Image Enhancement Using the Hypothesis Selection Filter: Theory and Application to JPEG Decoding

Tak-Shing Wong, *Member, IEEE*, Charles A. Bouman, *Fellow, IEEE*, and Ilya Pollak, *Senior Member, IEEE*

***Abstract*— We introduce the hypothesis selection filter (HSF) as a new approach for image quality enhancement. We assume that a set of filters has been selected *a priori* to improve the quality of a distorted image containing regions with different characteristics. At each pixel, HSF uses a locally computed feature vector to predict the relative performance of the filters in estimating the corresponding pixel intensity in the original undistorted image. The prediction result then determines the proportion of each filter used to obtain the final processed output. In this way, the HSF serves as a framework for combining the outputs of a number of different user selected filters, each best suited for a different region of an image. We formulate our scheme in a probabilistic framework where the HSF output is obtained as the Bayesian minimum mean square error estimate of the original image. Maximum likelihood estimates of the model parameters are determined from an offline fully unsupervised training procedure that is derived from the expectation-maximization algorithm. To illustrate how to apply the HSF and to demonstrate its potential, we apply our scheme as a post-processing step to improve the decoding quality of JPEG-encoded document images. The scheme consistently improves the quality of the decoded image over a variety of image content with different characteristics. We show that our scheme results in quantitative improvements over several other state-of-the-art JPEG decoding methods.**

***Index Terms*— Denoising, document image processing, image enhancement, image reconstruction, JPEG decoding.**

I. Introduction

IN MANY applications, a single specific image enhancement filter will not produce the best quality at all locations in a complex image. For example, a filter designed to remove Gaussian noise from smooth regions of an image, will tend to blur edge detail. Or a non-linear filter designed to preserve edge detail, may produce undesirable artifacts in smooth noisy regions of an image. Ideally, we would like to be able to achieve the best result by using a suite of linear or nonlinear filters, with each filter being applied to regions of the image for which it is best suited. However, this approach requires some methodology for selecting the best filter, among an available set of filters, at each location of an image.

In this paper, we present the Hypothesis Selection Filter (HSF) as a new approach for combining the outputs of distinct filters, each of which is chosen to improve a particular type of content in an image. In our formulation, we define M pixel classes where each pixel class is associated with one of the M image filters chosen for the scheme. During the processing of an image, HSF performs a soft classification of each pixel into the M pixel classes through the use of a locally computed feature vector. After classification, the filter outputs are weighted by the resulting class probabilities and combined to form the final processed output. The major contributions of our research include the basic architecture of HSF and a novel probabilistic model which is used to define the pixel classes and to capture the dependence between the feature vector and the defined pixel classes. Based on this model, we derive an unsupervised training procedure for the design of the classifier. The training procedure uses a set of example degraded images and their corresponding high-quality original images to estimate the probability distribution of the feature vector conditioned on each pixel class.

As an example to demonstrate its potential, we apply the HSF as a post-processing step for reducing the artifacts in JPEG-encoded document images. A document image typically consists of text, graphics, and natural images in a complex layout. Each type of content is distorted differently by the JPEG artifacts due to its unique characteristics. In this application, we use four image filters in the HSF. The image filters are selected to reduce the JPEG artifacts for different types of image content. Comparing with several other state-of-the-art approaches, HSF tends to improve the decoding quality more consistently over different types of image content.

A variety of approaches have been proposed for enhancing image quality. Traditional linear filters [1] have the main advantage of simplicity and are backed by a rich theoretical foundation. In the realm of nonlinear filtering, median filters [2]–[4], weighted median filters [5]–[7], the order-statistic filters [8], and stack filters [9], [10] are examples of the wide range of nonlinear filters that have been studied. More recently, many spatially adaptive methods of filtering have been developed to address the different aspects of image quality. Zhang *et al.* [11] proposed an adaptive version of the bilateral filter [12], [13] for image sharpening and denoising. The behavior of the filter is locally adaptive based on the response of a Laplacian of Gaussian [14] operator. In [15], Hu and de Haan combine the outputs of a linear filter and a rank order filter for reducing both Gaussian noise and impulsive noise. Coefficients of the combined filter are locally adjusted using Adaptive Dynamic Range Coding [16]. The non-local means algorithm [17], proposed by Buades *et al.*, performs

Manuscript received September 23, 2011; revised July 3, 2012; accepted September 3, 2012. Date of publication September 21, 2012; date of current version January 24, 2013. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Rafael Molina.

The authors are with the School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN 47907 USA (e-mail: wil@purdue.edu; bouman@purdue.edu; ipollak@purdue.edu).

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

Digital Object Identifier 10.1109/TIP.2012.2220149

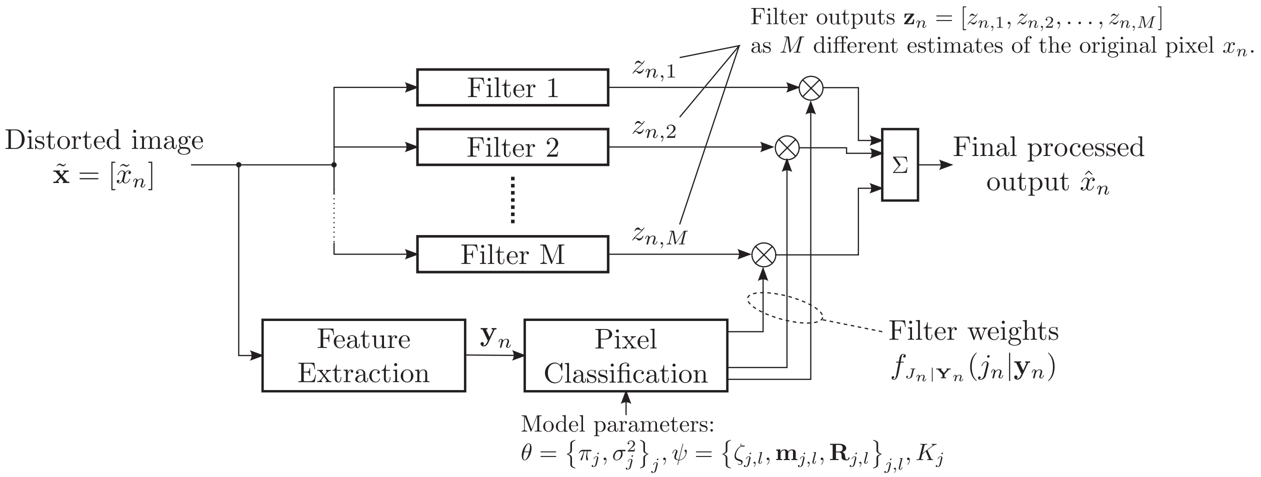

Filter outputs $\mathbf{z}_n = [z_{n,1}, z_{n,2}, \ldots, z_{n,M}]$
as M different estimates of the original pixel x_n.
Distorted image
$\tilde{\mathbf{x}} = [\tilde{x}_n]$
Filter 1
Filter 2
Filter M
$z_{n,1}$
$z_{n,2}$
$z_{n,M}$
Σ
Final processed output $\hat{x}_n$
Feature Extraction
$\mathbf{y}_n$
Pixel Classification
Filter weights
$f_{J_n|\mathbf{Y}_n}(j_n|\mathbf{y}_n)$
Model parameters:
$\theta = \{\pi_j, \sigma_j^2\}_j, \psi = \{\zeta_{j,l}, \mathbf{m}_{j,l}, \mathbf{R}_{j,l}\}_{j,l}, K_j$

Fig. 1. HSF processes the distorted image $\tilde{\mathbf{x}}$ by M filters chosen to improve the image quality for regions with different characteristics. At each pixel, a feature vector $\mathbf{y}_n$ is computed using local image data. The feature vector is then used to determine the optimal filter weights to form the final output. The mapping between the feature vector and the optimal filter weights is controlled by a number of model parameters which are determined through training.

image denoising by spatially varying averaging of image patches that are similar to the current local neighborhood. Shao *et al.* proposed the classification-based least squares trained filters [18], [19] for various video quality enhancement tasks including sharpening, interpolation, and coding artifact removal. Several other adaptive approaches for image denoising can be found in [20], [21]. In [22], Atkins *et al.* introduced the Resolution Synthesis (RS) framework and applied it for image interpolation. The scheme applies unsupervised clustering to identify important classes of local image structures, and designs a minimum mean square error filter for each class by training. In the comprehensive survey conducted by van Ouwerkerk [23], RS interpolation compared favorably to several other state-of-the-art image interpolation schemes. Hasib *et al.* further extended the framework to halftone descreening [24] and color correction [25]. Zhang *et al.* [26] also adapted the framework for coding artifact reduction by using encoded images for training. Other examples of spatially adaptive methods for filtering and quality improvement include, but are not limited to, image sharpening [27], [28], and contrast enhancement [29], [30]. Our scheme may also be considered as a spatially adaptive image enhancement method. However, instead of applying a particular algorithm adaptively to the local content, our scheme is unique in that it provides a framework for combining any set of filters or algorithms, whether they are linear or non-linear, iterative or non-iterative.

We introduce the overall architecture for HSF in Section II. The underlying probabilistic model for image pixels and feature vectors is discussed in Section II-A. We derive the output of HSF as the minimum mean square error estimate of the original image in Section II-B. This is followed by the description of the parameter estimation procedure in Section II-C. We apply HSF to improving the quality of JPEG decoding in Sections III and IV: Section III discusses the image filters and the features we use in HSF for this application; and Section IV presents our experimental results and comparisons with several state-of-the-art JPEG decoding methods.

II. Hypothesis Selection Filter

We first present the Hypothesis Selection Filter in a general setting. For a given image enhancement problem, suppose we have selected a set of image filters each of which is effective in improving the image quality for a certain type of content. Our objective is to combine the filter outputs to obtain an overall best quality result. In the following, we use upper-case letters to denote random variables and random vectors, and lower-case letters to denote their realizations.

The structure of HSF is shown in Fig. 1. In the upper portion of Fig. 1, the HSF processes the input image $\tilde{\mathbf{x}}$ using the M filters we have selected. We consider the input image as a distorted version of a high quality original image $\mathbf{x}$. For the n-th pixel, the filter outputs, denoted by $\mathbf{z}_n = \left[z_{n,1}, \ldots, z_{n,M}\right]^t$, will each serve as a different intensity estimate of the corresponding pixel in the original image, x_n. We assume that a judiciously selected feature vector $\mathbf{y}_n$ carries information on how to combine the filter outputs to produce the best estimate of x_n. We also define M pixel classes corresponding to the M image filters. In the lower portion of Fig. 1, HSF computes the feature vector for the current pixel, followed by a soft classification of the current pixel into the M pixel classes. The soft classification computes the conditional probability of the pixel to belong to each one of the pixel classes, given the feature vector. The resulting probabilities are used to weight the filter outputs, which are then summed to form the final processed output. The original pixel intensity x_n, the feature vector $\mathbf{y}_n$, and the filter outputs $\mathbf{z}_n$ are realizations of the random variables/vectors X_n, $\mathbf{Y}_n$, and $\mathbf{Z}_n$ respectively, whose probabilistic model we describe next.

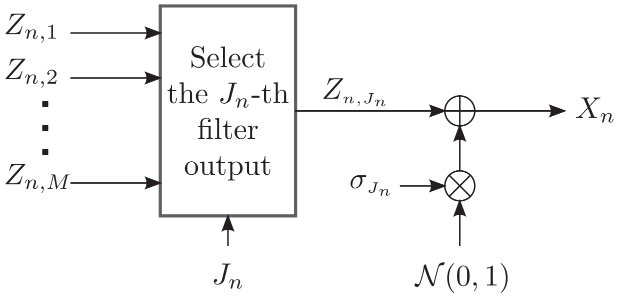


Fig. 2. Generative probabilistic model for the original pixel X_n. An unobserved discrete random variable $J_n \in \{1, \ldots, M\}$ is sampled with prior probability $\text{Prob}\{J_n{=}j\} = \pi_j$. Given $J_n = j$ and $\mathbf{Z}_n = \mathbf{z}_n$, we model X_n as a conditionally Gaussian random variable with mean $z_{n,j}$ and variance σ_j^2.

