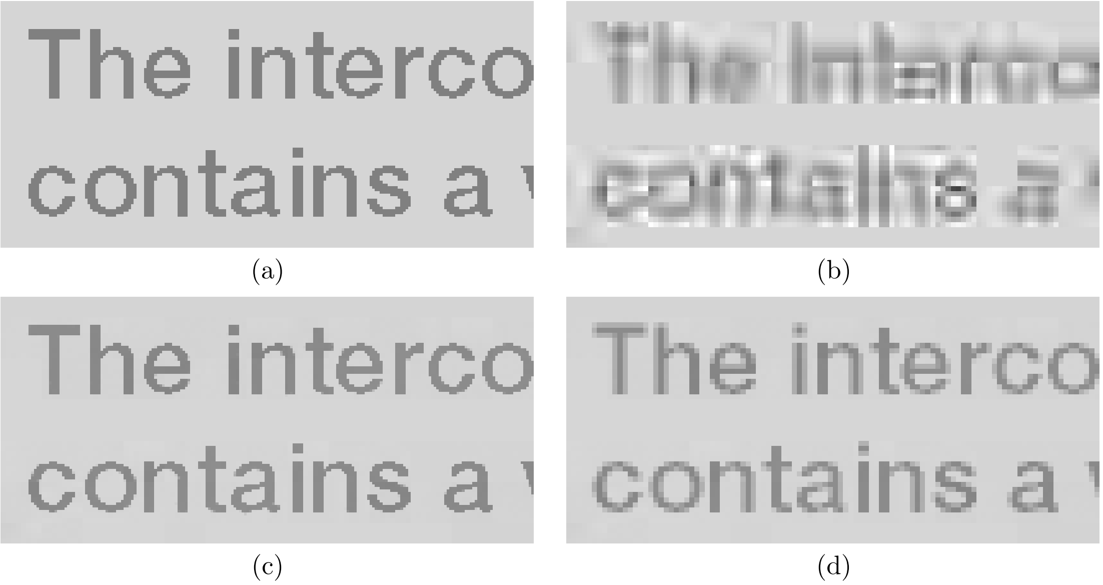


Fig. 1.11. Chrominance component (C_r) of the region shown in Fig. 1.10. (a) Original. (b) Decoded by conventional JPEG decoding and interpolated by pixel replication. (c) Decoded by our scheme. (d) Decoded by our scheme but interpolated by pixel replication.

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. 1.11(d) shows the result decoded by our scheme but interpolated by pixel replication. The text region decoded by our scheme in Fig. 1.11(c) is much clearer and sharper as compared to Fig. 1.11(d).

Fig. 1.12(c) shows the region completely decoded using our scheme. A comparison with the same region decoded by the conventional JPEG decoder in Fig. 1.12(b) reveals that the proposed algorithm significantly improves the quality of the decoded regions. Additional results for text regions in Figs. 1.13(c), 1.14(c), and 1.15(c) show 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

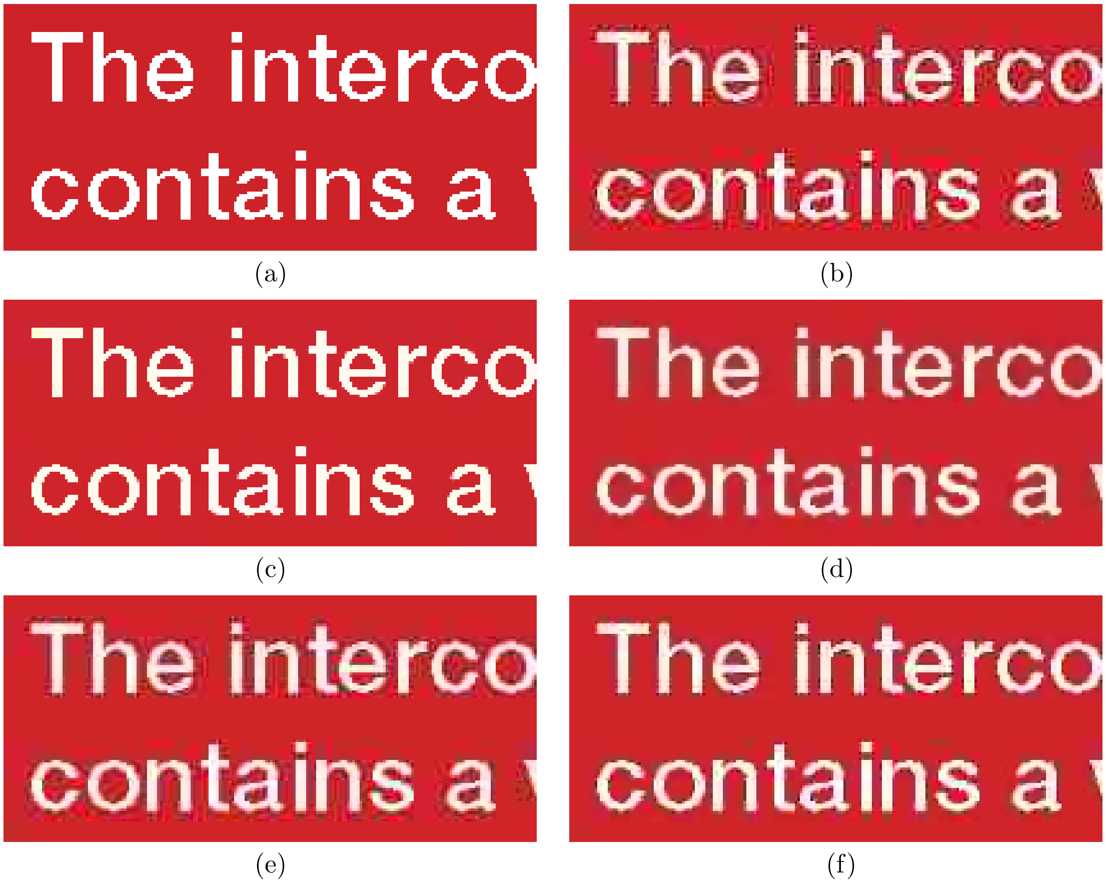
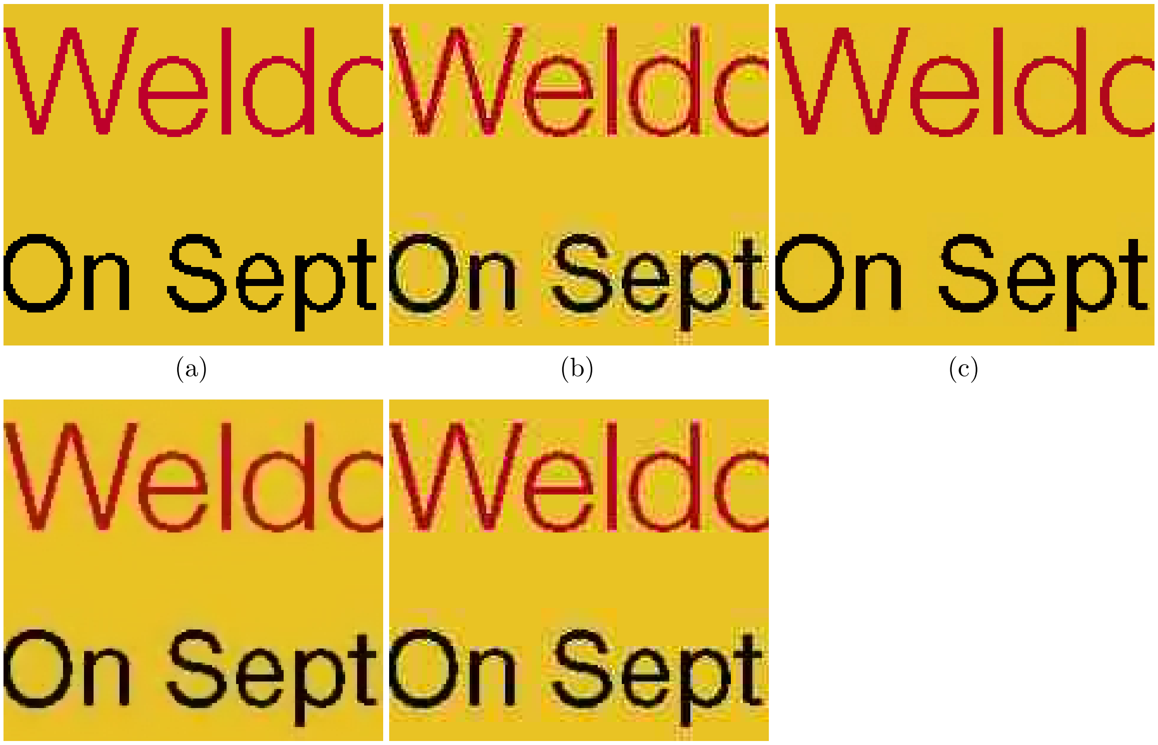


Fig. 1.12. A text region from Image 1. (a) Original. (b) Conventional JPEG decoding. (c) The proposed algorithm. (d) Algorithm I [11]. (e) Algorithm II [34]. (f) Algorithm III [3].

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. 1.12(d), 1.13(d), 1.14(d), and 1.15(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



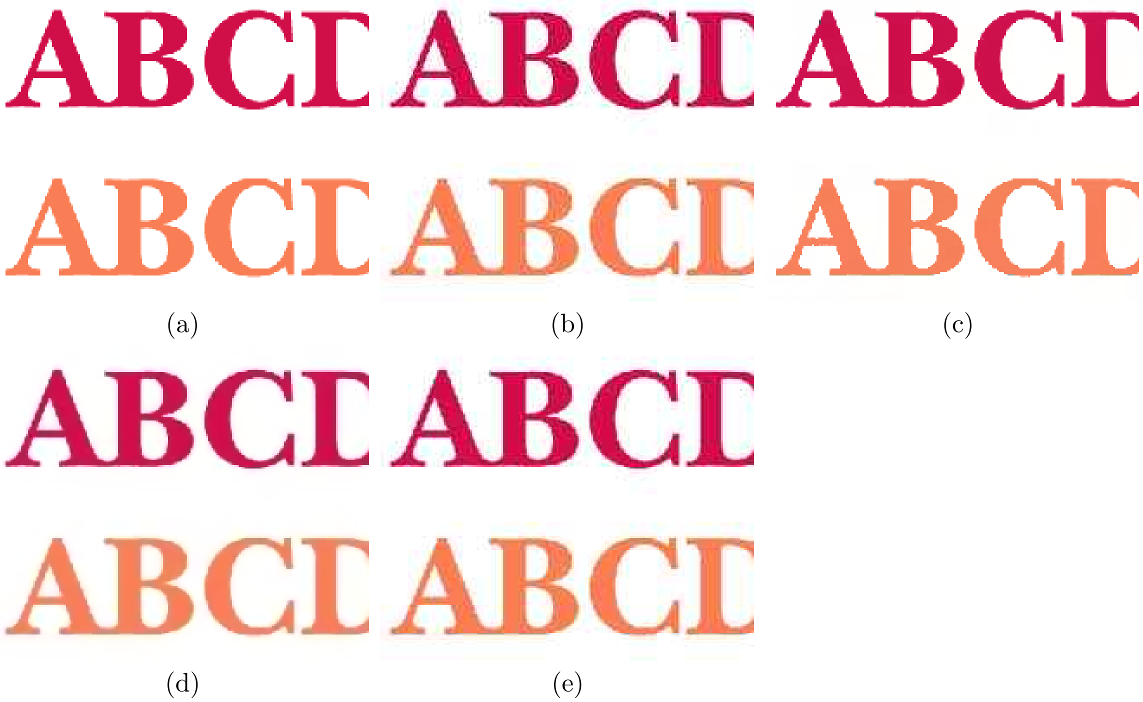
(a) (b) (c)

(d) (e)

Fig. 1.13. Another text region from Image 1. (a) Original. (b) Conventional JPEG decoding. (c) The proposed algorithm. (d) Algorithm I [11]. (e) Algorithm II [34].

