

Joint Demosaicing and Denoising

Keigo Hirakawa, and Thomas W. Parks, *Fellow, IEEE*

ABSTRACT

The output image of a digital camera is subject to a severe degradation due to noise in the image sensor. This paper proposes a novel technique to combine demosaicing and denoising procedures systematically into a single operation by exploiting their obvious similarities. We first design a filter *as if* we are optimally estimating a pixel value from a noisy single-color (sensor) image. With additional constraints, we show that the same filter coefficients are appropriate for CFA interpolation (demosaicing) given noisy sensor data. The proposed technique can combine many existing denoising algorithms with the demosaicing operation. In this paper, a total least squares denoising method is used to demonstrate the concept. The algorithm is tested on color images with pseudo-random noise and on raw sensor data from a real CMOS digital camera that we calibrated. The experimental results confirm that the proposed method suppresses noise (CMOS/CCD image sensor noise model) while effectively interpolating the missing pixel components, demonstrating a significant improvement in image quality when compared to treating demosaicing and denoising problems independently.

KEY WORDS

Image denoising, demosaicing, CFA interpolation, image sensor, digital camera.

I. INTRODUCTION

A typical digital camera is subject to influences from noise in the image sensor. This sensor noise, often characterized as signal-dependent noise, is amplified by a series of image processing steps needed to produce a full-color representation of an image displayable on a computer monitor or a printer. This phenomenon is especially evident when taking a picture in a low-light environment, and it is one of the major problems noticeable in commercial digital cameras today.

A cost-effective digital camera uses a single-chip image sensor, with alternating patterns of three or more color filters (whose sensitivity are spectrally shifted) applied to each pixel location (called color filter array, or CFA). Often these filters are red, green, and blue, although there are other choices of primary colors as well. A popular arrangement of primary RGB is Bayer pattern [2], which is shown in figure 4. A method for

Manuscript received January ?, 2005; revised Month Day, Year.

K. Hirakawa is with New England Conservatory of Music. During the development of this work, KH was on leave from Cornell University. Email: kh237@cornell.edu. T.W. Parks is with Cornell University. Email: parks@ece.cornell.edu. The work contained herein was supported by Texas Instruments.

reconstructing a full-color image by estimating the missing pixel components from the sensor data is called demosaicing (a.k.a. CFA interpolation). The source of conflict between the image processing pipeline and image sensor noise is this demosaicing step. Often in demosaicing, we would like to preserve the sharpness of the edges while interpolating the missing pixel components. In the presence of noise, however, noise patterns form false edge structures, sharpening amplifies high frequency noise, and interpolation adds a structure to the noise too complicated to analyze mathematically. While there are gray-scale and color image denoising techniques have been suggested [25] [12] [24] [21] [22] [28] [14], removing noise after demosaicing, therefore, is impractical.

Many demosaicing algorithms and related works have been published in recent years [8] [9] [16] [29] [17] [18] [19] [20] [26] [4] [23] [1]. Although the output images from these algorithms are impressive in the absence of sensor noise, none of them, including [26], address the image sensor noise problem explicitly (to the best of the knowledge of the authors). It should be noted that the method in [26] implicitly incorporates denoising by adapting a bilateral filtering structure [28]. However, the smoothing is applied to missing pixel components only, and the measured pixel components still remain noisy.

In recent years, a considerable amount of work has been done on denoising an image corrupted by signal-independent noise. While they are useful general methods, most algorithms neither take into account signal-dependent noise models (which is observed in CMOS and CCD sensors) nor accommodate denoising of a Bayer (or other CFA) pattern image. Generalizing these algorithms to removing noise before the image processing pipeline is problematic because determining an image structure, necessary for effective noise reduction, from a sparse sampling lattice is difficult.

Noting that image interpolation and image denoising are both estimation problems, this paper proposes a unified approach to performing demosaicing and image denoising simultaneously. The novelty of our work is the development of a *constraint*, under which an optimal filter for estimating a pixel value from a noisy *single-color* image is also an optimal filter for demosaicing given noisy sensor data. Furthermore, many existing image denoising algorithms can be combined with the demosaicing operation using this proposed technique because this constraint is not very restrictive. For example, one may choose bilateral filtering (a method to linearly combine nearby pixels to smooth out the noise) because of computational efficiency [28], while another may choose a more sophisticated image denoising method for higher image quality.

Performing demosaicing and image denoising simultaneously has various advantages over treating these problems separately. First, image quality is improved because we can tune the filter coefficients such that the edge structures are preserved without amplifying noise. Second, the estimation of the missing pixel components may explicitly incorporate the noise characteristics of the sensor. Finally, the combined algorithm reduces the computational complexity compared to performing these procedures separately.