A. Probabilistic Model

Our probabilistic model is based on four assumptions about the original pixel intensity X_n, the feature vector $\mathbf{Y}_n$, and the filter outputs $\mathbf{Z}_n$. Based on this model, we derive the output of HSF as the minimum mean square error [31] estimate of X_n.

Assumption 1: Given $\mathbf{Z}_n = \mathbf{z}_n$, the conditional distribution of X_n is a Gaussian mixture [32] with M components:

$$f_{X_n|\mathbf{Z}_n}(x_n \,|\, \mathbf{z}_n) = \sum_{j=1}^{M} \pi_j \, \mathcal{N}\Big(x_n; z_{n,j}, \sigma_j^2\Big) \tag{1}$$

where the means of the Gaussian components are the filter outputs $z_{n,j}$; π_j and σ_j^2 are parameters of the distribution function; and $\mathcal{N}\big(\,\cdot\,; \mu, \sigma^2\big)$ denotes the univariate Gaussian distribution function with mean μ and variance σ^2. We also define an unobserved random variable $J_n \in \{1, \ldots, M\}$ to represent the component index in (1). This random variable is independent of $\mathbf{Z}_n$, and the prior probability mass function of J_n is given by

$$\text{Prob}\{J_n{=}j\} = \pi_j \tag{2}$$

for $j = 1, \ldots, M$. The conditional probability distribution of X_n given $J_n = j$ and $\mathbf{Z}_n = \mathbf{z}_n$ is given by

$$f_{X_n|J_n,\mathbf{Z}_n}(x_n | j, \mathbf{z}_n) = \mathcal{N}\Big(x_n; z_{n,j}, \sigma_j^2\Big).$$

A graphical illustration of (1) is shown in Fig. 2[1].

Assumption 2: Given the component index $J_n = j$, the conditional distribution of the feature vector $\mathbf{Y}_n$ is a multivariate Gaussian mixture [32]:

$$f_{\mathbf{Y}_n|J_n}(\mathbf{y}_n | j) = \sum_{l=1}^{K_j} \zeta_{j,l} \, \mathcal{N}\big(\mathbf{y}_n; \, \mathbf{m}_{j,l}, \, \mathbf{R}_{j,l}\big) \tag{3}$$

where K_j is the order of the Gaussian mixture distribution; $\zeta_{j,l}$, $\mathbf{m}_{j,l}$, and $\mathbf{R}_{j,l}$ are respectively the weight, the mean vector, and the covariance matrix of the l-th Gaussian component; and $\mathcal{N}(\,\cdot\,; \mathbf{m}, \mathbf{R})$ denotes the multivariate Gaussian distribution function with mean vector $\mathbf{m}$ and covariance matrix $\mathbf{R}$. Similar to Assumption 1, we also define the discrete random variable $L_n \in \{1, \ldots, K\}$ to denote the component index in (3), where $K = \max_j K_j$, and $\zeta_{j,l} = 0$ whenever $l > K_j$. With this definition, the conditional probability of $L_n = l$ given $J_n = j$ is

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

for $l = \{1, \ldots, K\}$. The conditional probability distribution of $\mathbf{Y}_n$ given $J_n = j$ and $L_n = l$ is

$$f_{\mathbf{Y}_n|J_n,L_n}(\mathbf{y}_n | j, l) = \mathcal{N}\big(\mathbf{y}_n; \, \mathbf{m}_{j,l}, \, \mathbf{R}_{j,l}\big).$$

Combining (2) and (3), we are effectively assuming that $\mathbf{Y}_n$ is a mixture of Gaussian mixtures:

$$f_{\mathbf{Y}_n}(\mathbf{y}_n) = \sum_{j=1}^{M} \pi_j \left[\sum_{l=1}^{K_j} \zeta_{j,l} \, \mathcal{N}\big(\mathbf{y}_n; \, \mathbf{m}_{j,l}, \, \mathbf{R}_{j,l}\big) \right].$$

Assumption 3: The two sets of random quantities $\{X_n, \mathbf{Z}_n\}$ and $\{\mathbf{Y}_n, L_n\}$ are conditionally independent given the component index $J_n = j$:

$$\begin{aligned} &f_{X_n,\mathbf{Y}_n,\mathbf{Z}_n,L_n|J_n}(x_n, \mathbf{y}_n, \mathbf{z}_n, l | j) \\ &= f_{X_n,\mathbf{Z}_n|J_n}(x_n, \mathbf{z}_n | j) \; f_{\mathbf{Y}_n,L_n|J_n}(\mathbf{y}_n, l | j). \end{aligned}$$

Assumption 4: Given $X_n = x_n$ and $\mathbf{Z}_n = \mathbf{z}_n$, the component index J_n is conditionally independent of the feature vector $\mathbf{Y}_n$, or equivalently

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

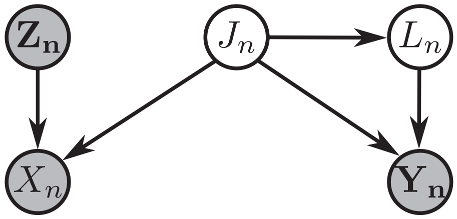


Fig. 3. Graphical model for the HSF. The shaded nodes X_n, $\mathbf{Y_n}$, and $\mathbf{Z_n}$ represent observable variables. Unshaded nodes J_n and L_n represent latent (unobservable) variables.

In Fig. 3, we show the graphical model [33] for the HSF to illustrate the conditional dependency of the different random quantities. The graphical model corresponds to the factorization of the joint probability distribution function

$$\begin{aligned} &f(x_n, \mathbf{y}_n, \mathbf{z}_n, j_n, l_n) \\ &= f(x_n | \mathbf{z}_n, j_n) f(\mathbf{y}_n | j_n, l_n) f(l_n | j_n) f(\mathbf{z}_n) f(j_n) \end{aligned}$$

where the subscripts of the distribution functions are omitted to simplify notation.

In Assumption 1, the use of (1) provides a mechanism whereby the M pixel classes may be defined from training based on the errors made by the image filters for a set of training samples. We assume that our training samples are of the form $(x_n, \mathbf{y}_n, \mathbf{z}_n)$, where x_n is the ground truth value of the original undistorted pixel, $\mathbf{y}_n$ is the feature vector extracted from the distorted image, and $\mathbf{z}_n$ are the corresponding filter outputs. Therefore, the absolute error made by each filter, $\big|x_n - z_{n,j}\big|$, can be computed during training. As explained in Section II-C, our training procedure uses (1) to compute the conditional class probability given the ground truth and

[1] Notice that we are interested in inferring X_n from $\mathbf{Z}_n$ and are modeling this relationship directly—which is one of the novelties about our approach. The usual approach of modeling $f_{\mathbf{Z}_n|X_n}(\mathbf{z}_n | x_n)$ requires also a model for $f_{X_n}(x_n)$ and complicated computations to calculate an estimate of X_n given $\mathbf{Z}_n$. Our approach turns this around.

the filter outputs, $f_{J_n|X_n,\mathbf{Z}_n}(j|x_n, \mathbf{z}_n)$. By applying Bayes' rule to (1), this conditional class probability can be written as

$$f_{J_n|X_n,\mathbf{Z}_n}(j|x_n, \mathbf{z}_n) = \frac{\frac{\pi_j}{\sqrt{2\pi\sigma_j^2}} \exp\left\{-\frac{|x_n - z_{n,j}|^2}{2\sigma_j^2}\right\}}{\sum_{j'} \frac{\pi_{j'}}{\sqrt{2\pi\sigma_{j'}^2}} \exp\left\{-\frac{|x_n - z_{n,j'}|^2}{2\sigma_{j'}^2}\right\}}.$$

Therefore, this conditional class probability is a monotonically decreasing function of the absolute error $|x_n - z_{n,j}|$ made by the j-th filter. This insures that filters making larger errors on the training data will have less influence in our soft classifier. In fact, the conditional class probability is also a monotonically decreasing function of the standard score [34], $\frac{|x_n - z_{n,j}|}{\sigma_j}$. As explained in Section II-C, the error variance of each filter, σ_j^2, is also estimated during the training procedure. Therefore, if filter j has been poorly chosen for the scheme and consistently makes large errors, its role in the scheme will be diminished by a large estimate of σ_j^2.

In assumption 2, we use the Gaussian mixture distribution as a general parametric model to characterize the feature vector for each pixel class.

In Assumption 3 and Assumption 4, we use the latent variable J_n to decouple the dependence among the variables X_n, $\mathbf{Y}_n$, and $\mathbf{Z}_n$ so as to obtain a simpler probabilistic model.

B. Optimal Filtering

HSF computes the Bayesian minimum mean square error (MMSE) estimate $\hat{X}_n$ of X_n based on observing the feature vector $\mathbf{Y}_n$ and the filter outputs $\mathbf{Z}_n$. The MMSE estimate is the conditional expectation of X_n given $\mathbf{Y}_n$ and $\mathbf{Z}_n$ [31]:

$$\begin{aligned}
\hat{X}_n &= E[X_n|\mathbf{Y}_n, \mathbf{Z}_n] && (4)\\
&= \sum_{j=1}^{M} E[X_n|J_n{=}j, \mathbf{Y}_n, \mathbf{Z}_n]\; f_{J_n|\mathbf{Y}_n,\mathbf{Z}_n}(j|\mathbf{Y}_n, \mathbf{Z}_n) && (5)\\
&= \sum_{j=1}^{M} E[X_n|J_n{=}j, \mathbf{Z}_n]\; f_{J_n|\mathbf{Y}_n,\mathbf{Z}_n}(j|\mathbf{Y}_n, \mathbf{Z}_n) && (6)\\
&= \sum_{j=1}^{M} Z_{n,j}\; f_{J_n|\mathbf{Y}_n,\mathbf{Z}_n}(j|\mathbf{Y}_n, \mathbf{Z}_n) && (7)\\
&= \sum_{j=1}^{M} Z_{n,j}\; f_{J_n|\mathbf{Y}_n}(j|\mathbf{Y}_n) && (8)\\
&= \sum_{j=1}^{M} Z_{n,j}\, \frac{\pi_j\, f_{\mathbf{Y}_n|J_n}(\mathbf{Y}_n|j)}{\sum_{j'=1}^{M} \pi_{j'}\, f_{\mathbf{Y}_n|J_n}(\mathbf{Y}_n|j')} && (9)
\end{aligned}$$

where we have used the total expectation theorem to go from (4) to (5); Assumption 3 that the pixel value X_n and the feature vector $\mathbf{Y}_n$ are conditionally independent given $J_n = j$ to go from (5) to (6); and Assumption 1 that $Z_{n,j}$ is the conditional mean of X_n given $J_n = j$ and $\mathbf{Z}_n$ to go from (6) to (7). To go from (7) to (8), we have assumed that the filter outputs $\mathbf{Z}_n$ and the class label J_n are conditionally independent given the feature vector $\mathbf{Y}_n$. This assumption does not result in any loss of generality since we can include filter outputs into the feature vector. Note that (8) expresses the estimate $\hat{X}_n$ as a weighted sum of the filter outputs, where the weights are the conditional probabilities of pixel classes given the feature vector. To go from (8) to (9), we have used Bayes' rule.

In order to evaluate the right-hand side of (9), we need to estimate the following unknown parameters: the component weights π_j and component variance σ_j^2 from (1); and all the parameters of the M different Gaussian mixture distributions, $f_{\mathbf{Y}_n|J_n}(\mathbf{Y}_n|j)$, from (3). We obtain maximum likelihood estimates of these parameters through the expectation-maximization algorithm, as detailed in the next subsection.

C. Parameter Estimation

Our statistical model consists of three sets of parameters: $\theta = \{\pi_j, \sigma_j^2\}_j$ from (1); $\psi = \{\zeta_{j,l}, \mathbf{m}_{j,l}, \mathbf{R}_{j,l}\}_{j,l}$ from (3); and the orders K_j of the Gaussian mixture distributions, $f_{\mathbf{Y}_n|J_n}(\mathbf{y}_n|j)$, also from (3). We estimate the parameters by an unsupervised training procedure. The training procedure makes use of a set of training samples, each of which is in the form of a triplet $(x_n, \mathbf{y}_n, \mathbf{z}_n)$, extracted from example pairs of original and degraded images.