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



(a) (b) (c)

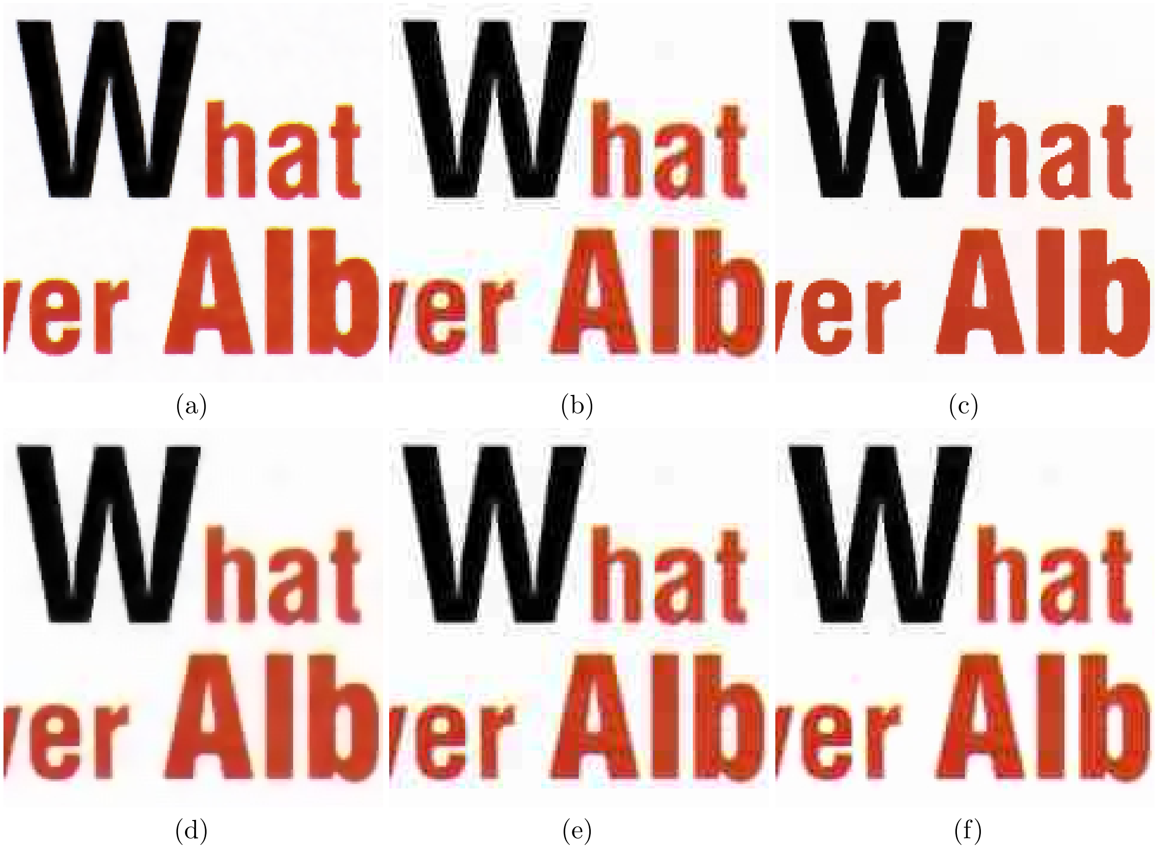
(d) (e)

Fig. 1.14. A text region from Image 2. (a) Original. (b) Conventional JPEG decoding. (c) The proposed algorithm. (d) Algorithm I [11]. (e) Algorithm II [34].

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. 1.12(e), 1.13(e), 1.14(e), and 1.15(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. 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



(a) (b) (c)

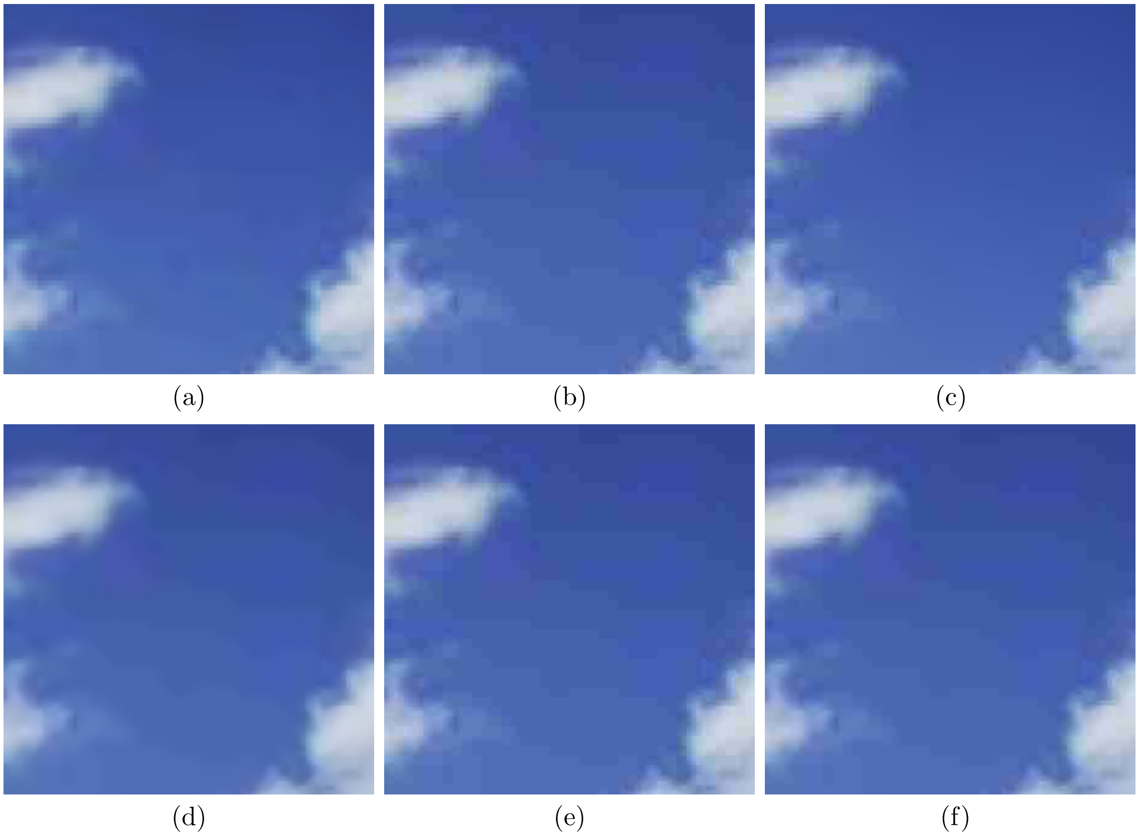
(d) (e) (f)

Fig. 1.15. A 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.

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 non-edge 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 pre-segmented 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. 1.12(a), and the result of decoding it with Algorithm III in Fig. 1.12(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. 1.15(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. 1.16 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. 1.16(b) shows the region decoded by the conventional JPEG decoder. The decoded region exhibits obvious contouring as a result of quantization. Algorithm I, Fig. 1.16(d), significantly reduces the blocking artifacts, but contouring in the blue sky is still apparent. Algorithm II uses the



(a) (b) (c)

(d) (e) (f)

Fig. 1.16. A 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].

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. 1.16(f). With our scheme, Fig. 1.16(c), contouring 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. 1.16(a), if

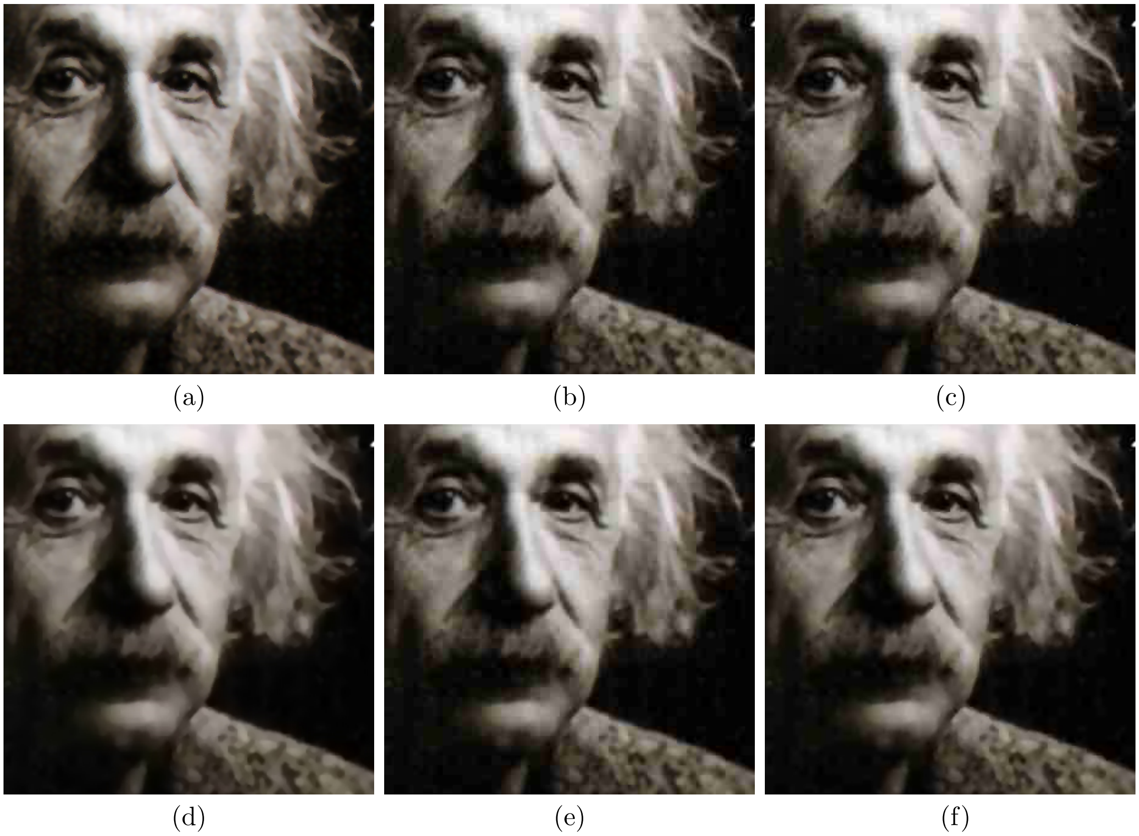


Fig. 1.17. A region from Image 3 containing mostly picture blocks. The image blocks corresponding to the face and shoulder are mostly labeled as picture blocks, and the remaining blocks are labeled as background blocks. (a) Original. (b) Conventional JPEG decoder. (c) The proposed algorithm. (d) Algorithm I [11]. (e) Algorithm II [34]. (f) Algorithm III [3].

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. 1.16 can accurately restore the artifacts.

Fig. 1.17 shows a region from Image 3 with most blocks classified as picture blocks. Among the five decoding schemes, Algorithm I in Fig. 1.17(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. 1.17(e) looks very similar to that decoded by the conventional JPEG decoder in Fig. 1.17(b). In Fig. 1.17(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. 1.17(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. 1.18 (a) and (b). In this case, the forward model [formulated in (1.2) and implemented through (1.25)–(1.27)] 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. 1.18 (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, Fig. 1.18 (c) and (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 4 images from the 60 test images. Each image is JPEG encoded at 4 different bit rates, resulting in a total of 16 encoded images. In each test, we vary one of the parameters in Table 1.1 (except $h_{r,s}$) over

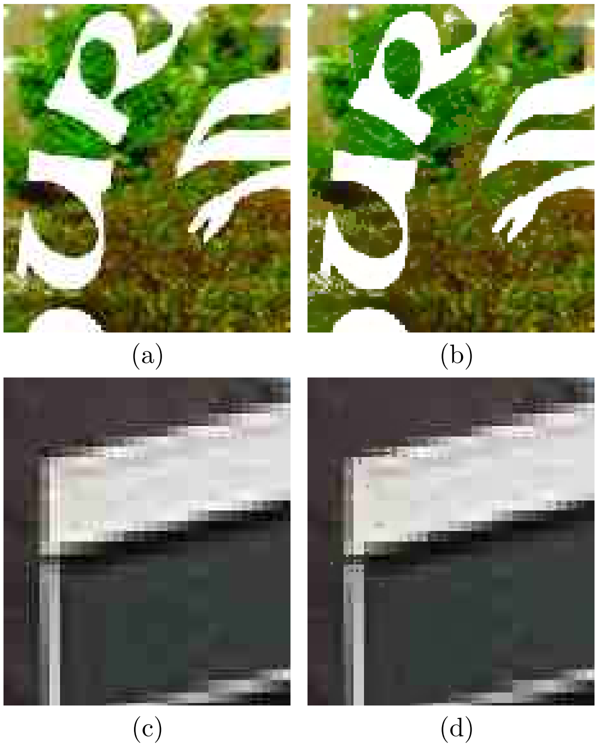
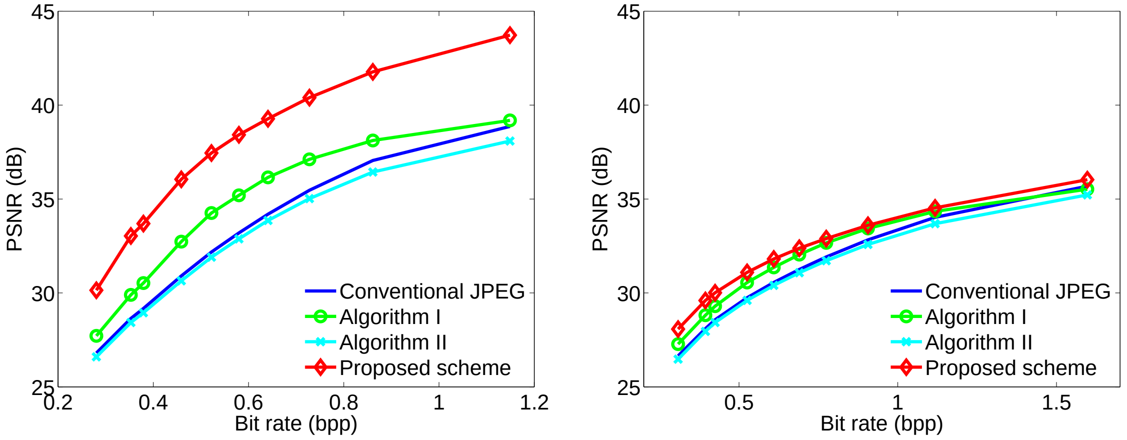


Fig. 1.18. Robustness of the proposed algorithm. (a), (b) Image patch where text blocks contain non-uniform background: (a) conventional JPEG decoder, (b) the proposed algorithm. (c), (d) Image patch where our segmentation algorithm misclassifies some of the picture blocks as text blocks: (c) conventional JPEG decoder, (d) the proposed algorithm.