This paper is organized as follows. Section II characterizes and analyzes the noise seen in CMOS and CCD image sensors, and a mathematical model for the noise is given. Section III provides motivation for combining denoising and demosaicing methods by posing the problem as a filter design. In section IV, an example demosaicing-denoising method is developed using the proposed technique, and section V presents experimental results using pseudo-random

and real sensor noise. We conclude in section VI. We emphasize that the algorithm in section IV is intended as an example *only*, and the main goal of this paper is to establish a general framework for combining demosaicing with denoising. This framework motivates a powerful and flexible technique for developing new demosaicing-denoising algorithms to fit the computational and/or image quality needs of various applications.

II. CMOS AND CCD SENSOR NOISE

A. Noise Characterization

In order for the denoising method to be effective, it is important to understand the noise characteristics in an image sensor. The CMOS photodiode active pixel sensor (APS) typically uses a photodiode and three transistors, all major sources of noise [3] [27]. The CCD sensors rely on the property of the material that when a photon strikes silicon, an electron-hole pair is generated [10]. While investigating the source of noise is beyond the scope of this paper, studies suggest that the readout noise (for both CMOS and CCD sensors) takes the following general form [27] [10]:

$$Y(i, j) = X(i, j) + (k_0 + k_1 X(i, j))\delta(i, j), \quad (1)$$

where $X(i, j)$ and $Y(i, j)$ are the ideal and measured sensor values at pixel location (i, j) , respectively, $\delta(i, j) \sim \mathcal{N}(0, 1)$ is noise, and $k_0, k_1 \in \mathbb{R}$ are sensor dependent parameters.

We independently verified the relationship in (1) for CMOS sensors by calibrating Agilent Technologies camera evaluation board HDCP-2000, equipped with a 300K pixel CMOS sensor [3]. Inside a room with controlled lighting, the Macbeth color chart is placed in the view of the camera in a fixed position. Assuming that the expectation $\mathcal{E}\{Y\}$ is X and that the colors inside the squares on the color chart are uniform, the average and the variance of 400 points measured from one square are taken to be the true X value and the noise variance for that X value, respectively. We repeat this experiment with varying levels of lighting to measure the noise variances for many different X values. The programmable gain amplifier (PGA) is held constant throughout the calibration experiments. All images are captured in unprocessed raw sensor data format. We assumed that the variation among the pixel sensors is small compared to the level of noise.

In figure 1, the standard deviation of the sensor values is plotted against the estimated X value. Red pixels, blue pixels, and odd indexed and even indexed green pixels are plotted separately. It is clear from these graphs that the standard deviation of the noise and the pixel values are roughly related by an affine equation, as in (1). Moreover, the histogram of the 400 data points measured from the same square in Macbeth color chart is shown in figure 2. It reveals that for each X , it is not unreasonable to call the shape of the noise distribution Gaussian.

B. Perceived Noise

Although the level of the noise is greater in bright regions of the image (due to $k_1 > 0$), the dark regions of an image are more difficult to denoise because the signal-to-noise ratio is smaller when the signal value is small (due

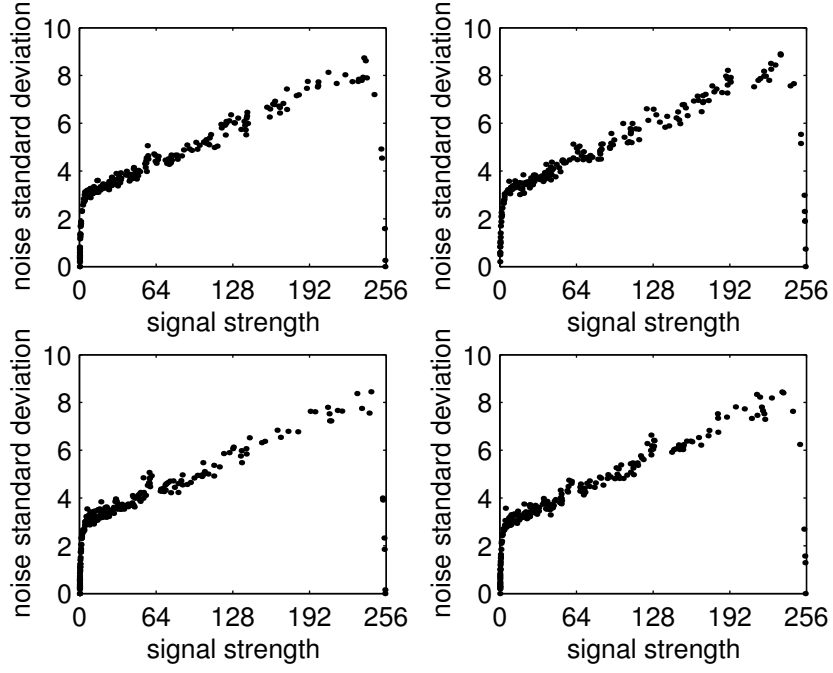


Fig. 1. Graphs of standard deviation of noise v.s. image sensor value (for a fixed PGA). From top-to-bottom, left-to-right, green sensors, red sensors, blue sensors, green sensors.