We seek the maximum likelihood (ML) estimates [31] of θ and ψ, which are defined as the values that maximize the likelihood of the observed training samples

$$\left(\hat{\theta}_{ML},\ \hat{\psi}_{ML}\right) = \underset{(\theta,\psi)}{\arg\max} \prod_n f_{X_n,\mathbf{Y}_n,\mathbf{Z}_n}(x_n, \mathbf{y}_n, \mathbf{z}_n|\theta, \psi)$$

where the probability distribution $f_{X_n,\mathbf{Y}_n,\mathbf{Z}_n}(x_n, \mathbf{y}_n, \mathbf{z}_n|\theta, \psi)$ is obtained from marginalizing the random variables J_n and L_n in $f_{X_n,\mathbf{Y}_n,\mathbf{Z}_n,J_n,L_n}(x_n, \mathbf{y}_n, \mathbf{z}_n, j, l|\theta, \psi)$. 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 [35]–[37]. We present the derivation of the EM algorithm in the Appendix. In this section, we describe the algorithmic procedure which implements the derived EM algorithm.

In deriving the EM algorithm, we apply Assumption 3 that $\{\mathbf{Y}_n, L_n\}$ are conditionally independent of $\{X_n, \mathbf{Z}_n\}$ given J_n 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:

$$\hat{f}_{J_n|X_n,\mathbf{Z}_n}(j|x_n, \mathbf{z}_n) \leftarrow \frac{\hat{\pi}_j\, \mathcal{N}\left(x_n; z_{n,j}, \hat{\sigma}_j^2\right)}{\sum_{j'} \hat{\pi}_{j'}\, \mathcal{N}\left(x_n; z_{n,j'}, \hat{\sigma}_{j'}^2\right)} \quad (10)$$

$$\hat{N}_j \leftarrow \textstyle\sum_n \hat{f}_{J_n|X_n,\mathbf{Z}_n}(j|x_n, \mathbf{z}_n) \quad (11)$$

$$\hat{t}_j \leftarrow \textstyle\sum_n (x_n - z_{n,j})^2 \hat{f}_{J_n|X_n,\mathbf{Z}_n}(j|x_n, \mathbf{z}_n) \quad (12)$$

$$\left(\hat{\pi}_j,\ \hat{\sigma}_j^2\right) \leftarrow \left(\frac{\hat{N}_j}{N},\ \frac{\hat{t}_j}{\hat{N}_j}\right) \quad (13)$$

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 $\left\{\zeta_{j,l}, \mathbf{m}_{j,l}, \mathbf{R}_{j,l}\right\}_l$ by applying the following set of update iterations until convergence:

$$\begin{aligned}&\hat{f}_{J_n,L_n|X_n,\mathbf{Y}_n,\mathbf{Z}_n}(j,l|x_n,\mathbf{y}_n,\mathbf{z}_n)\\&\leftarrow \hat{f}_{J_n|X_n,\mathbf{Z}_n}(j|x_n,\mathbf{z}_n)\,\frac{\hat{\zeta}_{j,l}\,\mathcal{N}\left(\mathbf{y}_n;\hat{\mathbf{m}}_{j,l},\hat{\mathbf{R}}_{j,l}\right)}{\sum\limits_{l'}\hat{\zeta}_{j,l'}\,\mathcal{N}\left(\mathbf{y}_n;\hat{\mathbf{m}}_{j,l'},\hat{\mathbf{R}}_{j,l'}\right)}\end{aligned} \tag{14}$$

$$\hat{N}_{j,l} \leftarrow \textstyle\sum_n \hat{f}_{J_n,L_n|X_n,\mathbf{Y}_n,\mathbf{Z}_n}(j,l|x_n,\mathbf{y}_n,\mathbf{z}_n), \tag{15}$$

$$\hat{\mathbf{u}}_{j,l} \leftarrow \textstyle\sum_n \mathbf{y}_n\,\hat{f}_{J_n,L_n|X_n,\mathbf{Y}_n,\mathbf{Z}_n}(j,l|x_n,\mathbf{y}_n,\mathbf{z}_n), \tag{16}$$

$$\hat{\mathbf{v}}_{j,l} \leftarrow \textstyle\sum_n \mathbf{y}_n\,\mathbf{y}_n^t\,\hat{f}_{J_n,L_n|X_n,\mathbf{Y}_n,\mathbf{Z}_n}(j,l|x_n,\mathbf{y}_n,\mathbf{z}_n), \tag{17}$$

$$\left(\hat{\zeta}_{j,l},\,\hat{\mathbf{m}}_{j,l},\,\hat{\mathbf{R}}_{j,l}\right) \leftarrow \left(\frac{\hat{N}_{j,l}}{\hat{N}_j},\,\frac{\hat{\mathbf{u}}_{j,l}}{\hat{N}_{j,l}},\,\frac{\hat{\mathbf{v}}_{j,l}}{\hat{N}_{j,l}} - \frac{\hat{\mathbf{u}}_{j,l}\,\hat{\mathbf{u}}_{j,l}^t}{\hat{N}_{j,l}^2}\right). \tag{18}$$

We apply the minimum description length (MDL) criterion [38] to estimate the order K_j of the Gaussian mixture distribution of (3), for each $j = 1, \ldots, M$. The MDL criterion 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 $\psi_j = \left\{\zeta_{j,l}, \mathbf{m}_{j,l}, \mathbf{R}_{j,l}\right\}_l$. More specifically, we start with an initial estimate $K_j = 30$ for the model order and successively reduce the value of K_j. For each value of K_j, we estimate ψ_j by (14)–(18) 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 [39].

D. Computation Complexity

We briefly summarize the complexity of the algorithm in Table I. In the training procedure, Stage 1 consists of the updates in (10)–(13), where the computation is dominated by (10)–(12). In one iteration, the update (10) has complexity $O(MN)$ since the computation of $\hat{\pi}_j\,\mathcal{N}\left(x_n; z_{n,j}, \hat{\sigma}_j^2\right)$ takes constant time. Both (11) and (12) are executed M times, where each execution has complexity $O(N)$. Thus the overall complexity is also $O(MN)$. Stage 2 of training consists of the updates in (14)–(18), where the computation is dominated by (14). In (14), the computation of $\mathcal{N}\left(\mathbf{y}_n; \hat{\mathbf{m}}_{j,l'}, \hat{\mathbf{R}}_{j,l'}\right)$ has complexity $O(D^2)$, where D is the dimension of the feature vector $\mathbf{y_n}$. The update (14) is executed once for each training sample, each filter class, and each sub-cluster $l = 1, \ldots, K$. Therefore, the overall complexity is $O(MNKD^2)$. During image filtering, each execution of (8) has complexity $O(MKD^2)$. This is because $f_{J_n|\mathbf{Y_n}}(j|\mathbf{y_n})$ is a K-th order Gussian mixture distribution, given by (3), where the computation of each Gaussian component has complexity $O(D^2)$. Therefore, the overall complexity of processing a N pixel image is $O(MNKD^2)$.

In Table I, we also summarize the typical execution time for the JPEG decoding application of HSF described in Section III, where we have used $M = 4$ image filters and a $D = 5$ dimensional feature vector. We have also fixed the order of the Gaussian mixture distribution $f_{J_n|\mathbf{Y_n}}(j|\mathbf{y_n})$ to $K = 20$. Note that in image filtering, we apply the trained scheme to the R, G, B components of the color image separately. Thus, the computation of (3) is performed three times for each pixel in the image.

TABLE I

HSF Computational Complexity and Typical Execution Time. M is the Number of Filters. N is the Number of Pixels. D is the Feature Vector Dimension. K is the Order of the Gaussian Mixture $f_{\mathbf{Y_n}|\mathbf{J_n}}$. Execution Times are Measured for the Application in Section III on a 2-GHz Dual Core CPU 2-GB Ram Computer, With $M = 4$, $D = 5$, and $K = 20$

Algorithm	Computational Complexity	Execution Time
Training, Stage 1 (per iteration)	$O(MN)$	0.48 sec (N = 952320)
Training, Stage 2 (per iteration)	$O(MNKD^2)$	102 sec (N = 952320)
Image Filtering	$O(MNKD^2)$	50.2 sec (N = 699392)

III. HSF for JPEG Decoding

We have presented the HSF in a general context for image enhancement in the previous section. This method can be applied to any 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, we present an application of the HSF to improving the decoding quality for any JPEG-encoded document images. JPEG compression [40] 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 sharp edges that lead to severe ringing artifacts in these regions, while natural images are usually more susceptible to 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 Sections III-A and III-B, respectively. Then we briefly describe our training procedure in Section III-C. 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.

A. Filter Selection

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

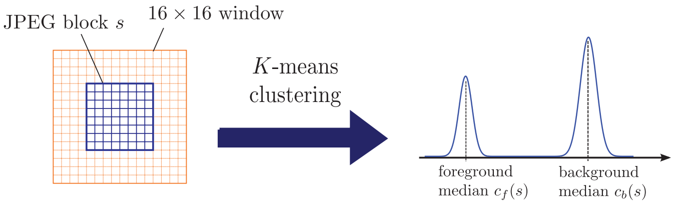


Fig. 4. Local foreground/background gray-level estimation. For each JPEG block, we center a 16×16 window around the block and partition the 256 pixels of the window into two groups by the K-means clustering algorithm. Then, we use the two cluster medians as the local foreground gray-level estimate and the local background gray-level estimate for the block.

The first filter that we select for the HSF is a Gaussian filter with kernel standard deviation $\sigma = 1$. We choose σ large enough so that the Gaussian filter can effectively eliminate the coding artifacts in the smooth picture regions, even for images encoded at a high compression ratio. Applied alone, however, the Gaussian filter will lead to significant blurring of image detail. Therefore, to handle the picture regions that contain edges and textures, we use a bilateral filter [12], [13] with geometric spread $\sigma_d = 0.5$ and photometric spread $\sigma_r = 10$ as the second filter in the HSF. We select σ_d to be relatively small, as compared to the Gaussian filter's kernel width σ, so that the bilateral filter can moderately remove the coding artifacts without blurring edges and image detail for non-smooth picture regions.

To eliminate the ringing artifacts around the text, we apply another two filters in the HSF. A text region typically consists of pixels which concentrate around two intensities, corresponding to the background and the text foreground. Assuming that the local region is a two-intensity text region, 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. 4. 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. For a text region that satisfies the two-intensity assumption, we can reduce the ringing artifacts by first classifying each pixel into either foreground or background. Foreground pixels will then be clipped to the foreground gray-level estimate, and background pixels will be clipped to the background gray-level estimate. In our feature vector, described in detail in Section III-B, a feature component is designed to detect whether the local region is a two-intensity text region. Another two feature components detect whether a pixel is a foreground pixel or a background pixel. Lastly, we should point out that there are also text regions which do not satisfy our two-intensity assumption. For examples, in graphic design, a text region may use different colors for the individual letters or a smoothly varying text foreground. Even in a text region that satisfies the two-intensity assumption, the edge pixels may still deviate substantially from the foreground and background intensities. For these cases, the pixels will be handled by the first two filters of the HSF.

B. 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. More specifically, our feature vector must capture the relevant information to distinguish between picture regions and text regions. For the picture regions, the feature vector also separates the smooth regions from the regions that contain textures and edges. For the text regions, the feature vector detects whether an individual pixel belongs to the background or to the text foreground.

For the first feature component, we compute the variance of the JPEG block associated with the current pixel. The block variance is used to detect the smooth regions in the image, which generally have small block variance.

For the second feature component, we compute the local gradient magnitude at the current pixel using the Sobel operator [41] to detect major edges. The Sobel operator consists of a pair of 3×3 kernels. Convolving the image with these two kernels produces a 2-dimensional local gradient estimate at each pixel. The second feature component is then computed as the norm of the gradient vector.

The third 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. The feature is designed to detect the text regions that satisfy the two-intensity assumption. 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

$$b_s = \frac{\sum_i \min\left(|u_{s,i} - c_b(s)|^2, |u_{s,i} - c_f(s)|^2\right)}{64\ |c_b(s) - c_f(s)|^2} \tag{19}$$

where $c_b(s)$ and $c_f(s)$ are the estimates of the local foreground gray-level and the local background gray-level respectively, obtained from the third and the forth filters in Section III-A. We compute the two-level normalized variance of the JPEG block associated with the current pixel as the third feature component of the pixel. In (19), the numerator is proportional to the mean square error between the image block and its two-level quantization to the local foreground and background estimates. The denumerator is proportional to the local contrast. Therefore, the feature component for a two-level text block will tend to be small.