Table 1.2
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



(a) (b)

Fig. 1.19. Average PSNR versus average bit rate computed for 30 digital images in (a), and another 30 scanned images in (b).

a ±10% interval and compute the average PSNR for the 16 decoded images. The maximum variation in the average PSNR, tabulated in Table 1.2, 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. 1.19 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, blocks, and picture.

1.8 Conclusions

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 1.7, 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.

2. IMAGE ENHANCEMENT USING THE HYPOTHESIS SELECTION FILTER: THEORY AND APPLICATION TO JPEG DECODING

2.1 Introduction

We propose a novel image enhancement approach, which we call Hypothesis Selection Filter (HSF). We illustrate HSF by applying it as a post-processing step to improve the decoding quality of JPEG-encoded document images.

HSF is inspired by the Resolution-Synthesis (RS) interpolation [35,36]. RS uses a feature vector to characterize the local edge structure. Classification results based on the feature vector are then used to combine the outputs of a number of linear filters to achieve optimal image interpolation. HSF follows the same processing structure as RS, but employs a different probabilistic model and design procedure. In RS, the design process starts with an unsupervised clustering of training feature vectors, followed by the estimation of the linear minimum mean squared error (MMSE) filter for each feature cluster. In HSF, the processing filters are specified *a priori*, followed by the characterization of feature vector for each filter. Therefore, the choices of filters to be used in the HSF are not limited to linear filters. This flexibility of filter selection is advantageous for certain image enhancement tasks.

Our algorithm processes an image using a filter bank and combines the filter outputs at every pixel into a convex combination, using spatially varying weights. The filter bank is constructed so that it contains filters capable of enhancing image regions of different types. For example, as detailed in Section 2.3, when we use HSF to enhance JPEG-decoded document images, we construct the filter bank so that it contains a smoothing filter capable of suppressing blocking artifacts in smooth picture regions, as well as edge enhancement filters which are capable of sharpening text.

For each pixel, the weights are estimated through the computation of a feature vector using a local window around the pixel. The purpose of the feature vector is to predict which filter(s) would be most accurate in enhancing the pixel. For example, in our JPEG application we include a feature to detect smooth regions where a smoothing filter would be most appropriate, as well as text-detection features to identify regions where text-enhancement filters would be most appropriate.

Once the feature vector identifies which filter is best to use for a particular pixel, it is tempting to simply select that filter's output as the enhanced pixel value. This approach, however, turns out to be suboptimal and not as effective as our approach which produces a weighted combination of filters' outputs. The mapping between the feature vector and the weights is learned from training data.

We apply HSF as a post-processing step to improve the decoding quality of JPEG-encoded document images. In our JPEG post-processing scheme described in Section 2.3, the HSF employs two nonlinear filters to estimate the local foreground gray-level and the local background gray-level. Our results show that these two filters are more effective than the linear filtering of RS for eliminating ringing artifacts around text regions. Experimental results show that our scheme improves the image quality more consistently over a variety of image contents with different characteristics, as compared to several other state-of-the-art JPEG decoding methods.

We introduce the overall architecture for HSF in Section 2.2. The underlying probabilistic model for pixels and feature vectors is then introduced in Section 2.2.1, followed by the description of the parameter estimation procedure in Section 2.2.2. We apply HSF to improving the quality of JPEG decoding in Sections 2.3 and 2.4: Section 2.3 discusses the filter bank and the features we use in HSF for this application; and Section 2.4 presents our experimental results and comparisons with several state of the art JPEG decoding methods.

2.2 Hypothesis Selection Filter

We use upper-case letters to denote random variables and random vectors, and lower-case letters to denote their realizations. We use the random variable X_n to represent the intensity of pixel n of the original image. We use $\tilde{x}_n$ to denote the n-th pixel intensity of the observed distorted version of the original. These distorted pixel values do not enter into our probabilistic model, and can therefore be assumed to be deterministic.

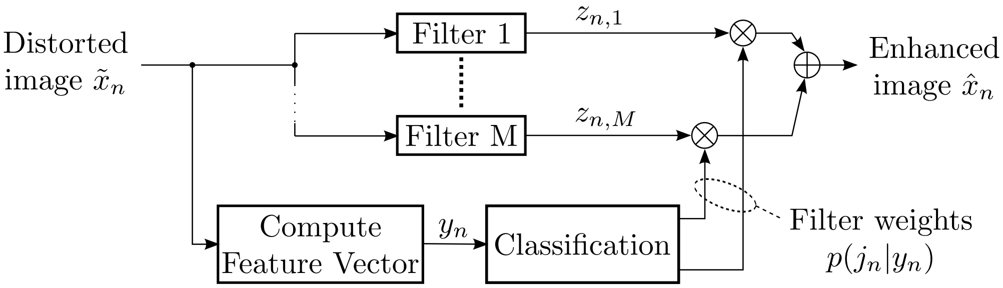

Distorted image $\tilde{x}_n$
Filter 1
Filter M
$z_{n,1}$
$z_{n,M}$
Enhanced image $\hat{x}_n$
Compute Feature Vector
y_n
Classification
Filter weights $p(j_n|y_n)$

Fig. 2.1. Hypothesis selection filter. The M filter outputs serve as different estimates of the original pixel value X_n. The feature vector $\mathbf{Y}_n$ captures the local characteristics around the pixel. The feature vector is then used to determine the optimal filter weights to form the processed output. For each pixel, the computation of the filter outputs and the feature vector may depend on a local window around the pixel. Computation of filter weights also depends on a number of model parameters which are determined through training.

The structure of HSF is shown in Fig. 2.1. In the upper portion of the diagram, HSF has a filter bank with M filters whose outputs at the n-th pixel are denoted by $\mathbf{z}_n = [z_{n,1}, \ldots, z_{n,M}]^t$. In our framework, it is not necessary to model these outputs as random variables, and therefore they are assumed to be deterministic.

In the lower portion of the diagram, feature extraction is first performed, resulting in a feature vector $\mathbf{Y}_n$ for each pixel n. Each feature vector is modeled as a random vector. The feature vector is then used to compute the filter weights, using a procedure described below in Section 2.2.1. The weights are then used to construct the enhanced pixel as a convex combination of the filter outputs.

2.2.1 Probabilistic Model

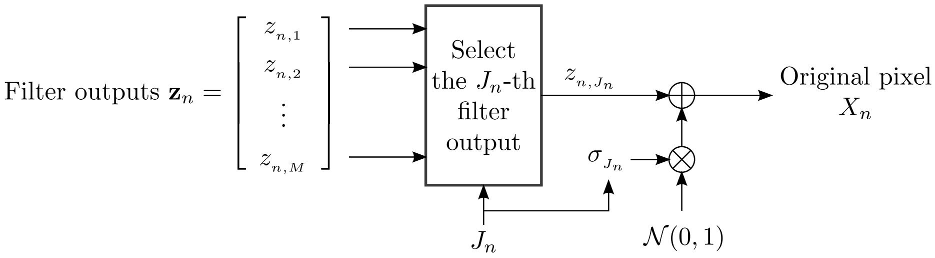


Fig. 2.2. Probabilistic model for the original pixel X_n. We introduce a hidden discrete random variable $J_n \in \{1, \ldots, M\}$, which is sampled with prior probability $\text{Prob}(J_n = j) = \pi_j$. Given J_n, we model X_n as a Gaussian random variable with mean z_{n,J_n} and variance $\sigma^2_{J_n}$.

Fig. 2.2 shows our probabilistic model for the original pixel X_n. We model the n-th pixel of the original image, X_n, as a mixture of M Gaussian random variables whose expectations are the filter outputs $z_{n,j}$:

$$f_{X_n}(x) = \sum_{j=1}^{M} \pi_j \mathcal{N}(x; z_{n,j}, \sigma_j^2), \tag{2.1}$$

where f_{X_n} is the pdf of X_n, $\mathcal{N}(\cdot; z_{n,j}, \sigma_j^2)$ is a Gaussian pdf with mean $z_{n,j}$ and variance σ_j^2, and π_j's are the mixture weights.[1] We define a hidden discrete random variable J_n such that

$$\text{Prob}(J_n = j) = \pi_j, \text{ for } j = 1, \ldots, M,$$

and such that the conditional pdf of X_n given $J_n = j$ is the j-th mixture component from (2.1):

$$f_{X_n|J_n}(x|j) = \mathcal{N}(x; z_{n,j}, \sigma_j^2). \tag{2.2}$$

[1] Since $z_{n,j}$ is the result of filtering a degraded version of $\mathbf{X}$, it may seem somewhat counterintuitive to model $\mathbf{X}$ as a random vector, and yet to model $z_{n,j}$ as deterministic. However, what we are modeling here is the process of reconstructing $\mathbf{X}$ from $z_{n,j}$'s.

We model the conditional distribution of $\mathbf{Y}_n$ given $J_n = j$ as a Gaussian mixture:

$$f_{\mathbf{Y}_n|J_n}(\mathbf{y}|j) = \sum_{l=1}^{K_j} \zeta_{j,l} \mathcal{N}(\mathbf{y}; \mathbf{m}_{j,l}, R_{j,l}), \tag{2.3}$$

where $\zeta_{j,l}$, $\mathbf{m}_{j,l}$, and $R_{j,l}$ are, respectively, the weight, the mean vector, and the covariance matrix of the l-th Gaussian component; and where K_j is the order of the Gaussian mixture.

We assume that X_n and $\mathbf{Y}_n$ are conditionally independent given J_n, which results in the following joint distribution of X_n, $\mathbf{Y}_n$, and J_n:

$$f_{X_n,\mathbf{Y}_n,J_n}(x, \mathbf{y}, j) = f_{X_n|J_n}(x|j) f_{\mathbf{Y}_n|J_n}(\mathbf{y}|j) \pi_j,$$

where the conditional distrbutions $f_{X_n|J_n}$ and $f_{\mathbf{Y}_n|J_n}$ are given by (2.2) and (2.3), respectively.

HSF computes the Bayesian minimum mean-square error estimate $\hat{X}_n$ of X_n based on observing the feature vector $\mathbf{Y}_n$. This estimate is the conditional expectation of X_n given $\mathbf{Y}_n$ [37]:

$$\begin{aligned}
\hat{X}_n &= E[X_n|\mathbf{Y}_n] && (2.4)\\
&= \sum_{j=1}^{M} E[X_n|\mathbf{Y}_n, J_n = j]\ f_{J_n|\mathbf{Y}_n}(j|\mathbf{Y}_n) && (2.5)\\
&= \sum_{j=1}^{M} E[X_n|J_n = j]\ f_{J_n|\mathbf{Y}_n}(j|\mathbf{Y}_n) && (2.6)\\
&= \sum_{j=1}^{M} z_{n,j}\ f_{J_n|\mathbf{Y}_n}(j|\mathbf{Y}_n) && (2.7)\\
&= \sum_{j=1}^{M} z_{n,j} \frac{f_{\mathbf{Y}_n|J_n}(\mathbf{Y}_n|j)\pi_j}{\sum_{j'=1}^{M} f_{\mathbf{Y}_n|J_n}(\mathbf{Y}_n|j')\pi_{j'}}, && (2.8)
\end{aligned}$$

where we have used the total expectation theorem to go from (2.4) to (2.5); our modeling assumption that the pixel value X_n and the feature vector $\mathbf{Y}_n$ are conditionally independent given J_n to go from (2.5) to (2.6); the fact that the conditional mean of X_n given $J_n = j$ is $z_{n,j}$ to go from (2.6) to (2.7); and Bayes' rule to go from (2.7) to (2.8). In order to construct this estimate, we need the following unknown parameters: the Gaussian mixture weights π_i from (2.1) and all the parameters of the Gaussian mixture model from (2.3). We replace these parameters with their maximum likelihood estimates obtained through the EM algorithm, as detailed in the next subsection.