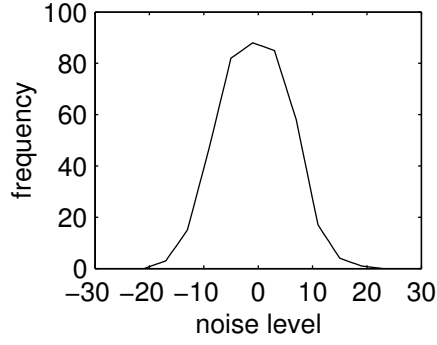


Fig. 2. The distribution of noise when $X = 160$.

to $k_0 > 0$). Furthermore, the signal-dependent noise model (1) means that the dark regions of the image appears most noisy to a human eye. To understand this, consider the definition of CIE-Lab color space [13]:

$$\begin{aligned} S_L &= 116(S_Y/T_Y)^{1/3} - 16 \\ S_a &= 500((S_X/T_X)^{1/3} - (S_Y/T_Y)^{1/3}) \\ S_b &= 200((S_Y/T_Y)^{1/3} - (S_Z/T_Z)^{1/3}), \end{aligned} \tag{2}$$

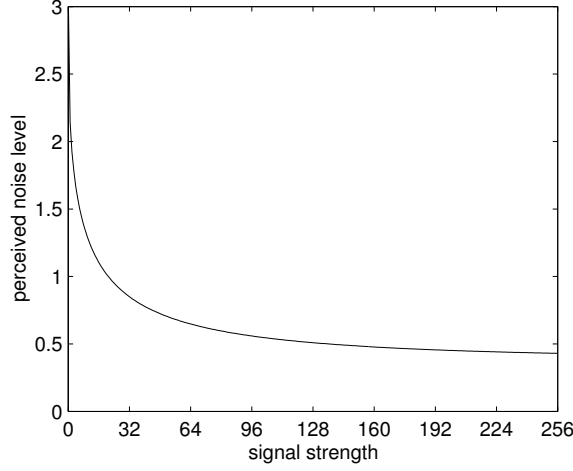


Fig. 3. $(q + (k_0 + k_1q))^{1/3} - q^{1/3}$ plotted against q . See text.

where $[S_X, S_Y, S_Z]$ are tri-stimulus values in XYZ color space, and $[T_X, T_Y, T_Z]$ are the reference white color, measured also in XYZ color space. Research has shown that CIE-Lab is a uniform color space [13]. That is, the CIE-Lab is a linear color space from human visual system point of view.

The mathematical feature of interest in (2) is the cube-root function. Consider figure 3, the plot of function $(q + (k_0 + k_1q))^{1/3} - q^{1/3}$ (perceived noise) against q (pixel value). It is clear that the perceived level of noise in the low-signal region is larger than that of the high-signal region. The graph also emphasizes the need for considering the signal-dependency of the noise in the design of image processing algorithm because the human eye is sensitive to the perturbations when the pixel values are small.

III. FILTER DESIGN

The following discussion is *independent* of the choice of the denoising algorithm. In this paper, we work with a Bayer pattern CFA [2], although the results can be extended to more general cases.

Let $R(i, j)$, $G(i, j)$, and $B(i, j)$ be the noise-free red, green, and blue pixel component values at pixel location indexed by the Cartesian coordinates, (i, j) , respectively. Define G_s as the green image G sampled by the image sensor according to the CFA pattern:

$$G_s(i, j) = \begin{cases} G(i, j) & \text{if } (i, j) \text{ corresponds to a green filter location on CFA.} \\ 0 & \text{if otherwise,} \end{cases}, \quad (3)$$

and R_s and B_s are defined similarly for the filter locations of their respective colors on CFA. For ease of notation, define $X(i, j) = R_s(i, j) + G_s(i, j) + B_s(i, j)$ as the ideal image sensor output. Let $Y(i, j)$ be the measured noisy sensor value corresponding to $X(i, j)$, and assume that (1) defines the relationship between X and Y (i.e. Y represents ideal sensor output X corrupted according to the CMOS/CCD noise model). Our objective is to estimate R , G , and B given Y . In this section, a technique to estimate G from Y is presented (estimation of R and B is done in the same manner).