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 $c_b(s)$:

$$\begin{aligned} y_{n,4} &= u_n - c_f(s) \\ y_{n,5} &= u_n - c_b(s) \end{aligned}$$

where u_n is the gray-level of the current pixel, and s is the JPEG block associated with the pixel. When the local region is a text region, these two features are used to distinguish between the text foreground and background. They help avoid applying the outputs of the last two filters when their errors are large.

TABLE II
COMPARISON OF DIFFERENT SCHEMES' PARAMETER SETTINGS

Scheme	Parameter Setting
BF	1) Parameters are chosen to match with the bilateral filter used in the HSF. 2) Geometric spread $\sigma_d = 0.5$. 3) Photometric spread $\sigma_r = 10$.
BR	1) Parameters are chosen according to Table I of [44].
RS	1) Number of classes is chosen by the MDL criterion [38]. 2) Parameters of classifier and linear filters are determined from training.
IRS	1) Number of classes is chosen by the MDL criterion [38]. 2) Parameters of classifier and linear filters are determined from training. 3) RS filtering followed by a 3 × 3 median filter [4].

(a) (b)

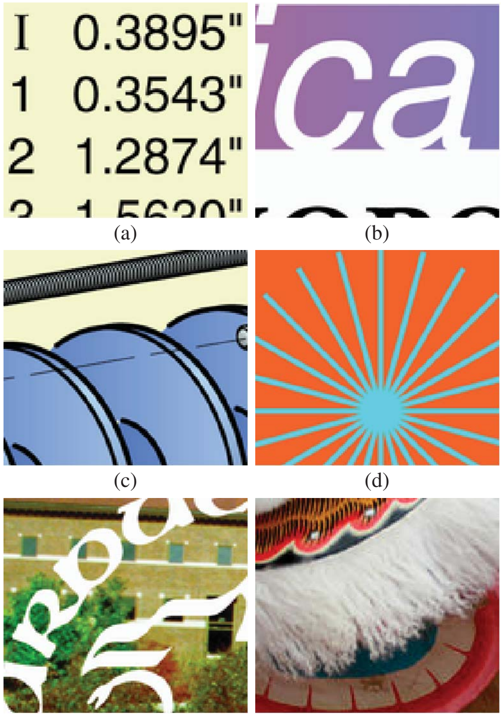
Fig. 5. Thumbnails of test images used for visual quality comparison. Image I was JPEG encoded at 0.69 bits per pixel (bpp). Image II was JPEG encoded at 0.96 bpp. (a) Dimension 3131 × 2395. (b) Dimension 683 × 1024.

C. Training

To perform training for the HSF, we use a set of 24 natural images and six text images. The natural images are from the Kodak lossless true color image suite [42], each of size 512 × 768. Each text image is 7200 × 6800, with text of different fonts, sizes, and contrasts. We use the Independent JPEG Group's implementation of JPEG encoder [43] to encode the original training images. Each original image is JPEG encoded at three different quality settings corresponding to medium to high compression ratios, resulting in 90 pairs of original and degraded images. We then apply our unsupervised training procedure described in Section II-C to the pairs of original and JPEG-encoded images. For simplicity, we only use monochrome images to perform training. That is, a true color image is first converted to a monochrome image before JPEG encoding. 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 IV, none of the test images are used in training.

IV. RESULTS AND COMPARISON

We compare our results with (i) the bilateral filter (BF) [12], [13] with parameters $\sigma_d = 0.5$ and $\sigma_r = 10$; (ii) a segmentation-based Bayesian reconstruction scheme (BR) [44]; (iii) Resolution Synthesis (RS) [22]; and (iv) Improved Resolution Synthesis (IRS) [26]. 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. In Table II, we briefly summarize how the parameters of these four schemes are determined. For the two training-based schemes, RS and IRS, we perform training using the same set of training images that we used for HSF, which are described in Section III-C.



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

Fig. 6. Enlarged regions of (a)–(e) image I and (f) image II, used for the comparison in Figs. 7–18. The regions are chosen as examples of (a) and (b) text, (c) and (d) graphics, (e) mixed (natural image overlaid with text), and (f) natural images.

In Fig. 5, we show the thumbnails of two test images that we use for comparing the visual quality of the different schemes. Fig. 6 shows six different regions extracted from the originals of the two test images. The regions contain examples of text

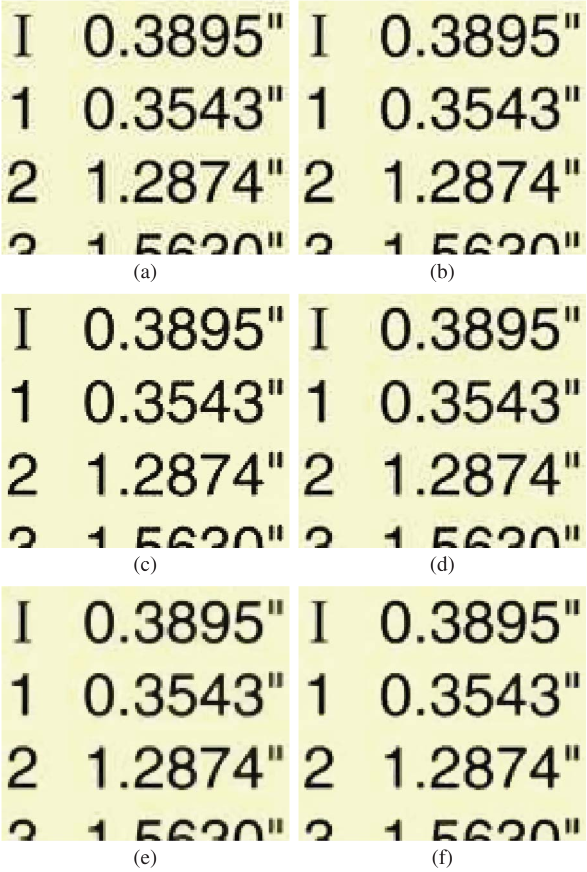


Fig. 7. Decoding results for Region 1. (a) JPEG. (b) BF. (c) BR. (d) RS. (e) IRS. (f) HSF.

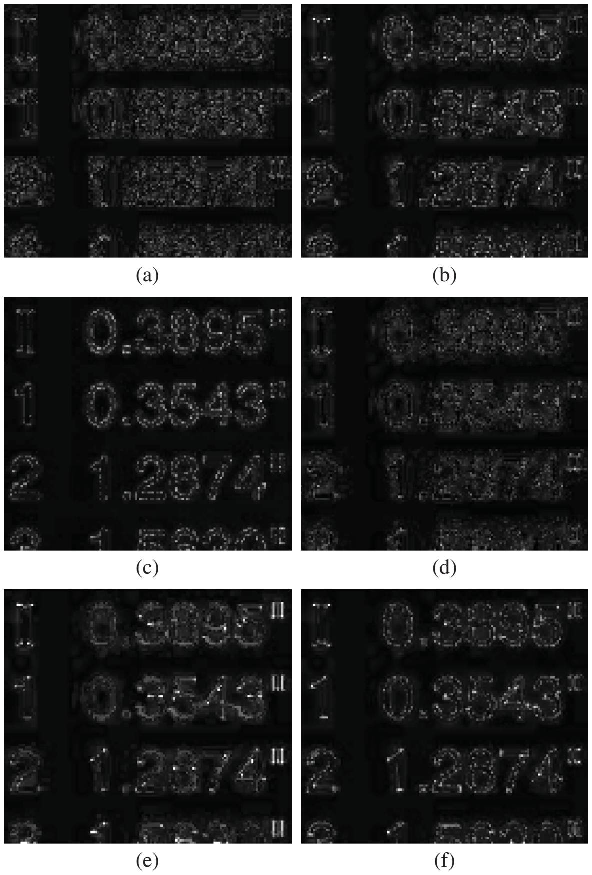


Fig. 8. Pixel-wise RMSE, scaled by 3, for Region 1. (a) JPEG. (b) BF. (c) BR. (d) RS. (e) IRS. (f) HSF.

(Fig. 6(a) and (b)), graphics (Fig. 6(c) and (d)), natural image overlaid with text (Fig. 6(e)), and natural image (Fig. 6(f)). Results of the schemes for these six regions are illustrated in Figs. 7–18.

Fig. 7 and Fig. 9 compare the results for two different text regions from Image I. To visualize the noise remaining in the results, we also form the error images to depict the pixel-wise root mean square error (RMSE) of the schemes for these two regions. The error images are shown in Fig. 8 for Region 1 and Fig. 10 for Region 2. Because the text regions contain many high-contrast sharp edges, the regions decoded by JPEG (Fig. 7(a) and Fig. 9(a)) are severely distorted by ringing artifacts. BF (Fig. 7(b) and Fig. 9(b)) slightly reduces some of the ringing artifacts around the text. BR (Fig. 7(c) and Fig. 9(c)) uses a specific prior model to decode the text and effectively removes most of the ringing artifacts. The results of RS (Fig. 7(d) and Fig. 9(d)) are slightly better than the results of BF, but they still contain a fair amount of ringing artifacts. However, most of the ringing artifacts remaining in the results of RS are successfully removed by the median filter employed by IRS (Fig. 7(e) and Fig. 9(e)). The results of HSF (Fig. 7(f) and Fig. 9(f)) 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. 7(c) has some isolated dark pixels in the background regions whereas the output of HSF in Fig. 7(f) does not. On the other hand, the output of BR is sharper than the output of HSF. In Fig. 8 and Fig. 10, a scheme that is more effective in reducing ringing artifacts shows lower noise around the sharp edges.

Fig. 11 and Fig. 13 compare the results for two different graphics regions from Image I. The corresponding error images are shown in Fig. 12 for Region 3 and Fig. 14 for Region 4. Similar to text, graphic art contains sharp edges and leads to severe ringing artifacts in the JPEG-decoded images (Fig. 11(a) and Fig. 13(a)). Both IRS (Fig. 11(e) and Fig. 13(e)) and HSF (Fig. 11(f) and Fig. 13(f)) 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 Region 3 (Fig. 11(e)), whereas HSF preserves the texture well (Fig. 11(f)). BR is similarly effective in removing the ringing artifacts in Region 3 (Fig. 11(c)); 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 4 (Fig. 13(c)). 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. 15 compares the results for a region with mixed content from Image I. The region contains a natural image overlaid with text. The corresponding error images are shown in Fig. 16. In the JPEG-decoded image of Fig. 15(a), the

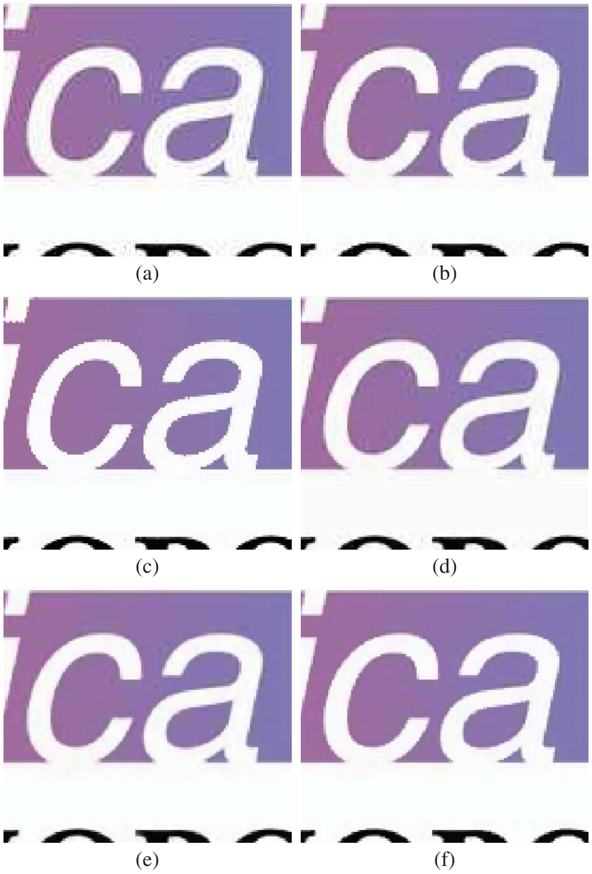


Fig. 9. Decoding results for Region 2. (a) JPEG. (b) BF. (c) BR. (d) RS. (e) IRS. (f) HSF.