2.2.2 Parameter Estimation

Our statistical model consists of three sets of parameters: $\theta = \{\pi_j, \sigma_j^2\}_j$ from (2.1); $\psi = \{\zeta_{j,l}, \mathbf{m}_{j,l}, R_{j,l}\}_{j,l}$ from (2.3); and the orders K_j of the Gaussian mixture distributions $f_{\mathbf{Y}_n|J_n}(\mathbf{y}|j)$, also from (2.3). We estimate the parameters by an unsupervised training procedure. This procedure makes use of a training set which consists of example values $\{(x_n, \mathbf{y}_n, \mathbf{z}_n)\}_n$. To describe the training procedure, we also need to define another hidden discrete random variable L_n as the component index in (2.3), such that

$$\text{Prob}(L_n = l | J_n = j) = \zeta_{j,l},$$

and

$$f_{\mathbf{Y}_n|J_n,L_n}(\mathbf{y}|j,l) = \mathcal{N}(\mathbf{y}; \mathbf{m}_{j,l}, R_{j,l}), \tag{2.9}$$

for $l = 1, \ldots, K_j$ and $j = 1, \ldots, M$.

We seek the maximum likelihood (ML) estimates of θ and ψ. Because the random variables J_n and L_n are unobserved, we solve this so-called incomplete data problem using the expectation-maximization (EM) algorithm [38]. In deriving the training procedure, we make the assumption that J_n is conditionally independent of $\mathbf{Y}_n$ given X_n.

$$f_{J_n|X_n,\mathbf{Y}_n}(j|x_n, \mathbf{y}_n) = f_{J_n|X_n}(j|x_n). \tag{2.10}$$

We make this assumption so as to decompose the EM algorithm into two simpler sub-problems. In the first sub-problem, we compute the ML estimate of θ by applying the following set of update iterations:

$$\begin{aligned}
\hat{f}_{J_n|X_n}(j|x_n) &\leftarrow \frac{\hat{\pi}_j \, \text{Normal}(x_n; z_{n,j}, \hat{\sigma}_j^2)}{\sum_{j'} \hat{\pi}_{j'} \, \text{Normal}(x_n; z_{n,j'}, \hat{\sigma}_{j'}^2)}, && (2.11)\\
\hat{N}_j &\leftarrow \textstyle\sum_n \hat{f}_{J_n|X_n}(j|x_n), && (2.12)\\
\hat{t}_j &\leftarrow \textstyle\sum_n (x_n - z_{n,j})^2 \; \hat{f}_{J_n|X_n}(j|x_n), && (2.13)\\
(\hat{\pi}_j, \; \hat{\sigma}_j^2) &\leftarrow \left(\frac{\hat{N}_j}{N}, \; \frac{\hat{t}_j}{\hat{N}_j} \right), && (2.14)
\end{aligned}$$

where N is the total number of training samples. After the convergence of the first set of update iterations, we use the ML estimate $\hat{\theta}$ in the second sub-problem to compute the ML estimate of ψ. For each $j = 1, \ldots, M$, we compute the ML estimates of $\{\zeta_{j,l}, \mathbf{m}_{j,l}, R_{j,l}\}_l$ by applying the following set of update iterations:

$$\begin{aligned}
\hat{f}_{J_n,L_n|X_n,\mathbf{Y}_n}(j,l|x_n,\mathbf{y}_n) &\leftarrow \hat{f}_{J_n|X_n}(j|x_n) \frac{\hat{\zeta}_{j,l} \, \text{Normal}(\mathbf{y}_n; \hat{\mathbf{m}}_{j,l}, \hat{R}_{j,l})}{\sum_{l'} \hat{\zeta}_{j,l'} \, \text{Normal}(\mathbf{y}_n; \hat{\mathbf{m}}_{j,l'}, \hat{R}_{j,l'})}, && (2.15)\\
\hat{N}_{j,l} &\leftarrow \textstyle\sum_n \hat{f}_{J_n,L_n|X_n,\mathbf{Y}_n}(j,l|x_n,\mathbf{y}_n), && (2.16)\\
\hat{\mathbf{u}}_{j,l} &\leftarrow \textstyle\sum_n \mathbf{y}_n \; \hat{f}_{J_n,L_n|X_n,\mathbf{Y}_n}(j,l|x_n,\mathbf{y}_n), && (2.17)\\
\hat{v}_{j,l} &\leftarrow \textstyle\sum_n \mathbf{y}_n \mathbf{y}_n^t \; \hat{f}_{J_n,L_n|X_n,\mathbf{Y}_n}(j,l|x_n,\mathbf{y}_n), && (2.18)\\
(\hat{\zeta}_{j,l}, \; \hat{\mathbf{m}}_{j,l}, \; \hat{R}_{j,l}) &\leftarrow \left(\frac{\hat{N}_{j,l}}{N}, \; \frac{\hat{\mathbf{u}}_{j,l}}{\hat{N}_{j,l}}, \; \frac{\hat{v}_{j,l}}{\hat{N}_{j,l}} - \frac{\hat{\mathbf{u}}_{j,l} \hat{\mathbf{u}}_{j,l}^t}{\hat{N}_{j,l}^2} \right). && (2.19)
\end{aligned}$$

We apply the minimum description length (MDL) principle [39] to estimate the order K_j of the Gaussian mixture distribution of (2.3), for each $j = 1, \ldots, M$. The MDL principle estimates the optimal model order by minimizing the number of bits that would be required to code both the training samples and the model parameters $\psi_j == \{\zeta_{j,l}, \mathbf{m}_{j,l}, R_{j,l}\}_l$. More specifically, we start with an initial estimate $K_j = K = 30$ for the model order and successively reduce the value of K_j. For each value of K_j, we estimate ψ_j by (2.15) – (2.19) and compute the value of a cost function which approximates the encoding length for the training samples and the model parameters. The optimal model order is then determined as the one which minimizes the encoding length. Further details of this method can be found from [32].

The derivation of the training procedure is explained in detail in Appendix A.

2.3 HSF for JPEG Decoding

In the previous section, we presented the HSF in a general context for image enhancement. This method can be applied to any linear/nonlinear filtering problem. To apply the HSF to a particular application, we only need to specify the set of desired image filters, and a feature vector that is well suited for selecting among the filters. Once the filters and the feature vector have been chosen, then our training procedure can be used to estimate the parameters of the HSF from training data.

In order to illustrate how the HSF can be utilized for image enhancement, in this section, we present an application of the HSF for improving the decoding quality for any JPEG encoded document images. JPEG compression [1] typically leads to the ringing and blocking artifacts in the decoded image. For a document image that contains text, graphics, and natural images, the different types of image content will be affected differently by the JPEG artifacts. For example, text and graphics regions usually contain many of sharp edges that lead to severe ringing artifacts in these regions, while natural images are usually more susceptible to the blocking artifacts. To improve the quality of the decoded document image, we first decode the image using a conventional JPEG decoder, and then post-process it using an HSF to reduce the JPEG artifacts. We describe the specific filters and the feature vector we use to construct the HSF in Section 2.3.1 and Section 2.3.2, respectively. Then we briefly describe our training procedure in Section 2.3.3. For simplicity, we design the HSF for monochrome images. Consequently, we also perform the training procedure using a set of monochrome training images. To process a color image, we apply the trained HSF to the R, G, and B color components of the image separately.

2.3.1 Filter Selection

Document images generally contain both text and pictures. Text regions contain many sharp edges and suffer significantly from ringing artifacts. Pictorial regions suffer from both blocking artifacts (in smooth parts) and ringing artifacts (around sharp edges). Our HSF makes use of four filters to eliminate these artifacts.

The first filter in the HSF is a bilateral filter [40,41] with geometric spread $\sigma_d = 0.5$ and photometric spread $\sigma_r = 10$. The bilateral filter is used to remove some of the blocking and ringing artifacts without blurring edges and image details.

The second filter is a Gaussian filter with kernel standard derivation $\sigma = 1$. We choose σ large enough so that the Gaussian filter can effectively eliminate both blocking and ringing artifacts in the pictorial regions, even for images encoded at a high compression ratio. Applied alone, however, the Gaussian filter will lead to significant blurring of edges and textures.

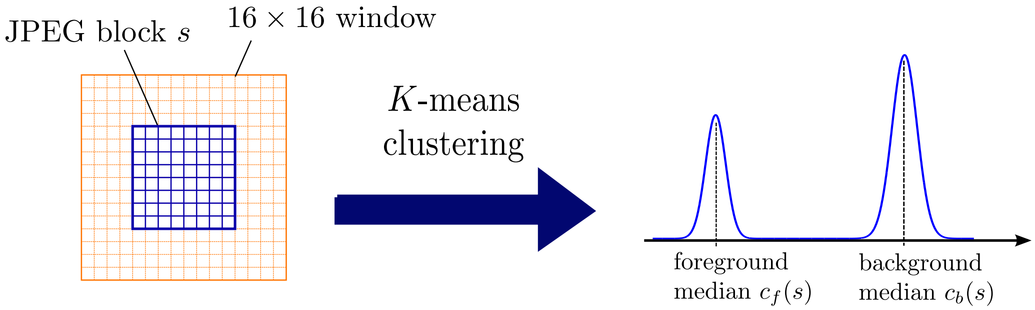


Fig. 2.3. Local foreground/background gray-level estimation. For each JPEG block, we center a 16×16 window around the block and partition the 256 pixels of the window into two groups by the K-means clustering algorithm. Then, we use the two cluster medians as the local foreground gray-level estimate and the local background gray-level estimate for the block.

Text pixels usually concentrate around either the foreground gray-level or the background gray-level of the text region. In the HSF, the third filter estimates the local foreground gray-level, and the fourth filter estimates the local background gray-level. The computation is illustrated in Fig. 2.3. For each 8×8 JPEG block s, we center a 16×16 window around the block, and apply the K-means clustering algorithm to partition the 256 pixels of the window into two groups. Then, we use the two cluster medians, $c_f(s)$ and $c_b(s)$, as the local foreground gray-level estimate and the local background gray-level estimate of the JPEG block.

2.3.2 Feature Vector

We use a five-dimensional feature vector to separate out the different types of image content which are most suitably processed by each of the filters employed in the HSF.

The first feature component is the variance of the JPEG block associated with the current pixel. The block variance is used to detect the smooth regions in the image.

The second feature component is used to evaluate how well the JPEG block associated with the current pixel can be approximated by using only the local foreground gray-level and the local background gray-level. For the JPEG block s, suppose $\{u_{s,i} : i = 0, \ldots, 63\}$ are the gray-levels of the 64 pixels in the block. We define the two-level normalized variance as

$$b_s = \frac{\sum_i \min\left(|u_{s,i} - c_b(s)|^2, |u_{s,i} - c_f(s)|^2\right)}{64\ |c_b(s) - c_f(s)|^2}. \tag{2.20}$$

We compute the two-level normalized variance of the JPEG block associated with the current pixel as the second feature component of the pixel.

For the third feature component, we compute the local gradient magnitude at the current pixel using a pair of Sobel operators [21] to detect major edges.

The fourth feature component is the difference between the pixel's gray-level and the local foreground gray-level $c_f(s)$; the fifth component is the difference between the pixel's gray-level and the local background gray-level $c_b(s)$:

$$y_{n,4} = u_n - c_f(s), \tag{2.21}$$

$$y_{n,5} = u_n - c_b(s), \tag{2.22}$$

where u_n is the gray-level of the current pixel, and s is the JPEG block associated with the pixel. These two features help to avoid applying the outputs of the last two filters when their errors are large.

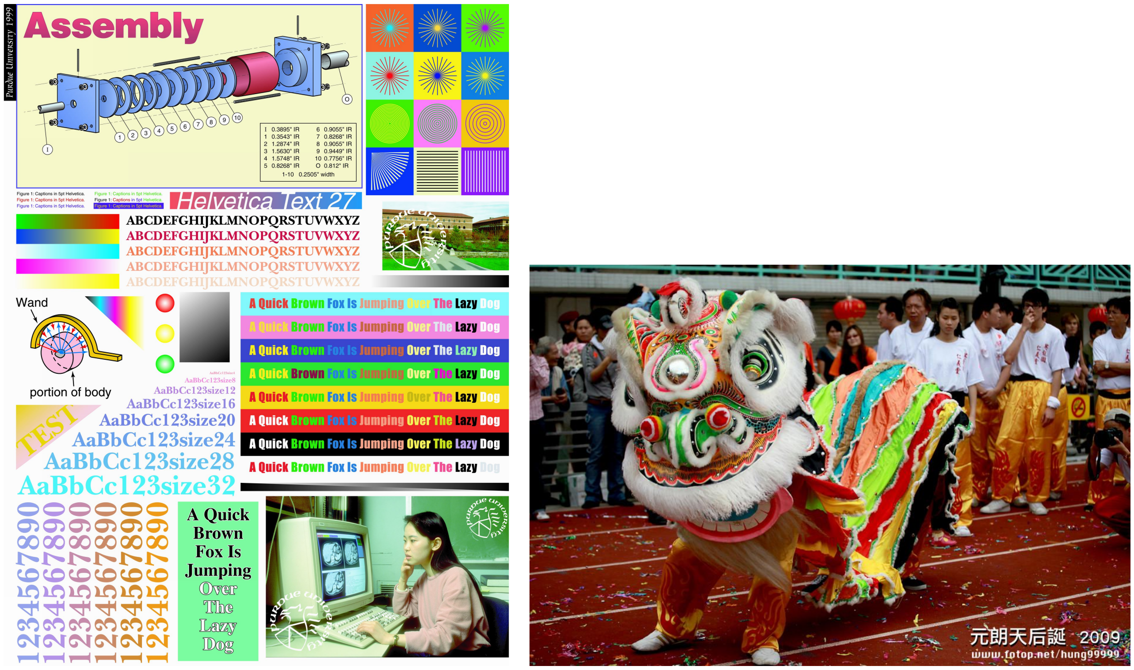
2.3.3 Training

To perform training for the HSF, we collected a set of document images and natural images to provide training samples of different types of image content. The training images were then JPEG encoded at different compression ratios to create the degraded images. We then applied our unsupervised training procedure described in Section 2.2.2 to the pairs of original and JPEG encoded images. For simplicity, we only used monochrome images to perform training. To process a color image, we apply the trained HSF to the different color components of the image separately. In our experiments presented in Section 2.4, none of the test images were used for training.

2.4 Results and Comparison

We compare our results with (i) the bilateral filter (BF) [40, 41] with parameters $\sigma_d = 0.5$ and $\sigma_r = 10$; (ii) a segmentation-based Bayesian reconstruction scheme (BR) [42]; (iii) Resolution Synthesis (RS) [35]; and (iv) Improved Resolution Synthesis (IRS) [36]. The bilateral filter BF is the same bilateral filter we use in the HSF. BR first segments the JPEG blocks into 3 classes corresponding to text, picture, and smooth regions. For text and smooth blocks, BR applies specific prior models to decode the image blocks for high quality decoding results. For picture blocks, BR simply uses the conventional JPEG decoder to decode the blocks. RS is a classification based image interpolation scheme which achieves optimal interpolation by training from image pairs at low resolution and high resolution. We adapt RS to JPEG artifact removal by setting the scaling factor to 1 and by using JPEG encoded images to perform training. IRS is an improved version of RS, designed for image interpolation and JPEG artifact removal. IRS employs a median filter at the output of RS to remove the remaining ringing artifacts left over by RS. Similar to RS, we set the scaling factor of IRS to 1, and use JPEG encoded images to perform training. We should also point out that all the schemes, except BR, are post-processing approaches. BR, on the other hand, is a Bayesian reconstruction approach that requires the access to the JPEG encoded DCT coefficients. Fig. 2.4 shows the thumbnails of two test images we use for comparing the visual quality of the different schemes.

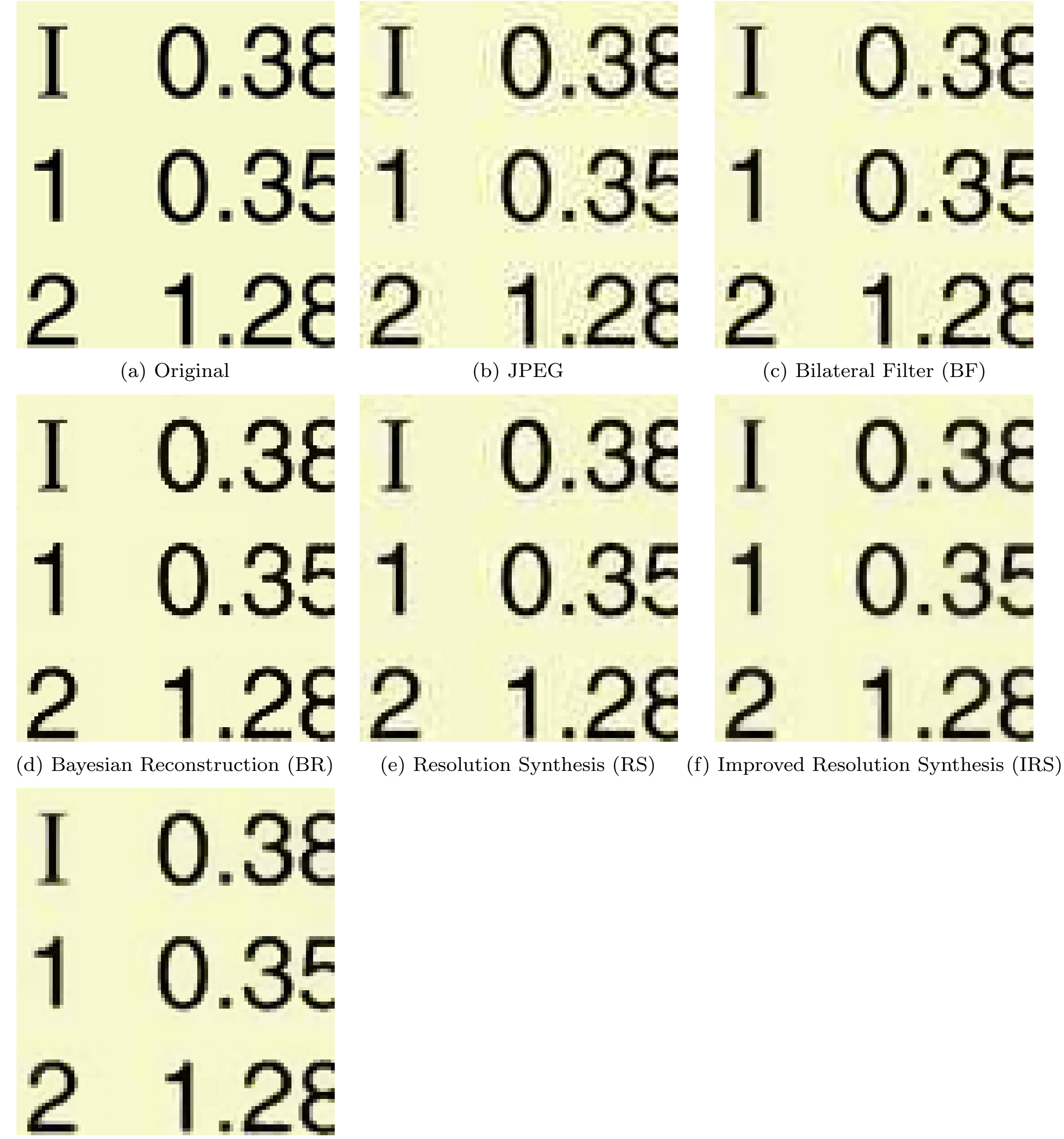
Fig. 2.5 and Fig. 2.6 compare the results for two different text regions from Image I. Because the text regions contain many high-contrast sharp edges, the regions decoded by JPEG (Fig. 2.5(b) and Fig. 2.6(b)) are severely distorted by ringing artifacts. BF (Fig. 2.5(c) and Fig. 2.6(c)) slightly reduces some of the ringing artifacts around the text. BR (Fig. 2.5(d) and Fig. 2.6(d)) uses a specific prior model to decode the text and effectively removes most of the ringing artifacts. The results of RS (Fig. 2.5(e) and Fig. 2.6(e)) are slightly better than the results of BF, but they still contain a fair amount of ringing artifacts. However, most of the ringing artifacts remained in the results of RS are successfully removed by the median filter employed by IRS (Fig. 2.5(f) and Fig. 2.6(f)). The results of HSF (Fig. 2.5(g) and Fig. 2.6(g)) contain fewer ringing artifacts than the results of BF, RS, and IRS, and the results are comparable to those of BR. For text region 1, the output of BR in Fig. 2.5(d) has some isolated dark pixels in the background regions whereas the output of HSF in Fig. 2.5(g) does not. On the other hand, the output of BR is sharper than the output of HSF.



(a) Image I

(b) Image II

Fig. 2.4. Thumbnails of test images used for visual quality comparison. Image I was JPEG encoded at 0.69 bits per pixel (bpp). Image II was JPEG encoded at 0.96 bpp.



(a) Original

(b) JPEG

(c) Bilateral Filter (BF)

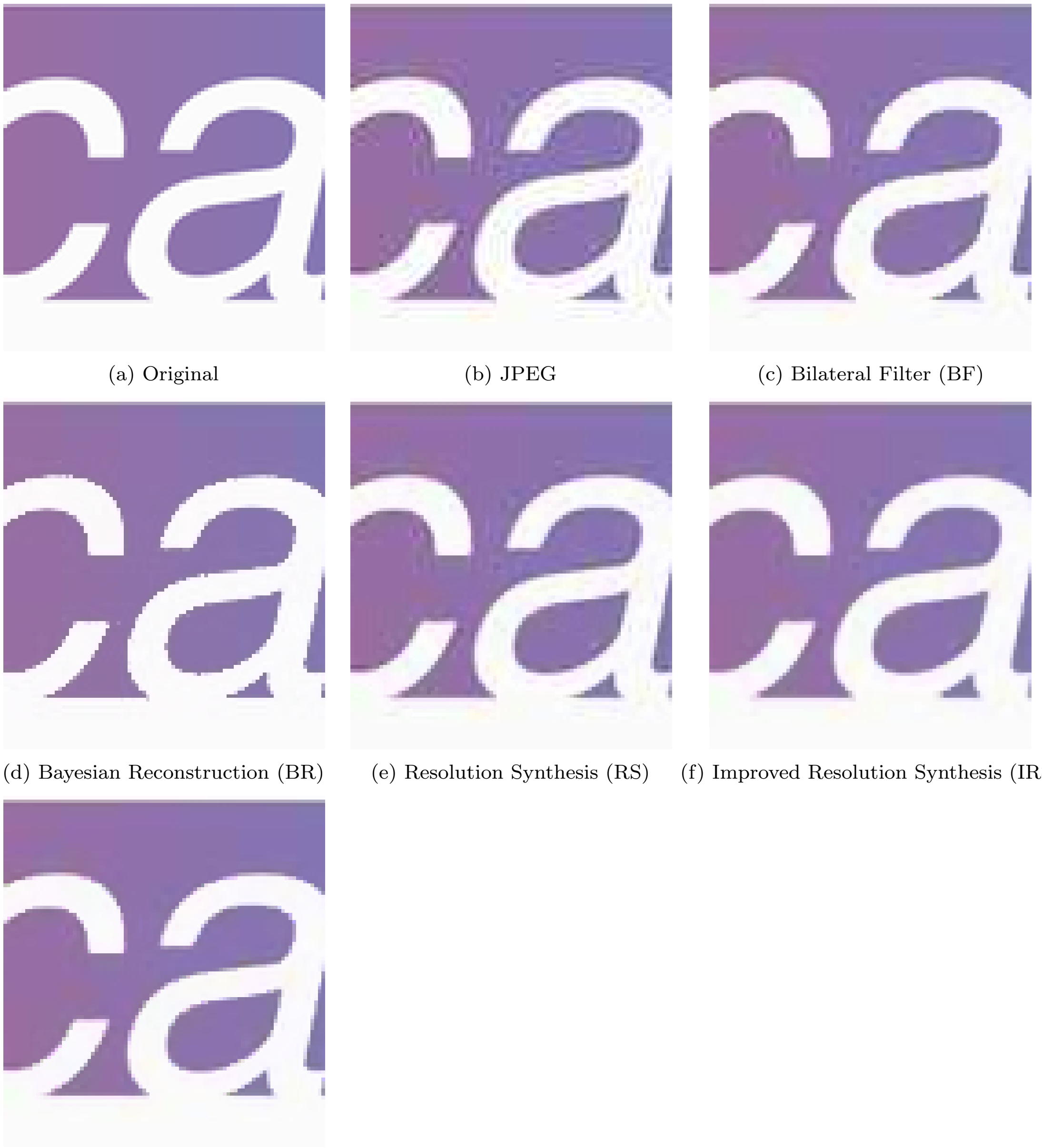
(d) Bayesian Reconstruction (BR)

(e) Resolution Synthesis (RS)

(f) Improved Resolution Synthesis (IRS)

(g) Hypothesis Selection Filter (HSF)

Fig. 2.5. Comparison of decoding results for text region 1 from Image I.