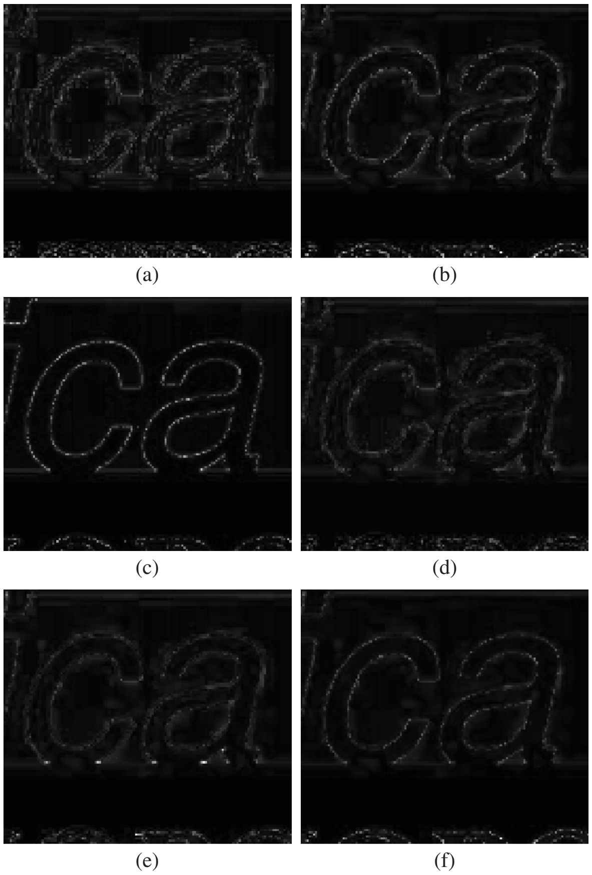


Fig. 10. Pixel-wise RMSE, scaled by 3, for Region 2. (a) JPEG. (b) BF. (c) BR. (d) RS. (e) IRS. (f) HSF.

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. 15(c). 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. 15(e) and HSF in Fig. 15(f) are significantly better than BR and slightly better than BF (Fig. 15(c)) and RS (Fig. 15(e)).

Fig. 17 compares the results for a picture region from Image II. The error images are shown in Fig. 18. This region is mainly corrupted by the blocking artifacts of JPEG (Fig. 17(a)). 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.

Table III summarizes the numerical results of the schemes based on a set of 22 test images. The test set includes the two images in Fig. 5, as well as another 20 images extracted from different sources, like magazines and product catalogues. Since we are primarily interested in document images, each test image generally contains both text and natural images. Also, three test images consist of natural images overlaid with large regions of text. The sizes of the images range from .7 megapixels (MP) to 7.5 MP, with the average size equals to 2.2 MP. We use the IJG JPEG encoder [43] to encode all the images with average bitrate .72 bpp. In the first three columns of Table III, we summarize the peak signal-to-noise ratio (PSNR), the root-mean-square error (RMSE), and the standard error of the RMSE of the different schemes. The standard error is computed according to Section II-C of [45] to estimate the standard deviation of the RMSE. Since the quality of a natural image is heavily affected by the blocking artifacts, we also manually segment the natural images from the test set, and measure the amount of blockiness in the processed natural images, using the scheme proposed in [46]. The blockiness measure assumes that a 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 the discrete Fourier transform. We summarize the blockiness measure of the segmented images processed by the different schemes in the last column of Table III. Here, a larger blockiness measure suggests that the images contain more blocking artifacts.

TABLE III

PSNR, RMSE, STANDARD ERROR OF RMSE [45], AND BLOCKINESS MEASURE [46] OF THE SCHEMES COMPUTED ON 22 TEST IMAGES. SMALLER VALUE IN THE LAST COLUMN SUGGESTS A SMALLER AMOUNT OF BLOCKING ARTIFACTS

Scheme	PSNR (dB)	RMSE	Std. Err. of RMSE [45]	Blockiness Measure [46]
JPEG	30.34	7.75	8×10^{-4}	0.8506
BF	30.53	7.59	8×10^{-4}	0.6852
BR	30.93	7.24	7×10^{-4}	0.8993
RS	30.74	7.41	8×10^{-4}	0.4228
IRS	30.42	7.68	8×10^{-4}	0.3326
HSF	31.35	6.90	7×10^{-4}	0.4150

TABLE IV

PSNR, RMSE, STANDARD ERROR OF RMSE [45], AND BLOCKINESS MEASURE [46] OF THE HSF AFTER ADDING THE FIFTH FILTER COMPUTED ON 22 TEST IMAGES. SMALLER VALUE IN THE LAST COLUMN SUGGESTS A SMALLER AMOUNT OF BLOCKING ARTIFACTS

Options of the Fifth Filter for the HSF	PSNR (dB)	RMSE	Std. Err. of RMSE [45]	Blockiness Measure [46]
Option 0: None	31.35	6.90	7×10^{-4}	0.4150
Option 1: Bilateral Filter, $\sigma_d = 0.3, \sigma_r = 10$	31.44	6.83	7×10^{-4}	0.4056
Option 2: Bilateral Filter, $\sigma_d = 0.7, \sigma_r = 10$	31.62	6.69	7×10^{-4}	0.4130
Option 3: Gaussian Filter, $\sigma = 0.5$	31.56	6.74	7×10^{-4}	0.4168
Option 4: Gaussian Filter, $\sigma = 1.5$	31.26	6.97	7×10^{-4}	0.4189

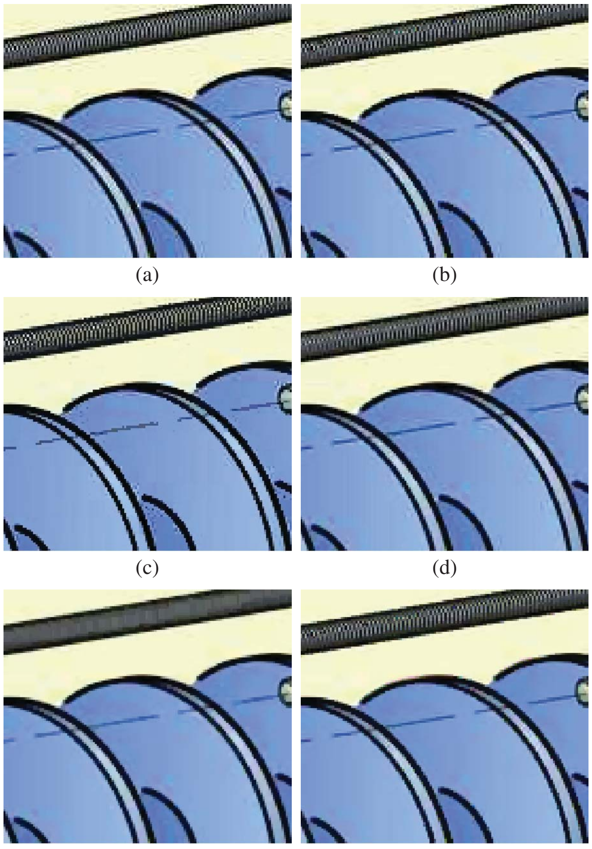


Fig. 11. Decoding results for Region 3. (a) JPEG. (b) BF. (c) BR. (d) RS. (e) IRS. (f) HSF.

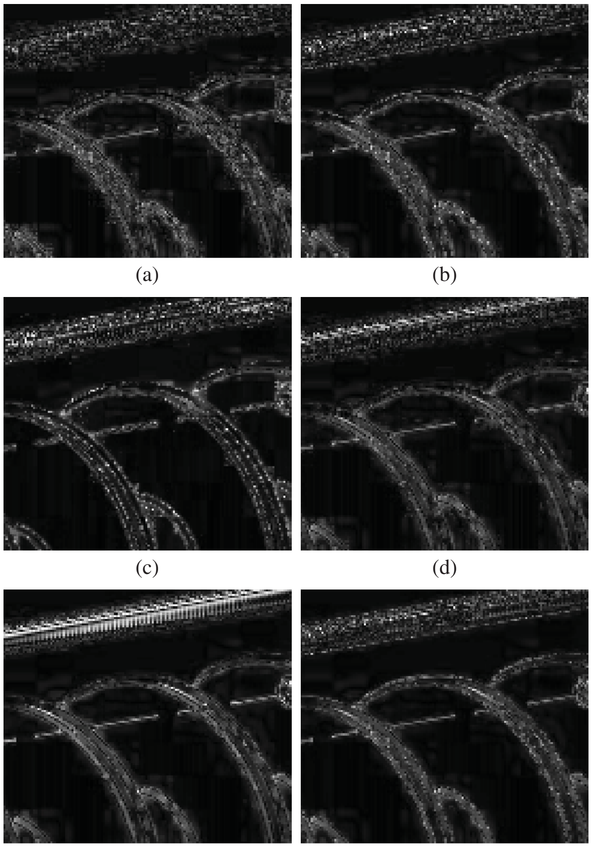


Fig. 12. Pixel-wise RMSE, scaled by 3, for Region 3. (a) JPEG. (b) BF. (c) BR. (d) RS. (e) IRS. (f) HSF.

From the first column of Table III, HSF achieves the highest PSNR, which is followed by BR, RS, BF, and IRS, while JPEG has the lowest PSNR. BR employs a specificially designed text model that is very effective in reducing the ringing artifacts in the text regions, which explains BR's relatively good performance. However, for image blocks that

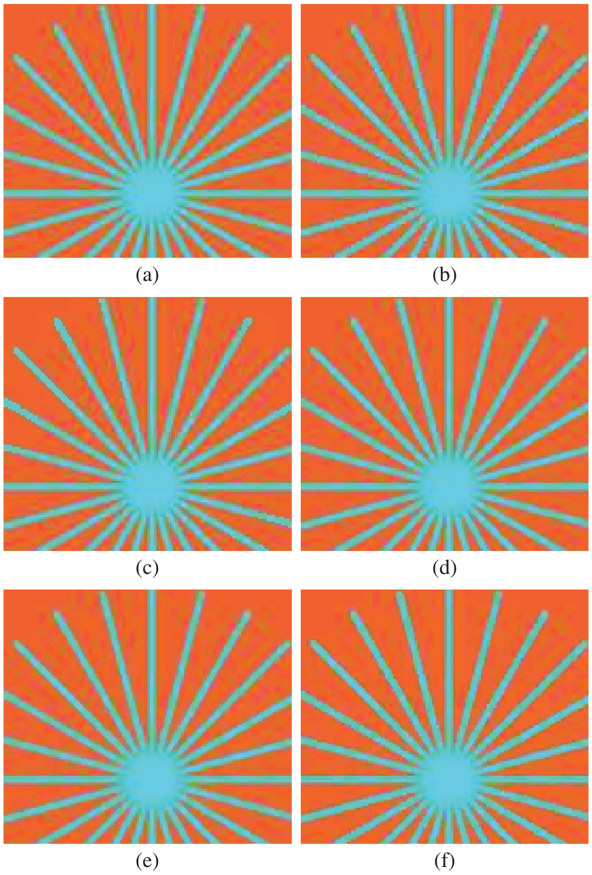


Fig. 13. Decoding results for Region 4. (a) JPEG. (b) BF. (c) BR. (d) RS. (e) IRS. (f) HSF.