(a) Original

(b) JPEG

(c) Bilateral Filter (BF)

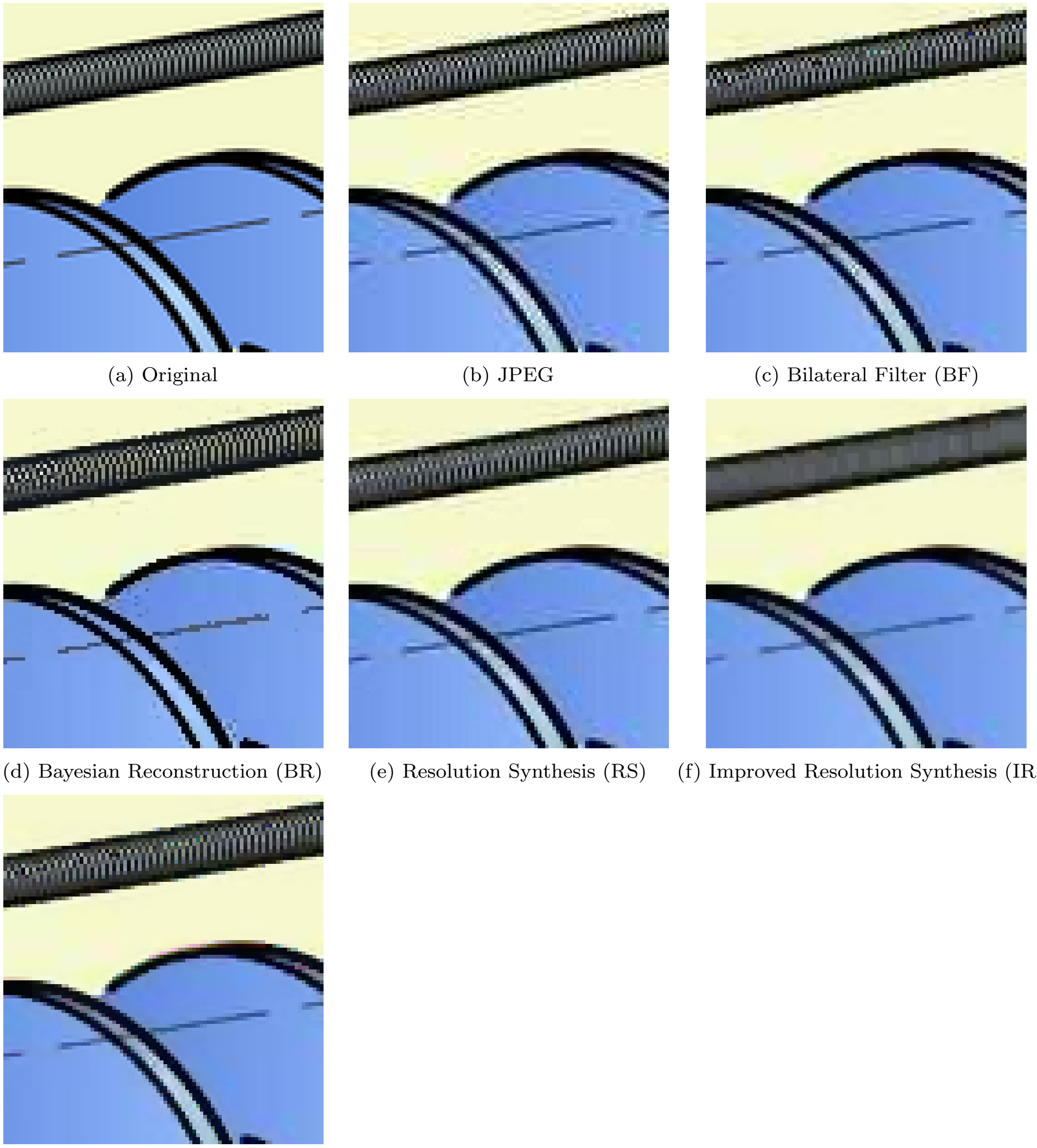
(d) Bayesian Reconstruction (BR)

(e) Resolution Synthesis (RS)

(f) Improved Resolution Synthesis (IRS)

(g) Hypothesis Selection Filter (HSF)

Fig. 2.6. Comparison of decoding results for text region 2 from Image II.



(a) Original

(b) JPEG

(c) Bilateral Filter (BF)

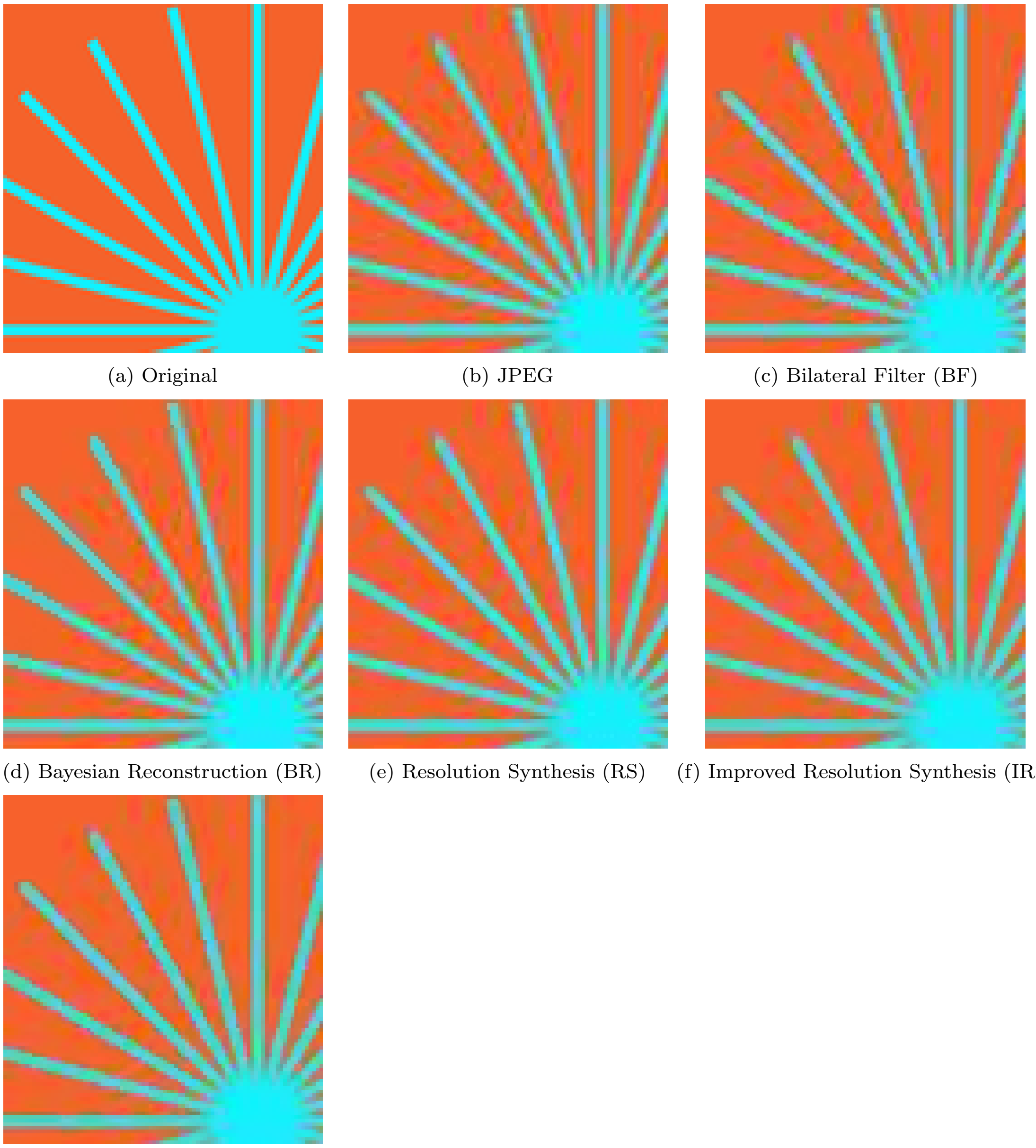
(d) Bayesian Reconstruction (BR)

(e) Resolution Synthesis (RS)

(f) Improved Resolution Synthesis (IRS)

(g) Hypothesis Selection Filter (HSF)

Fig. 2.7. Comparison of decoding results for graphic region 1 from Image I.



(a) Original

(b) JPEG

(c) Bilateral Filter (BF)

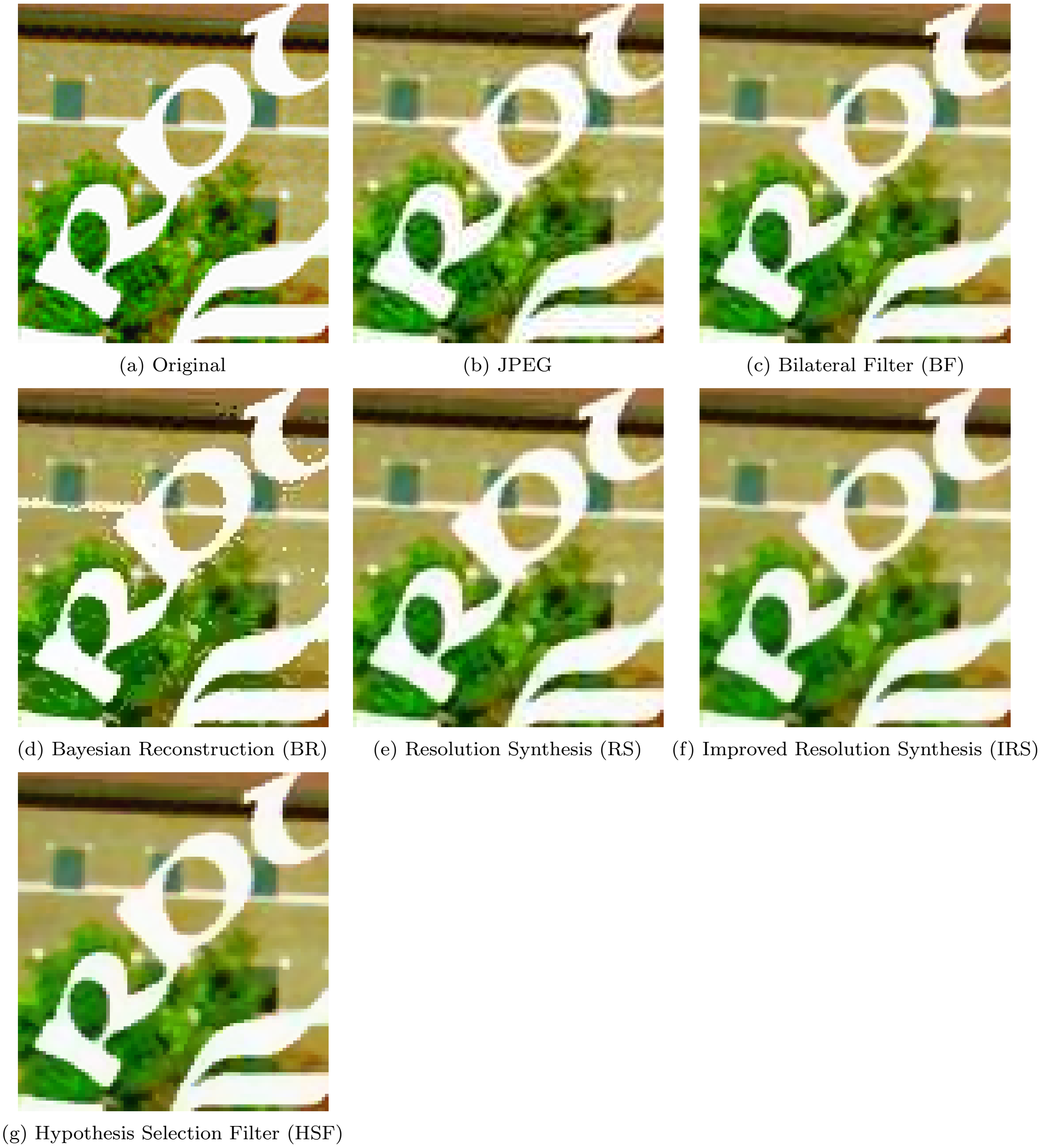
(d) Bayesian Reconstruction (BR)

(e) Resolution Synthesis (RS)

(f) Improved Resolution Synthesis (IRS)

(g) Hypothesis Selection Filter (HSF)

Fig. 2.8. Comparison of decoding results for graphic region 2 from Image I.