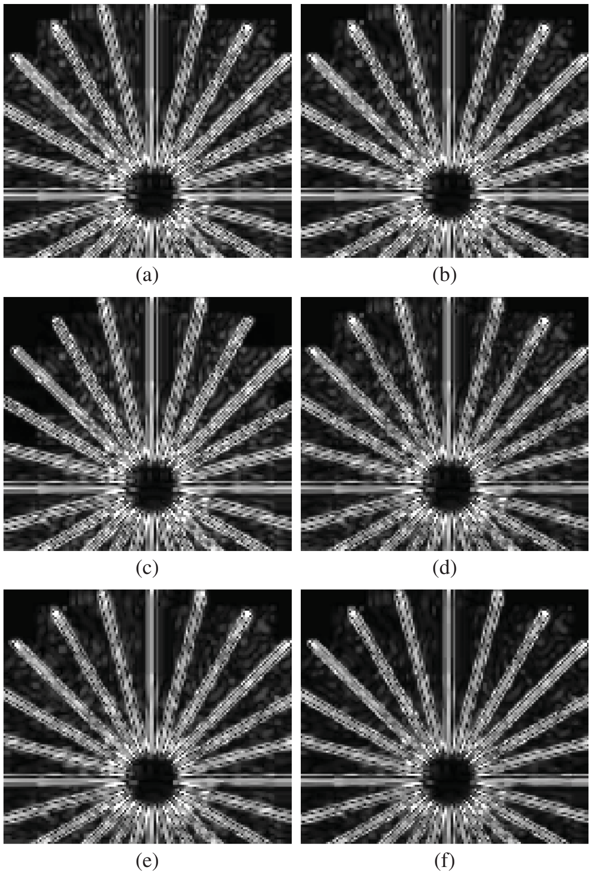


Fig. 14. Pixel-wise RMSE, scaled by 3, for Region 4. (a) JPEG. (b) BF. (c) BR. (d) RS. (e) IRS. (f) HSF.

are classified as picture blocks by BR's classifier, BR simply returns the block as decoded by the JPEG decoder and makes no attempt to improve the quality of the block. BR also suffers from a robustness issue. If a picture block is misclassified as a text block, BR usually leads to unnatural artifacts in the decoded block. These factors negatively affect BR's overall PSNR performance. RS is generally ineffective in removing the ringing artifacts, primarily because it is restricted to using only linear filters. Although IRS removes the ringing artifacts in the text regions satisfactorily, the application of the median filter to the whole image tends to cause more severe loss of image details (Fig. 13(e)) and artifacts (Fig. 17(e)). Thus, both RS and IRS have relatively lower PSNR. From Fig. 7(f)–Fig. 16(f), we have seen that filter 3 and filter 4 of HSF, the estimators of text foreground and text background intensities, are also effective in reducing the ringing artifacts around text. The text regions processed by the HSF has quality matching that of BR. Further, HSF applies the bilateral filter and the Gaussian filter adaptively to remove the JPEG artifacts in natural image regions without significant loss of image details. Thus, HSF is able to achieve the highest overall PSNR.

According to the last column of Table III, IRS has the lowest (best) blockiness measure, which is followed closely by HSF and RS. BF, JPEG, and BR have substantially inferior blockiness measures. IRS achieves better blockiness measure because of its median filter which further suppresses the blocking artifacts after the RS processing, but at the cost of blurring out image details, as evident from Fig. 13(e) and Fig. 17(e) and from the PSNR comparisons. Since BR simply uses the JPEG decoding algorithm to decode the image blocks in natural images, it has high blockiness measure despite the fact that it achieves the second best PSNR for the test set. HSF achieves the second best blockiness measure. The de-blocking performance of HSF is close to that of IRS, but it generally retains the image details better with higher PSNR.

Visual comparisons made in Fig. 7–Fig. 18 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. 13(c) and Fig. 15(c). There are at least two sources of this non-robustness. First, artifacts such as those in Fig. 15(c) 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. 13(c). Further, as seen from Fig. 7–Fig. 18 and Table III, BR is not very effective in reducing the blocking artifacts. IRS also suffers from robustness

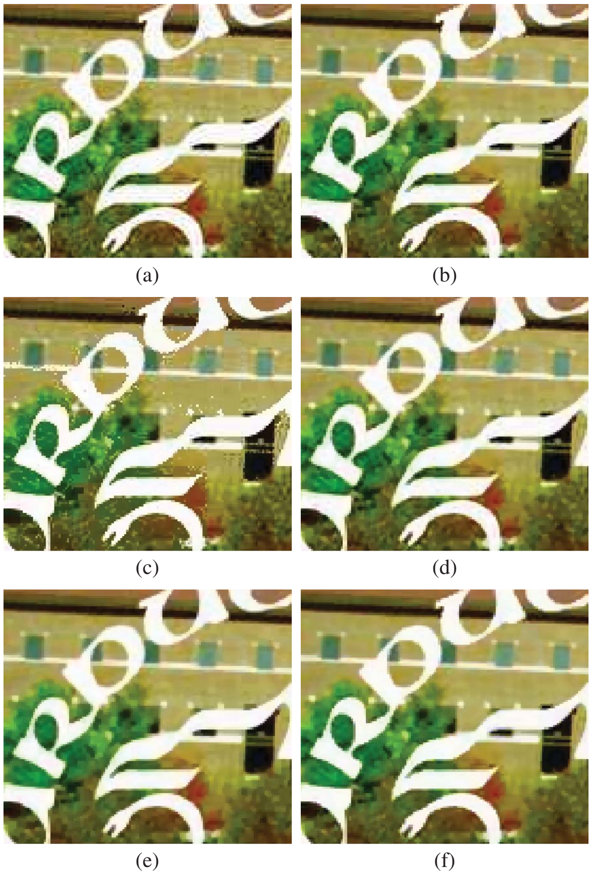


Fig. 15. Decoding results for Region 5. (a) JPEG. (b) BF. (c) BR. (d) RS. (e) IRS. (f) HSF.

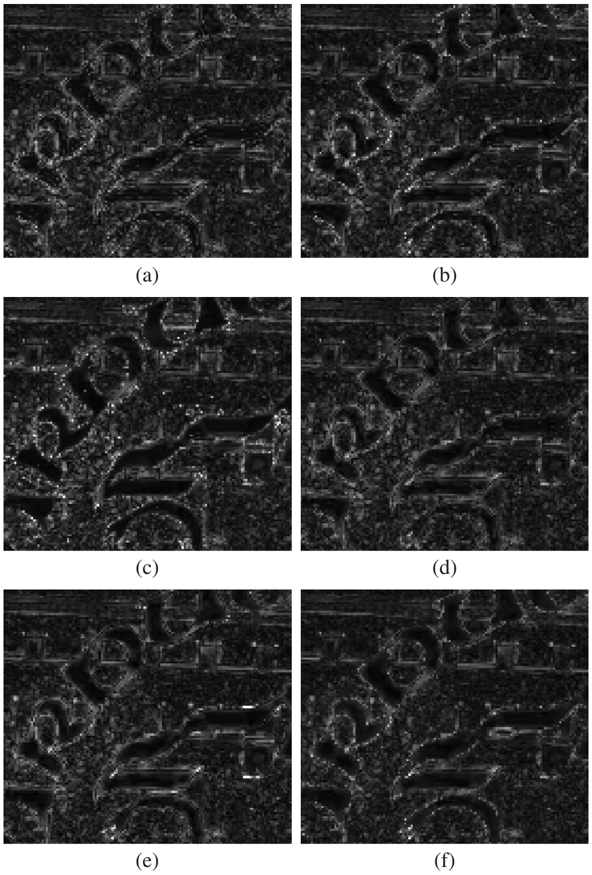


Fig. 16. Pixel-wise RMSE, scaled by 3, for Region 5. (a) JPEG. (b) BF. (c) BR. (d) RS. (e) IRS. (f) HSF.

issue in that its median filter causes over-blurring for certain highly texture image regions.

Finally, we also investigate whether adding an additional filter to the HSF will improve the performance further. We consider four different options for the fifth filter, which are summarized in Table IV as options 1–4, along with their PSNR, RMSE, standard error of RMSE, and blockiness measure for the 22 test images. For each option, the first four filters are the same as what we described in Section III-A. To facilitate comparison, we also repeat in Table IV the corresponding result of the HSF with four filters from Table III, as option 0. Option 1, option 2, and option 3 all result to a slight improvement in PSNR over the HSF with four filters. In option 4, we found that adding the fifth filter reduces the PSNR slightly. Options 1–2 have slightly lower (better) blockiness measures, while options 3–4 have slightly higher (worse) blockiness measures. Results of option 1–3 suggest that we can improve the performance by using more filters in the HSF, but even with the four filters we are proposing, the HSF is still outperforming the other state-of-the-art methods. For option 4, we found that the fifth filter, a Gaussian filter with $\sigma = 1.5$, usually produces an overly blurry output image and rarely generates better estimates of the original pixel values than the other filters. The training process of the HSF was able to alleviate this problem to some extend by reducing the class prior probability π_5 to a small value of 0.0078. However, the extra complexity due to the fifth filter makes the classifier perform less accurately, and hence leads to a lower PSNR.

V. 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 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 probabilistic model which leads to an unsupervised training procedure for the design of an optimal soft classifier. The resulting classifier distinguishes the different types of image content and appropriately adjusts the weighting factors of the image filters. 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 four 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

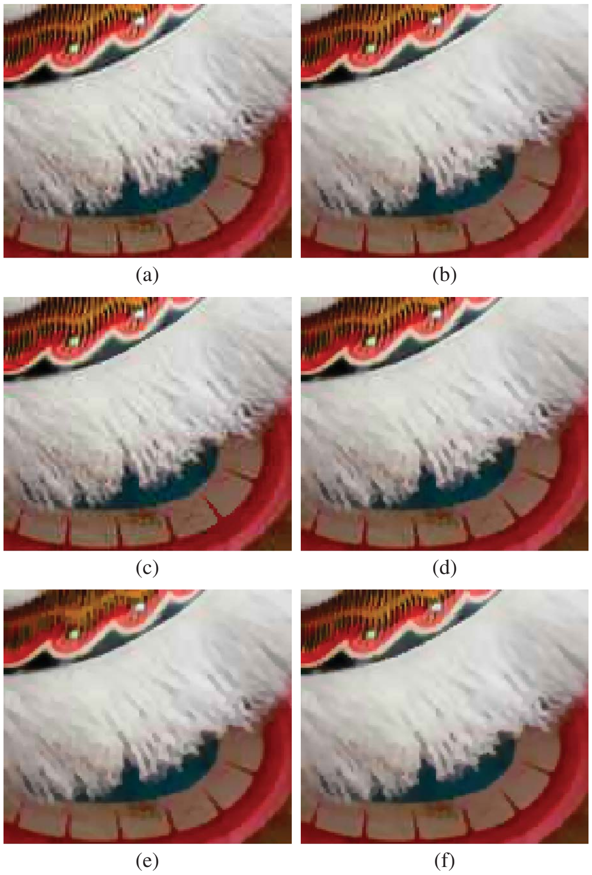


Fig. 17. Decoding results for Region 6. (a) JPEG. (b) BF. (c) BR. (d) RS. (e) IRS. (f) HSF.

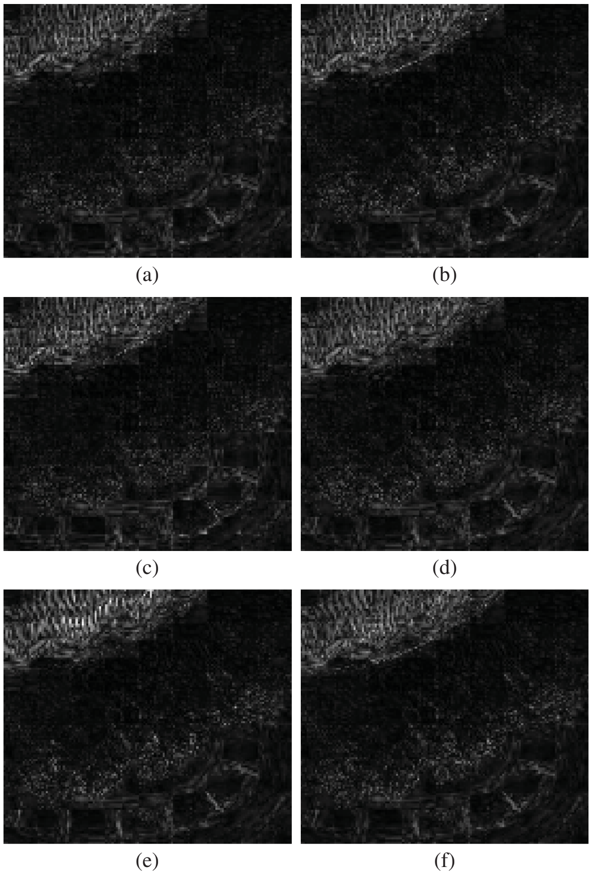


Fig. 18. Pixel-wise RMSE, scaled by 3, for Region 6. (a) JPEG. (b) BF. (c) BR. (d) RS. (e) IRS. (f) HSF.

method outperforms state-of-the-art JPEG decoding methods. Potential future research include exploring other options of more advanced image filters and features for the application of JPEG artifact reduction, as well as applying the scheme to other quality enhancement and image reconstruction tasks.

Appendix
Derivation of the EM Algorithm for the Training Procedure

We present the derivation of the EM algorithm [35]–[37] for estimating the two sets of parameters $\theta = \left\{\pi_j, \sigma_j^2\right\}_j$ in (1) and $\psi = \left\{\zeta_{j,l}, \mathbf{m}_{j,l}, \mathbf{R}_{j,l}\right\}_{j,l}$ in (3). To simplify notation, we define $\mathbf{W}_n = \left[X_n, \mathbf{Y}_n^t, \mathbf{Z}_n^t\right]^t$ to denote the concatenation of the observable random quantities X_n, $\mathbf{Y}_n$, and $\mathbf{Z}_n$. We use the lower case $\mathbf{w}_n = \left[x_n, \mathbf{y}_n^t, \mathbf{z}_n^t\right]^t$ to represent a realization of $\mathbf{W}_n$. The ML estimates of θ and ψ are defined by

$$\hat{\theta}_{ML},\ \hat{\psi}_{ML} = \arg\max_{\theta,\psi} \prod_n f_{\mathbf{W}_n}\left(\mathbf{w}_n|\theta, \psi\right). \tag{20}$$

Direct computation of $\hat{\theta}_{ML}$ and $\hat{\psi}_{ML}$ based on (20) is difficult because $f_{\mathbf{W}_n}\left(\mathbf{w}_n|\theta, \psi\right)$ must be obtained from marginalizing $f_{\mathbf{W}_n,J_n,L_n}\left(\mathbf{w}_n, j_n, l_n|\theta, \psi\right)$ since both J_n and L_n are unobserved. This is called the incomplete data problem, which can be solved by the EM algorithm.

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

$$\theta^{(k+1)}, \psi^{(k+1)} = \arg\max_{\theta,\phi}\ Q\big(\theta, \psi|\theta^{(k)}, \psi^{(k)}\big) \tag{21}$$

where the function Q is defined as

$$\begin{gathered} Q\big(\theta, \psi|\theta^{(k)}, \psi^{(k)}\big) = \textstyle\sum_n \\ E\big[\log f_{\mathbf{W}_n,J_n,L_n}(\mathbf{w}_n, J_n, L_n|\theta, \psi)\big|\mathbf{W}_n{=}\mathbf{w}_n, \theta^{(k)}, \psi^{(k)}\big]. \end{gathered} \tag{22}$$

By applying the iteration in (21), it had been proved that the resulting likelihood sequence is monotonic increasing [36], i.e.

$$\prod_n f_{\mathbf{W}_n}\left(\mathbf{w}_n|\theta^{(k+1)}, \psi^{(k+1)}\right) \geq \prod_n f_{\mathbf{W}_n}\left(\mathbf{w}_n|\theta^{(k)}, \psi^{(k)}\right)$$

for all k. Also, if the likelihood function is upper bound, the likelihood sequence converges to a local maximum [36].

If we apply the chain rule to $f_{\mathbf{W}_n,J_n,L_n}\left(\mathbf{w}_n, J_n, L_n|\theta, \psi\right)$ and use Assumption 3 in Section II-A that $\mathbf{Y}_n$ and L_n are conditionally independent of X_n and $\mathbf{Z}_n$ given J_n, we have

$$\begin{aligned} &\log f_{X_n,\mathbf{Y}_n,\mathbf{Z}_n,J_n,L_n}\left(x_n, \mathbf{y}_n, \mathbf{z}_n, J_n, L_n|\theta, \psi\right) \\ &\quad = \log f_{\mathbf{Y}_n,L_n|J_n}\left(\mathbf{y}_n, L_n|J_n, \psi\right) \\ &\qquad + \log f_{X_n,J_n|\mathbf{Z}_n}\left(x_n, J_n|\mathbf{z}_n, \theta\right) + \log f_{\mathbf{Z}_n}\left(\mathbf{z}_n\right). \end{aligned} \tag{23}$$

Notice that in the first term on the right hand side of (23), the conditional distribution $f_{\mathbf{Y}_n,L_n|J_n}$ is independent of θ. In the second term, the conditional distribution $f_{X_n,J_n|\mathbf{Z}_n}$ is independent of ψ. In the last term, the marginal distribution $f_{\mathbf{Z}_n}(\mathbf{z}_n)$ is independent of both θ and ψ. Combining (22) and (23), we decompose the function Q as a sum of three terms

$$Q\big(\theta,\psi|\theta^{(k)},\psi^{(k)}\big)=Q_1\big(\psi|\theta^{(k)},\psi^{(k)}\big)+Q_2\big(\theta|\theta^{(k)}\big)+C \quad (24)$$

where

$$\begin{aligned} Q_1\big(\psi|\theta^{(k)},\psi^{(k)}\big) &= \textstyle\sum_n E\big[\log f_{\mathbf{Y}_n,L_n|J_n}(\mathbf{y}_n,L_n|J_n,\psi)\,\big| \\ &\qquad X_n{=}x_n,\mathbf{Y}_n{=}\mathbf{y}_n,\mathbf{Z}_n{=}\mathbf{z}_n,\theta^{(k)},\psi^{(k)}\big] \quad (25)\\ Q_2\big(\theta|\theta^{(k)}\big) &= \textstyle\sum_n E\big[\log f_{X_n,J_n|\mathbf{Z}_n}(x_n,J_n|\mathbf{z}_n,\theta)\,\big| \\ &\qquad X_n{=}x_n,\mathbf{Z}_n{=}\mathbf{z}_n,\theta^{(k)}\big] \quad (26)\\ C &= \textstyle\sum_n E\big[\log f_{\mathbf{Z}_n}(\mathbf{z}_n)\,\big|\mathbf{Z}_n{=}\mathbf{z}_n\big]. \quad (27)\end{aligned}$$

The function Q_2 takes the form of (26) due to Assumption 4 in Section II-A that J_n is conditionally independent of $\mathbf{Y}_n$ given X_n and $\mathbf{Z}_n$. Further, we omit $\psi^{(k)}$ in the conditioning of the expectation because $f_{J_n|X_n,\mathbf{Z}_n}$ is independent of $\psi^{(k)}$.

The function Q written in the form of (24) allows us to simplify the EM algorithm into a simpler two-stage procedure. During the optimization of Q, we may ignore C because it is independent of both θ and ψ. 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

$$\begin{aligned}\theta^{(i+1)} &= \arg\max_{\theta} Q_2\big(\theta|\theta^{(i)}\big) \quad (28)\\ \hat{\theta} &= \lim_{i\to\infty}\theta^{(i)}. \quad (29)\end{aligned}$$

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

$$\begin{aligned}\psi^{(k+1)} &= \arg\max_{\phi} Q_1\big(\psi|\hat{\theta},\psi^{(k)}\big) \quad (30)\\ \hat{\psi} &= \lim_{k\to\infty}\psi^{(k)}. \quad (31)\end{aligned}$$

The fact that the procedure in (28)–(31) is equivalent to (21) can be seen by considering the limit $\hat{\theta}$ in (29) as the initial value $\theta^{(0)}$ in (21). The update iteration (30) is then the same as (21) if we set $\hat{\theta}=\theta^{(k)}$ in (21) for all k.

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

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

where

$$\begin{aligned}N_j^{(i)} &= \textstyle\sum_n f_{J_n|X_n,\mathbf{Z}_n}\big(j|x_n,\mathbf{z}_n,\theta^{(i)}\big), \quad (32)\\ t_j^{(i)} &= \textstyle\sum_n (x_n-z_{n,j})^2 f_{J_n|X_n,\mathbf{Z}_n}\big(j|x_n,\mathbf{z}_n,\theta^{(i)}\big) \quad (33)\end{aligned}$$

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

$$\begin{aligned}\pi_j^{(i+1)} &= \frac{N_j^{(i)}}{N} \quad (34)\\ \Big[\sigma_j^{(i+1)}\Big]^2 &= \frac{t_j^{(i)}}{N_j^{(i)}}. \quad (35)\end{aligned}$$

The set of update formulas in (32)–(35) are equivalent to the updates in (10)–(13).

Similarly, we derive the update formulas for ψ by expanding the function Q_1 as

$$\begin{aligned}&Q_1\big(\psi|\hat{\theta},\psi^{(k)}\big)\\ &= \textstyle\sum_n E\big[\log f_{\mathbf{Y}_n,L_n|J_n}(\mathbf{y}_n,L_n|J_n,\psi)\,\big|\\ &\quad X_n{=}x_n,\mathbf{Y}_n{=}\mathbf{y}_n,\mathbf{Z}_n{=}\mathbf{z}_n,\hat{\theta},\psi^{(k)}\big]\\ &= \textstyle\sum_j\sum_l\sum_n f_{J_n,L_n|X_n,\mathbf{Y}_n,\mathbf{Z}_n}\big(j,l|x_n,\mathbf{y}_n,\mathbf{z}_n,\hat{\theta},\psi^{(k)}\big)\\ &\quad\times\log f_{\mathbf{Y}_n,L_n|J_n}(\mathbf{y}_n,l|j,\psi).\end{aligned}$$

We further expand $\log f_{\mathbf{Y}_n,L_n|J_n}(\mathbf{y}_n,l|j,\psi)$ as

$$\begin{aligned}&\log f_{\mathbf{Y}_n,L_n|J_n}(\mathbf{y}_n,l|j,\psi)\\ &= -\frac{1}{2}(\mathbf{y}_n-\mathbf{m}_{j,l})^t\,\mathbf{R}_{j,l}^{-1}(\mathbf{y}_n-\mathbf{m}_{j,l})\\ &\quad+\log\zeta_{j,l}-\frac{1}{2}\log\big|\mathbf{R}_{j,l}\big|-\frac{D}{2}\log(2\pi)\end{aligned}$$

which leads to the following form of the function Q_1

$$\begin{aligned}&Q_1\big(\psi|\hat{\theta},\psi^{(k)}\big)\\ &= \textstyle\sum_j\sum_l\Bigg\{N_{j,l}^{(k)}\Big[\log\zeta_{j,l}-\dfrac{1}{2}\log|\mathbf{R}_{j,l}|-\dfrac{1}{2}\mathbf{m}_{j,l}^t\mathbf{R}_{j,l}^{-1}\mathbf{m}_{j,l}\Big]\\ &\quad-\frac{1}{2}\,\mathrm{tr}\left(\mathbf{R}_{j,l}^{-1}\,\mathbf{v}_{j,l}^{(k)}\right)+\mathbf{m}_{j,l}^t\mathbf{R}_{j,l}^{-1}\mathbf{u}_{j,l}^{(k)}\Bigg\}-\frac{DN\log(2\pi)}{2}\end{aligned}$$

where

$$\begin{aligned}N_{j,l}^{(k)} &= \textstyle\sum_n f_{J_n,L_n|X_n,\mathbf{Y}_n,\mathbf{Z}_n}\big(j,l|x_n,\mathbf{y}_n,\mathbf{z}_n,\hat{\theta},\psi^{(k)}\big) \quad (36)\\ \mathbf{u}_{j,l}^{(k)} &= \textstyle\sum_n \mathbf{y}_n\, f_{J_n,L_n|X_n,\mathbf{Y}_n,\mathbf{Z}_n}\big(j,l|x_n,\mathbf{y}_n,\mathbf{z}_n,\hat{\theta},\psi^{(k)}\big) \quad (37)\\ \mathbf{v}_{j,l}^{(k)} &= \textstyle\sum_n \mathbf{y}_n\mathbf{y}_n^t\, f_{J_n,L_n|X_n,\mathbf{Y}_n,\mathbf{Z}_n}\big(j,l|x_n,\mathbf{y}_n,\mathbf{z}_n,\hat{\theta},\psi^{(k)}\big) \quad (38)\end{aligned}$$

D is the dimension of $\mathbf{y}_n$, and $\mathrm{tr}(\cdot)$ 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 $\mathbf{R}_{j,l}$ yields the following update formulas for $\zeta_{j,l}$, $\mathbf{m}_{j,l}$, and $\mathbf{R}_{j,l}$

$$\begin{aligned}\zeta_{j,l}^{(k+1)} &= \frac{N_{j,l}^{(k)}}{\hat{N}_j} \quad (39)\\ \mathbf{m}_{j,l}^{(k+1)} &= \frac{\mathbf{u}_{j,l}^{(k)}}{N_{j,l}^{(k)}} \quad (40)\\ \mathbf{R}_{j,l}^{(k+1)} &= \frac{\mathbf{v}_{j,l}^{(k)}}{N_{j,l}^{(k)}}-\frac{\mathbf{u}_{j,l}^{(k)}\big(\mathbf{u}_{j,l}^{(k)}\big)^t}{\big(N_{j,l}^{(k)}\big)^2}. \quad (41)\end{aligned}$$

The update formulas (36)–(41) are equivalent to (14)–(18).