(a) Original

(b) JPEG

(c) Bilateral Filter (BF)

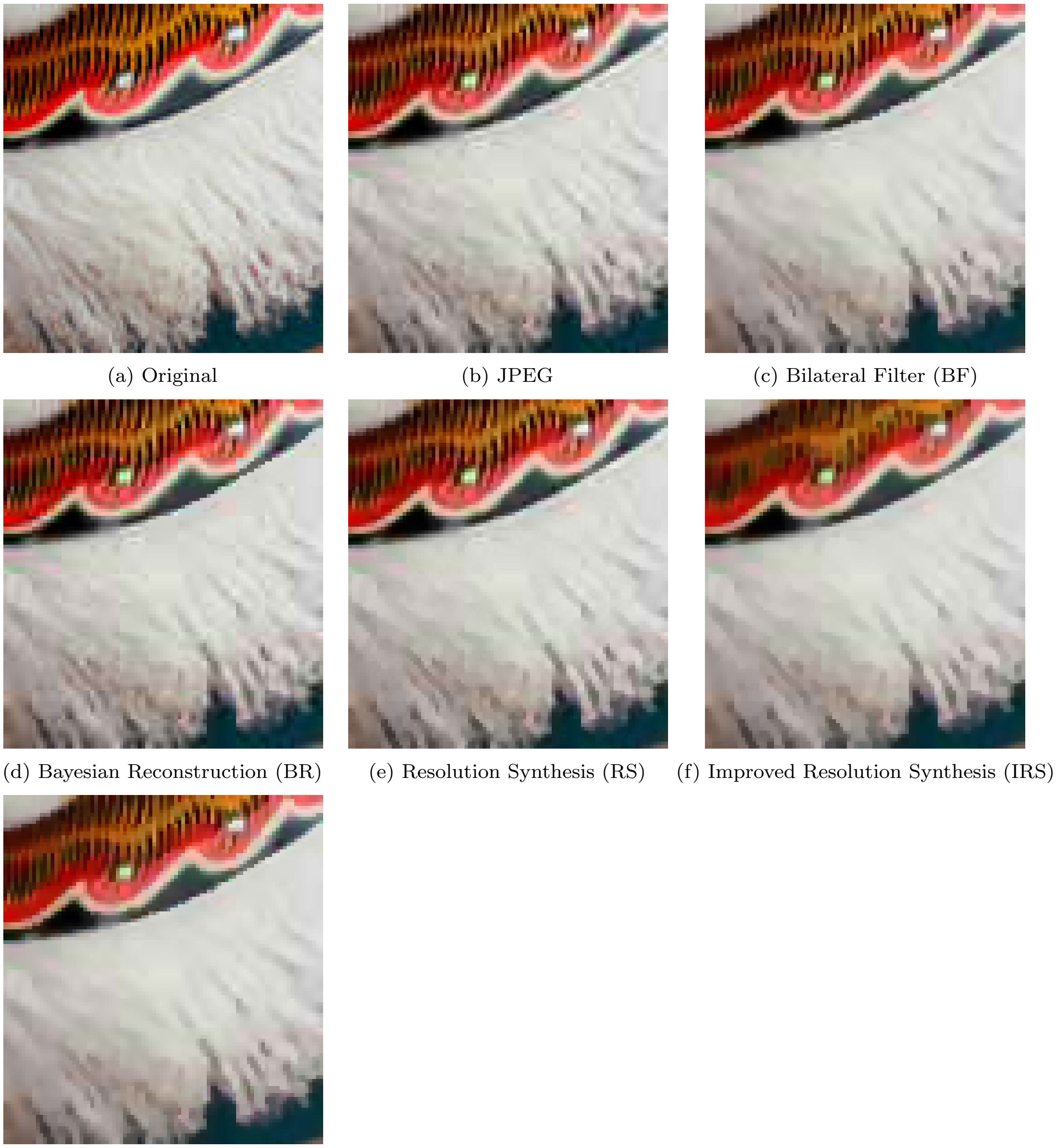
(d) Bayesian Reconstruction (BR)

(e) Resolution Synthesis (RS)

(f) Improved Resolution Synthesis (IRS)

(g) Hypothesis Selection Filter (HSF)

Fig. 2.9. Comparison of decoding results for mixed region 1 (text and natural image) from Image I.



(a) Original

(b) JPEG

(c) Bilateral Filter (BF)

(d) Bayesian Reconstruction (BR)

(e) Resolution Synthesis (RS)

(f) Improved Resolution Synthesis (IRS)

(g) Hypothesis Selection Filter (HSF)

Fig. 2.10. Comparison of post-processed results for pictorial region 2 from Image II.

Fig. 2.7 and Fig. 2.8 compare the results for two different graphics regions from Image I. Similar to text, graphic art contains sharp edges and leads to severe ringing artifacts in the JPEG decoded images (Fig. 2.7(b) and Fig. 2.8(b)). Both IRS (Fig. 2.7(f) and Fig. 2.8(f)) and HSF (Fig. 2.7(g) and Fig. 2.8(g)) are better than the other methods at removing the ringing artifacts in the two graphics regions. However, the median filter employed by IRS obliterates the fine texture in the upper half of graphics region 1 (Fig. 2.7(f)), whereas HSF preserves the texture well (Fig. 2.7(g)). BR is similarly effective in removing the ringing artifacts in graphics region 1 (Fig. 2.7(d)); however, just as in text region 1, it produces many isolated dark pixels inside light areas—a behavior typical for BR around high-contrast edges. Note that BR performs very poorly in graphics region 2 (Fig. 2.8(d)). This is due to the fact that BR misclassifies many blocks in this region as picture blocks and proceeds to simply use JPEG for these blocks.

Fig. 2.9 compares the results for a region with mixed content from Image I. The region contains a natural image overlaid with text. In the JPEG decoded image of Fig. 2.9(b), the text foreground is corrupted by ringing artifacts and the background image contains blocking artifacts. For this region, the text model of BR is unable to properly model the non-uniform background. This leads to artifacts in the form of salt-and-pepper noise around the text in Fig. 2.9(d). The decoded region also contains a fair number of blocking artifacts in the image background, for example, around the building roof near the top of the region. In terms of reducing the ringing and blocking artifacts, the outputs of IRS in Fig. 2.9(f) and HSF in Fig. 2.9(g) are significantly better than BR and slightly better than BF (Fig. 2.9(c)) and RS (Fig. 2.9(e)).

Fig. 2.10 compares the results for a picture region from Image II. This region is mainly corrupted by the blocking artifacts of JPEG (Fig. 2.10(b)). For BR, many JPEG blocks are classified as picture blocks by the segmentation algorithm. These JPEG blocks are simply decoded by the conventional JPEG algorithm, which results in many blocking artifacts. In the results of BF, RS, IRS, and HSF, blocking artifacts are mostly eliminated, especially around the smooth areas. However, similar to the graphics regions, the IRS results in very severe blurring of image details. In addition, BF and HSF produce sharper edges than both RS and IRS.

For the test images processed by the different schemes, we compute their peak signal-to-noise ratios (PSNR) and summarize the results in Table 2.1. Since the quality of a natural image is heavily affected by blocking artifacts, for Image II, we also measure the amount of blockiness in the processed images using the scheme proposed in [43]. The blockiness measure scheme assumes that the blocky image is the result of corrupting the original image by a blocking signal. The blockiness measure is computed by estimating the power of the blocking signal using discrete Fourier transform. The blockiness measures for Image II processed by the different schemes are summarized in Table 2.2. A larger blockiness measure suggests that the image contains more blocking artifacts.

For Image I, BR achieves the second highest PSNR whereas for image II BR's PSNR is the lowest. In addition, BR has a significantly worse blockiness measure for Image II than the other five methods. On the other hand, JPEG has the lowest PSNR for Image I, the highest PSNR for Image II, and the second-worst blockiness measure for Image II. Our proposed HSF algorithm has the best PSNR for Image I and the third-best PSNR for Image II, as well as the second best blockiness measure for Image II.

Image I contains a significant amount of text and graphics. The text model employed by BR is very effective in reducing the ringing artifacts in the text regions of Image I, which explains BR's good performance on this image. JPEG, originally designed for natural images, does not do well with text and graphics. However, the results from Fig. 2.5 – Fig. 2.8 and the PSNR comparison for Image I in Table 2.1, show that the performance of HSF in reducing the ringing artifacts in text and graphics is at least compariable to, or even superior than, BR.

Image II, on the other hand, is mostly a natural image. BR's classifier misclassifies a number of its blocks as text, resulting in a worse PSNR than JPEG, and a worse blockiness measure than JPEG. Image II illustrates the well-known fact that measuring image quality or fidelity using PSNR has shortcomings: even though BF, RS, IRS, and HSF all produce results that are considerably less blocky than both JPEG and BR—as seen, for example, in Fig. 2.10 and evidenced by Table 2.2—they all result in lower PSNR than JPEG. Note that, according to Table 2.2, IRS achieves the lowest blockiness measure, followed by HSF. However, IRS achieves a slightly better blockiness measure than HSF at the cost of the loss of image detail, which is apparent from the PSNR comparison in Table 2.1 and from Fig. 2.7 and Fig. 2.10. We also find that BF achieves a similar blockiness measure as HSF; however, BF is less effective than HSF in reducing the ringing artifacts in text regions as illustrated by Fig. 2.5 and Fig. 2.6.

Visual comparisons made in Fig. 2.5-Fig. 2.10 provide compelling evidence that HSF outperforms the other decoding methods. As discussed above, overall, HSF is more robust than the other schemes, as evidenced by its consistently good performance on image patches with different characteristics, as well as its good performance as measured by PSNR and the blockiness metric. BR, being a more complex method that requires access to the JPEG encoded DCT coefficients, is not as robust, as illustrated by Fig. 2.8(d) and Fig. 2.9(d). There are at least two sources of this non-robustness. First, artifacts such as those in Fig. 2.9(d) arise when the characteristics of a text region are significantly different from BR's text model. Second, misclassifications produced by BR's segmentation algorithm may result in significant degradation of performance, as in Fig. 2.8(d). Further, as seen from Fig. 2.5-Fig. 2.10 and Table 2.2, BR is not very effective in reducing the blocking artifacts. IRS also suffers from robustness issue in that its median filter causes over-blurring for certain highly texture image regions.

2.5 Conclusion

We proposed the Hypothesis Selection Filter (HSF) as a new approach for image enhancement. The HSF provides a systematic method for combining the advantages

Table 2.1
PSNR of the test images processed by the different schemes.

Scheme	**PSNR of Image I (dB)**	**PSNR of Image II (dB)**
JPEG	27.23	27.84
BF	27.49	27.81
BR	28.17	27.29
RS	28.01	27.55
IRS	27.99	27.50
HSF	28.87	27.57

Table 2.2
Blockiness measure [43] of Image II processed by the different schemes.
A smaller value implies a smaller amount of blocking artifacts.

Scheme	**Blockiness Measure of Image II**
JPEG	1.58
BF	0.89
BR	2.03
RS	0.93
IRS	0.82
HSF	0.88

of multiple image filters, linear or nonlinear, into a single general framework. Our major contributions include the basic architecture of the HSF and a novel unsupervised training procedure for the design of an optimal pixel classifier. The resulting classifier distinguishes the different types of image content and appropriately adjusts the weighting factors of the image filters so that each filter is applied to the regions for which it is most appropriate. This method is particularly appealing in applications where image data are heterogeneous and can benefit from the use of more than one filter.

We demonstrated the effectiveness of the HSF by applying it as a post-processing step for JPEG decoding so as to reduce the JPEG artifacts in the decoded document image. In our scheme, we incorporated 4 different image filters for reducing the JPEG artifacts in the different types of image content that are common in document images, like text, graphics, and natural images. Based on several evaluation methods, including visual inspection of a variety of image patches with different types of content, global PSNR, and a global blockiness measure, our method outperforms state-of-the-art JPEG decoding methods. In addition, the generic structure and the training basis of HSF makes the scheme potentially applicable to many other quality enhancement and image reconstruction tasks.

LIST OF REFERENCES

LIST OF REFERENCES

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

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

[3] B. Oztan, A. Malik, Z. Fan, and R. Eschbach, "Removal of artifacts from JPEG compressed document images," *Proc. SPIE Color Imaging XII: Processing, Hardcopy, and Applications*, vol. 6493, Jan. 2007.

[4] 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, pp. 410–425, 1998.

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

[6] K. Ramchandran and M. Vetterli, "Rate-distortion optimal fast thresholding with complete JPEG/MPEG decoder compatibility," *IEEE Transactions on Image Processing*, vol. 3, no. 5, pp. 700–704, 1994.

[7] M. G. Ramos and S. S. Hemami, "Edge-adaptive JPEG image compression," in *Visual Communications and Image Processing '96*, vol. 2727, pp. 1082–1093, SPIE, 1996.

[8] K. Konstantinides and D. Tretter, "A JPEG variable quantization method for compound documents," *IEEE Trans. Image Process.*, vol. 9, 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, 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, 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, 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. 3, 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, 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, pp. 200–212, Feb. 2005.

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

[16] M.-Y. Shen and C.-C. J. Kuo, "Review of postprocessing techniques for compression artifact removal," *Journal of Visual Communication and Image Representation*, vol. 9, 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," *Journal of Mathematical Imaging and Vision*, vol. 3, 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, pp. 490–500, Apr. 1999.

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

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

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

[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 Conference*, (Scottsdale, AZ), pp. 238–246, Nov. 1996.

[23] H. L. Van Trees, *Detection, Estimation, and Modulation Theory, Part I*, pp. 54–63. John Wiley & Sons, Inc., 1st ed., 1968.

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

[25] J. Besag, "Spatial interaction and the statistical analysis of lattice systems," *J. Roy. Stat. 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, pp. 1745–1759, Oct. 2000.

[28] D. R. Hunter and K. Lange, "A tutorial on MM algorithms," *Amer. Statistician*, vol. 58, pp. 30–37, Feb. 2004.

[29] J. McQueen, “Some methods for classification and analysis of multivariate observations,” *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, pp. 281–297, 1967.

[30] R. L. de Queiroz, “Processing JPEG-compressed images and documents,” *IEEE Trans. Image Process.*, vol. 7, 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, pp. 162–177, Mar. 1994.

[32] C. A. Bouman, “Cluster: An unsupervised algorithm for modeling Gaussian mixtures.” Available from http://www.ece.purdue.edu/˜bouman/software/cluster, Apr. 1997.

[33] H. Siddiqui and C. A. Bouman, “Training-based descreening,” *IEEE Trans. Image Process.*, vol. 16, pp. 789–802, Mar. 2007.