References

[1] A. C. Bovik and S. T. Acton, "Basic linear filtering with application to image enhancement," in *Handbook of Image and Video Processing*. San Francisco, CA: Academic, 2000, pp. 71–80.

[2] J. W. Tukey, "Nonlinear (nonsuperposable) methods for smoothing data," in *Proc. EASCON Conf. Rec.*, 1974, pp. 673–681.

[3] J. W. Tukey, *Exploratory Data Analysis*. Reading, MA: Addison-Wesley, 1977.

[4] B. I. Justusson, "Median filtering: Statistical properties," in *Two-Dimensional Digital Signal Prcessing II* (Topics in Applied Physics), vol. 43. Berlin, Germany: Springer-Verlag, 1981, pp. 161–196.

[5] D. R. K. Brownrigg, "The weighted median filter," *Commun. ACM*, vol. 27, no. 8, pp. 807–818, Aug. 1984.

[6] S. J. Ko and Y. H. Lee, "Center weighted median filters and their applications to image enhancement," *IEEE Trans. Circuits Syst.*, vol. 38, no. 9, pp. 984–993, Sep. 1991.

[7] L. Yin, R. Yang, M. Gabbouj, and Y. Neuvo, "Weighted median filters: A tutorial," *IEEE Trans. Circuits Syst. II, Analog Digit. Signal Process.*, vol. 43, no. 3, pp. 157–192, Mar. 1996.

[8] A. C. Bovik, T. S. Huang, and D. C. Munson, "A generalization of median filtering using linear combinations of order statistics," *IEEE Trans. Acoust., Speech Signal Process.*, vol. 31, no. 6, pp. 1342–1350, Dec. 1983.

[9] P. Wendt, E. Coyle, and N. C. Gallagher, "Stack filters," *IEEE Trans. Acoust., Speech Signal Process.*, vol. 34, no. 4, pp. 898–911, Aug. 1986.

[10] M. Gabbouj, E. J. Coyle, and N. C. Gallagher, "An overview of median and stack filtering," *Circuits, Syst. Signal Process.*, vol. 11, no. 1, pp. 7–45, Jan. 1992.

[11] B. Zhang and J. P. Allebach, "Adaptive bilateral filter for sharpness enhancement and noise removal," *IEEE Trans. Image Process.*, vol. 17, no. 5, pp. 664–678, May 2008.

[12] C. Tomasi and R. Manduchi, "Bilateral filtering for gray and color images," in *Proc. 6th Int. Conf. Comput. Vis.*, Jan. 1998, pp. 839–846.

[13] S. M. Smith and J. M. Brady, "SUSAN–a new approach to low level image processing," *Int. J. Comput. Vis.*, vol. 23, no. 1, pp. 45–78, May 1997.

[14] P. A. Mlsna and J. J. Rodríguez, "Gradient and Laplacian-type edge detection," in *Handbook of Image and Video Processing*. San Francisco, CA: Academic, 2000, pp. 415–431.

[15] H. Hu and G. de Haan, "Classification-based hybrid filters for image processing," in *Proc. Visual Commun. Image Process. Conf.*, Jan. 2006, pp. 1–10.

[16] T. Kondo, Y. Fujimori, S. Ghosal, and J. J. Carrig, "Method and apparatus for adaptive filter tap selection according to a class," U.S. Patent 6 192 161, Feb. 20, 2001.

[17] A. Buades, B. Coll, and J.-M. Morel, "A non-local algorithm for image denoising," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2005, pp. 60–65.

[18] L. Shao, H. Zhang, and G. de Haan, "An overview and performance evaluation of classification-based least squares trained filters," *IEEE Trans. Image Process.*, vol. 17, no. 10, pp. 1772–1782, Oct. 2008.

[19] L. Shao, J. Wang, I. Kirenko, and G. de Haan, "Quality adaptive least squares trained filters for video compression artifacts removal using a no-reference block visibility metric," *J. Visual Commun. Image Rep.*, vol. 22, no. 1, pp. 23–32, Jan. 2011.

[20] T.-S. Wong, C. A. Bouman, J.-B. Thibault, and K. D. Sauer, "Medical image enhancement using resolution synthesis," in *Proc. Comput. Imag. Conf.*, Feb. 2011, pp. 787307-1–787307-6.

[21] S. G. Chang, B. Yu, and M. Vetterli, "Spatially adaptive wavelet thresholding with context modeling for image denoising," *IEEE Trans. Image Process.*, vol. 9, no. 9, pp. 1522–1531, Sep. 2000.

[22] C. B. Atkins, C. A. Bouman, and J. P. Allebach, "Optimal image scaling using pixel classification," in *Proc. Int. Conf. Image Process.*, Oct. 2001, pp. 864–867.

[23] J. V. Ouwerkerk, "Image super-resolution survey," *Image Vis. Comput.*, vol. 24, no. 10, pp. 1039–1052, 2006.

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

[25] H. Siddiqui and C. A. Bouman, "Hierarchical color correction for camera cell phone images," *IEEE Trans. Image Process.*, vol. 17, no. 11, pp. 2138–2155, Nov. 2008.

[26] B. Zhang, J. S. Gondek, M. T. Schramm, and J. P. Allebach, "Improved resolution synthesis for image interpolation," in *Proc. Int. Conf. Digit. Print. Technol.* Sep. 2006, pp. 343–345.

[27] A. Polesel, G. Ramponi, and V. J. Mathews, "Image enhancement via adaptive unsharp masking," *IEEE Trans. Image Process.*, vol. 9, no. 3, pp. 505–510, Mar. 2000.

[28] S. H. Kim and J. P. Allebach, "Optimal unsharp mask for image sharpening and noise removal," *J. Electron. Imag.*, vol. 14, no. 2, pp. 23005–23018, May 2005.

[29] J. A. Stark, "Adaptive image contrast enhancement using generalizations of histogram equalization," *IEEE Trans. Image Process.*, vol. 9, no. 5, pp. 889–896, May 2000.

[30] Z. Yu and C. Bajaj, "A fast and adaptive method for image contrast enhancement," in *Proc. Int. Conf. Image Process.*, Oct. 2004, pp. 1001–1004.

[31] A. Papoulis and S. U. Pillai, *Probability, Random Variables, and Stochastic Processes*, 3rd ed. New York: McGraw-Hill, 2002.

[32] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, 2nd ed. New York: Springer-Verlag, 2009, ch. 6, pp. 214–215.

[33] M. I. Jordan, Z. Ghahramani, T. S. Jaakkola, and L. K. Saul, "An introduction to variational methods for graphical models," *Mach. Learn.*, vol. 37, pp. 183–233, Nov. 1999.

[34] R. J. Larsen and M. L. Marx, *An Introduction to Mathematical Statistics and Its Applications*, 5th ed. Englewood Cliffs, NJ: Prentice-Hall, 2011.

[35] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *J. Royal Stat. Soc., Ser. B, Methodol.*, vol. 39, no. 1, pp. 1–38, 1977.

[36] C. Wu, "On the convergence properties of the EM algorithm," *Ann. Stat.*, vol. 11, no. 1, pp. 95–103, Mar. 1983.

[37] R. A. Boyles, "On the convergence of the EM algorithm," *J. Royal Stat. Soc., Ser. B, Methodol.*, vol. 45, no. 1, pp. 47–50, 1983.

[38] J. Rissanen, "A universal prior for integers and estimation by minimum description length," *Ann. Stat.*, vol. 11, no. 2, pp. 416–431, Jun. 1983.

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

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

[41] A. K. Jain, *Fundamentals of Digital Image Processing*, 1st ed. Englewood Cliffs, NJ: Prentice-Hall, 1989, ch. 9, pp. 347–357.

[42] R. Franzen. (2002). *Kodak Lossless True Color Image Suite* [Online]. Available: http://http://r0k.us/graphics/kodak/

[43] *Independent JPEG Group Website*. (2012) [Online]. Available: http://www.ijg.org/

[44] 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, no. 11, pp. 2518–2535, Nov. 2009.

[45] S. Ahn and J. A. Fessler, "Standard errors of mean, variance, and standard deviation estimators," Dept. Electr. Eng. Comput. Sci., Dep. Univ. Michigan, Ann Arbor, Tech. Rep. TR 413, Jul. 2003.

[46] Z. Wang, A. C. Bovik, and B. L. Evans, "Blind measurement of blocking artifacts in images," in *Proc. Int. Conf. Image Process.*, Sep. 2000, pp. 981–984.

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, Hong Kong, in 1997 and 2000, respectively, and the Ph.D. degree in electrical engineering from Purdue University, West Lafayette, IN, in 2011.

He has been a Research Engineer with Sony Electronics, San Jose, CA, from 2011. His current research interests include statistical image processing, image segmentation, document image analysis and processing, image denoising, and camera pipeline signal processing.

Charles A. Bouman received the B.S. degree from the University of Pennsylvania, Philadelphia, and the M.S. degree from the University of California at Berkeley in 1981 and 1982, respectively, and the Ph.D. degree from Princeton University, Princeton, NJ, in 1989, all in electrical engineering.

He was a Full Staff Member with MIT Lincoln Laboratory, Lexington, MA, from 1982 to 1985. He joined Purdue University, West Lafayette, IN, in 1989, where he is currently the Michael J. and Katherine R. Birck Professor of electrical and computer engineering, in a courtesy appointment with the School of Biomedical Engineering, and is a Co-Director of the Magnetic Resonance Imaging Facility, Purdue's Research Park. His current research interests include the use of statistical image models, multiscale techniques, and fast algorithms in applications including tomographic reconstruction, medical imaging, and document rendering and acquisition.

Prof. Bouman was a recipient of the IS&Ts Raymond C. Bowman Award for outstanding contributions to digital imaging education and research, the College of Engineering Engagement/Service Award, and Team Award. He is a fellow of the American Institute for Medical and Biological Engineering, the Society for Imaging Science and Technology (IS&T), and the SPIE Professional Society. He was the Editor-in-Chief of the IEEE TRANSACTIONS ON IMAGE PROCESSING and a Distinguished Lecturer for the IEEE Signal Processing Society. He is currently a member of the Board of Governors, an Associate Editor of the IEEE TRANSACTIONS ON IMAGE PROCESSING and the IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE. He is a Co-Chair of the SPIE/IS&T Symposium on Electronic Imaging in 2006, a Co-Chair of the SPIE/IS&T conferences on Visual Communications and Image Processing 2000, the Vice President of Publications and a member of the Board of Directors for the IS&T Society, the Founder and a Co-Chair of the SPIE/IS&T Conference on Computational Imaging and a Purdue University Faculty Scholar.

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

He was a post-doctoral researcher at the Division of Applied Mathematics, Brown University, Providence, RI, in 1999 to 2000. Since 2000, he has been with Purdue University, West Lafayette, IN, where he is currently Associate Professor of Electrical and Computer Engineering. His research interests are in signal and image processing and financial engineering.

Dr. Pollak received a CAREER award from the National Science Foundation in 2001. He received Eta Kappa Nu Outstanding Faculty Award in 2002 and in 2007, and Chicago-Area Alumni Young Faculty Award in 2003. He was Lead Guest Editor of the IEEE Signal Processing Magazine Special Issue on Signal Processing for Financial Applications which appeared in September 2011. He is Co-Chair of the SPIE/IS&T Conference on Computational Imaging.