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

[35] C. B. Atkins, C. A. Bouman, and J. P. Allebach, “Optimal image scaling using pixel classification,” in *International Conference on Image Processing*, vol. 3, pp. 864–867, 2001.

[36] B. Zhang, J. S. Gondek, M. T. Schramm, and J. P. Allebach, “Improved resolution synthesis for image interpolation,” *NIP22: International Conference on Digital Printing Technologies, Denver, Colorado*, pp. 343–345, Aug. 2006.

[37] A. Papoulis and S. U. Pillai, *Probability, Random Variables and Stochastic Processes*, ch. 7, pp. 261–272. Mcgraw-Hill, 4th ed., 2002.

[38] A. P. Dempster, N. M. Laird, and D. B. Rubin, “Maximum likelihood from incomplete data via the EM algorithm,” *Journal of the Royal Statistical Society, Series B (Methodological)*, vol. 39, no. 1, pp. 1–38, 1977.

[39] J. Rissanen, “A universal prior for integers and estimation by minimum description length,” *The Annals of Statistics*, vol. 11, no. 2, pp. 416–431, 1983.

[40] C. Tomasi and R. Manduchi, “Bilateral filtering for gray and color images,” in *ICCV ’98: Proceedings of the Sixth International Conference on Computer Vision*, (Washington, DC, USA), p. 839, IEEE Computer Society, 1998.

[41] S. M. Smith and J. M. Brady, “Susan—a new approach to low level image processing,” *Int. J. Comput. Vision*, vol. 23, no. 1, pp. 45–78, 1997.

[42] T.-S. Wong, C. A. Bouman, I. Pollak, and Z. Fan, “A document image model and estimation algorithm for optimized JPEG decompression,” *IEEE Trans. Image Process.*, vol. 18, pp. 2518–2535, Nov. 2009.

[43] Z. Wang, A. Bovik, and B. Evan, “Blind measurement of blocking artifacts in images,” in *Proceedings of International Conference on Image Processing*, vol. 3, pp. 981–984, 2000.

[44] C. Wu, “On the convergence properties of the em algorithm,” *The Annals of Statistics*, vol. 11, pp. 95–103, Mar 1983.

APPENDICES

A. DERIVATION OF THE TRAINING PROCEDURE FOR THE HYPOTHESIS SELECTION FILTER

The statistical model of the HSF consists of two sets of parameters $\theta = \{\pi_j, \sigma_j^2\}_j$ in (2.1) and $\psi = \{\zeta_{j,l}, \mathbf{m}_{j,l}, R_{j,l}\}_{j,l}$ in (2.3). We estimate θ and ψ by their ML estimates, which are defined by

$$\hat{\theta}_{ML}, \hat{\psi}_{ML} = \arg\max_{\theta,\psi} \prod_n f_{X_n,\mathbf{Y}_n}(x_n, \mathbf{y}_n|\theta, \psi). \tag{A.1}$$

Direct computation of $\hat{\theta}_{ML}$ and $\hat{\psi}_{ML}$ based on (A.1) is difficult in that $f_{X_n,\mathbf{Y}_n}(x_n, \mathbf{y}_n|\theta, \psi)$ must be obtained from marginalizing $f_{X_n,\mathbf{Y}_n,J_n,L_n}(x_n, \mathbf{y}_n, j_n, l_n|\theta, \psi)$ since both J_n and L_n are unobserved. This is called the incomplete data problem, which can be solved by the EM algorithm [38, 44].

In its general form, the EM algorithm is given by the following iteration in k

$$\theta^{(k+1)}, \psi^{(k+1)} = \arg\max_{\theta,\phi} Q\left(\theta, \psi|\theta^{(k)}, \psi^{(k)}\right), \tag{A.2}$$

where the function Q is defined as

$$Q\left(\theta, \psi|\theta^{(k)}, \psi^{(k)}\right) = \sum\nolimits_n E\left[\log f_{X_n,\mathbf{Y}_n,J_n,L_n}(x_n, \mathbf{y}_n, J_n, L_n|\theta, \psi)\middle| X_n{=}x_n, \mathbf{Y}_n{=}\mathbf{y}_n, \theta^{(k)}, \psi^{(k)}\right]. \tag{A.3}$$

By applying the iteration in (A.2), it had been proved that the resulting likelihood sequence is monotonic increasing, i.e.

$$\prod_n f_{X_n,\mathbf{Y}_n}(x_n, \mathbf{y}_n|\theta^{(k+1)}, \psi^{(k+1)}) \geq \prod_n f_{X_n,\mathbf{Y}_n}(x_n, \mathbf{y}_n|\theta^{(k)}, \psi^{(k)}) \tag{A.4}$$

for all k. Also, the likelihood sequence converges to a local maximum.

By applying the chain rule to $f_{X_n,\mathbf{Y}_n,J_n,L_n}(x_n,\mathbf{y}_n,J_n,L_n|\theta,\psi)$ and using the fact that (Y_n,L_n) are conditionally independent of X_n given J_n, we have

$$\log f_{X_n,\mathbf{Y}_n,J_n,L_n}(x_n,\mathbf{y}_n,J_n,L_n|\theta,\psi) = \log f_{\mathbf{Y}_n,L_n|J_n}(\mathbf{y}_n,L_n|J_n,\psi) + \log f_{X_n,J_n}(x_n,J_n|\theta) \tag{A.5}$$

By (A.5) and the assumption in (2.10), we can then decompose the function Q as a sum of two terms

$$Q\left(\theta,\psi|\theta^{(k)},\psi^{(k)}\right) = Q_1(\psi|\theta^{(k)},\psi^{(k)}) + Q_2(\theta|\theta^{(k)}), \tag{A.6}$$

where

$$Q_1(\psi|\theta^{(k)},\psi^{(k)}) = \\ \textstyle\sum_n E\left[\log f_{\mathbf{Y}_n,L_n|J_n}(\mathbf{y}_n,L_n|J_n,\psi)\middle|X_n{=}x_n,\mathbf{Y}_n{=}\mathbf{y}_n,\theta^{(k)},\psi^{(k)}\right], \tag{A.7}$$

$$Q_2(\theta|\theta^{(k)}) = \textstyle\sum_n E\left[\log f_{X_n,J_n}(x_n,J_n|\theta)\middle|X_n{=}x_n,\theta^{(k)}\right]. \tag{A.8}$$

The function Q_2 takes the form of (A.8) due to the assumption in (2.10) that J_n is conditionally independent of Y_n given X_n. Further, the density function $f_{X_n,J_n}(x_n,j_n)$ is independent of ψ.

The function Q written in the form of (A.6) allows us to simplify the EM algorithm into a simpler two-stage procedure. During the optimization of Q, since Q_1 is independent of θ, and Q_2 is independent of both ψ and $\psi^{(k)}$, we can first iteratively maximize Q_2 with respect to θ with the following iteration in i

$$\theta^{(i+1)} = \arg\max_{\theta} Q_2\left(\theta|\theta^{(i)}\right), \tag{A.9}$$

$$\hat{\theta} = \lim_{i\to\infty}\theta^{(i)}. \tag{A.10}$$

After the convergence of θ, we maximize Q_1 iteratively with respect to ψ

$$\psi^{(k+1)} = \arg\max_{\phi} Q_1\left(\psi|\hat{\theta}, \psi^{(k)}\right). \tag{A.11}$$

$$\hat{\psi} = \lim_{k\to\infty} \psi^{(k)}. \tag{A.12}$$

The fact that the procedure in (A.9)–(A.12) is equivalent to (A.2) can be seen by considering the limit $\hat{\theta}$ in (A.10) as the initial value $\theta^{(0)}$ in (A.2). The update iteration (A.11) is then the same as (A.2) if we set $\hat{\theta} = \theta^{(k)}$ in (A.2) for all k.

To obtain the explicit update formulas for θ, we first expand the function Q_2 as the following:

$$\begin{aligned} Q_2\left(\theta|\theta^{(i)}\right) &= \textstyle\sum_n \sum_j f_{J_n|X_n}(j|x_n, \theta^{(i)}) \log f_{X_n,J_n}(x_n, j|\theta) \\ &= \textstyle\sum_j \sum_n f_{J_n|X_n}(j|x_n, \theta^{(i)}) \left[\log \pi_j - \dfrac{\log \sigma_j^2}{2} - \dfrac{(x_n - z_{n,j})^2}{2\sigma_j^2} - \dfrac{\log(2\pi)}{2}\right] \\ &= \textstyle\sum_j \left[N_j^{(i)} \log \pi_j - N_j^{(i)} \dfrac{\log \sigma_j^2}{2} - t_j^{(i)} \dfrac{1}{2\sigma_j^2}\right] - \dfrac{N \log(2\pi)}{2}, \end{aligned}$$

where

$$\begin{aligned} N_j^{(i)} &= \textstyle\sum_n f_{J_n|X_n}(j|x_n, \theta^{(i)}), \\ t_j^{(i)} &= \textstyle\sum_n (x_n - z_{n,j})^2 f_{J_n|X_n}(j|x_n, \theta^{(i)}), \end{aligned}$$

and N is the number of training samples. Maximization of Q_2 with respect to π_j (subject to $\sum_j \pi_j = 1$) and σ_j^2 then results to the following update formulas for π_j and σ_j^2

$$\pi_j^{(i+1)} = \frac{N_j^{(i)}}{N}, \tag{A.13}$$

$$\left[\sigma_j^{(i+1)}\right]^2 = \frac{t_j^{(i)}}{N_j^{(i)}}, \tag{A.14}$$

which are equivalent to (2.11)–(2.14).

Similarly, to obtain the update formulas for ψ, we expand the function Q_1 as

$$\begin{aligned} Q_1(\psi|\hat{\theta}, \psi^{(k)}) &= \textstyle\sum_n \sum_j \sum_l f_{J_n, L_n|X_n, \mathbf{Y}_n}(j, l|x_n, \mathbf{y}_n, \hat{\theta}, \psi^{(k)}) \log f_{\mathbf{Y}_n, L_n|J_n}(\mathbf{y}_n, L_n|J_n, \psi) \\ &= \textstyle\sum_j \sum_l \Big[N_{j,l}^{(k)} \log \zeta_{j,l} - \dfrac{1}{2} N_{j,l}^{(k)} \log |R_{j,l}| - \dfrac{1}{2} \operatorname{tr}\left(v_{j,l}^{(k)} R_{j,l}^{-1} \right) \\ &\qquad + \left(\mathbf{u}_{j,l}^{(k)} \right)^t R_{j,l}^{-1} \mathbf{m}_{j,l} - \dfrac{1}{2} N_{j,l}^{(k)} \mathbf{m}_{j,l}^t R_{j,l}^{-1} \mathbf{m}_{j,l} \Big] - \dfrac{DN \log(2\pi)}{2}, \end{aligned}$$

where

$$\begin{aligned} N_{j,l}^{(k)} &= \textstyle\sum_n f_{J_n, L_n|X_n, \mathbf{Y}_n}(j, l|x_n, \mathbf{y}_n, \hat{\theta}, \psi^{(k)}), \\ \mathbf{u}_{j,l}^{(k)} &= \textstyle\sum_n \mathbf{y}_n f_{J_n, L_n|X_n, \mathbf{Y}_n}(j, l|x_n, \mathbf{y}_n, \hat{\theta}, \psi^{(k)}), \\ v_{j,l}^{(k)} &= \textstyle\sum_n \mathbf{y}_n \mathbf{y}_n^t f_{J_n, L_n|X_n, \mathbf{Y}_n}(j, l|x_n, \mathbf{y}_n, \hat{\theta}, \psi^{(k)}), \end{aligned}$$

D is the dimension of $\mathbf{y}_n$, and $\operatorname{tr}(\cdot)$ denotes the trace of a square matrix. Maximization of Q_1 with respect to $\zeta_{j,l}$ (subject to $\sum_l \zeta_{j,l} = 1$), $\mathbf{m}_{j,l}$, and $R_{j,l}$ yields the following update formulas for $\zeta_{j,l}, \mathbf{m}_{j,l}$, and $R_{j,l}$

$$\zeta_{j,l}^{(k+1)} = \frac{N_{j,l}^{(k)}}{N}, \tag{A.15}$$

$$\mathbf{m}_{j,l}^{(k+1)} = \frac{\mathbf{u}_{j,l}^{(k)}}{N_{j,l}^{(k)}}, \tag{A.16}$$

$$R_{j,l}^{(k+1)} = \frac{v_{j,l}^{(k)}}{N_{j,l}^{(k)}} - \frac{\mathbf{u}_{j,l}^{(k)} \left(\mathbf{u}_{j,l}^{(k)} \right)^t}{\left(N_{j,l}^{(k)} \right)^2}. \tag{A.17}$$

The update formulas (A.15)–(A.17) are equivalent to (2.15)–(2.19).

VITA

VITA

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 1999, 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 signal and image processing, statistical image models, document image analysis and processing, image segmentation, inverse problems, machine learning, and computer vision.