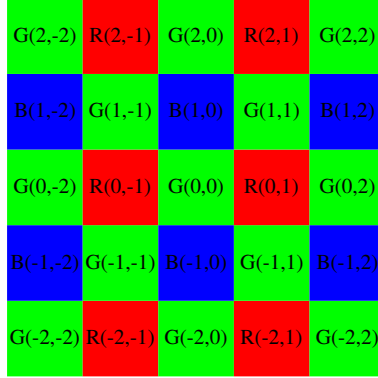


Fig. 4. Bayer pattern CFA. Cartesian coordinate is used to label the pixel locations.

Without loss of generality, let $(0, 0)$ refer to the pixel location in G whose pixel value we are interested in. We estimate the ideal green pixel value $G(0, 0)$ by taking a linear combination of the noisy sensor values Y :

$$\hat{G}(0, 0) = \sum_{i,j} \alpha(i, j) Y(i, j), \quad (4)$$

where $\hat{G}(0, 0)$ is our estimate for $G(0, 0)$, and $\alpha(i, j)$ is the filtering coefficient (or the weight) used for $Y(i, j)$ in the linear combination. Note that even if $(0, 0)$ is a pixel location that corresponds to a green filter on the CFA, we must still estimate the noise-free green pixel value. Therefore unlike the demosaicing problem, we do not draw a distinction between estimation of a missing pixel component value and estimation of the ideal pixel value from noisy pixel values. One obvious approach to choosing α is to treat each color plane separately, i.e. use only green pixels to estimate $G(0, 0)$. However, many have argued that this is ineffective because it does not take advantage of the spatial redundancies between the different colors [8] [9] [16] [15] [29].

We instead begin by assuming that the difference images $R - G$, $B - G$, $R - B$ are bandlimited signals [11]. This is equivalent to stating that the high-frequency components of R , G , and B are similar, while the low-frequency components may be dissimilar. Therefore, we impose a constraint that the coefficients corresponding to noisy red and blue values add up to 0 when estimating $G(0, 0)$, respectively. These coefficients are high-pass filters, effectively, and this guarantees that the low-frequency components of R_s and B_s do not contribute to the estimation of G .

Since only the high frequency components of R_s and B_s are passed by the filter coefficients $\alpha(i, j)$, we make the following substitution:

$$\begin{aligned} \hat{G}(0, 0) &= \sum_{i,j} \alpha(i, j) Y(i, j) \\ &= \sum_{i,j} \alpha(i, j) [G(i, j) + (k_0 + k_1 X(i, j)) \delta(i, j)]. \end{aligned}$$

This substitution suggests that we design filter coefficients $\alpha(i, j)$ as if we are optimally estimating a pixel value from a noisy single-color image.

Suppose we adapt another generalization, motivated by multi-resolution analysis and wavelets [12] [23]: if α is chosen such that

$$G(0,0) \simeq \sum_{i,j} \alpha(i,j)[G(2i,2j) + (k_0 + k_1 X(i,j))\delta(i,j)],$$

then

$$G(0,0) \simeq \sum_{i,j} \alpha(i,j)[G(i,j) + (k_0 + k_1 X(i,j))\delta(i,j)].$$

That is, if the filter coefficients $\alpha(\cdot, \cdot)$ are designed to estimate $G(0,0)$ from the downsampled green image, $G(2i,2j)$, then the same coefficients would also yield a satisfactory estimate for $G(0,0)$ when they are applied to the green image, $G(i,j)$, instead. Observe also that downsampling Y by two in horizontal and vertical directions yields single-color images. These facts imply that if filter coefficients were chosen by existing (single-color) image denoising methods to optimally estimate $G(0,0)$ by linearly combining the downsampled image $Y(2i,2j)$, the same coefficients can be used to linearly combine the noisy sensor data $Y(i,j)$ to get a reasonable estimate of $G(0,0)$.

To summarize, the strategy for choosing the filter coefficients to estimate $G(0,0)$, regardless of the color of $Y(0,0)$, consists of three major steps:

- 1) Design filter α as if we are estimating $G(0,0)$ by taking a linear combination of $\{G(2i,2j)\}$.
- 2) Add a restriction to the filter such that the filter coefficients corresponding to the noisy red and blue values add up to zero, respectively.
- 3) Apply the filter to noisy image sensor output Y using (4).

We remind the readers that the same technique is used for estimating $R(0,0)$ and $B(0,0)$ from Y .

The strategy outlined in this section can be extended to a wavelet framework, also. Instead of requiring that the filter coefficients corresponding to red (blue) values add up to zero, these coefficients can be restricted to a subspace orthogonal to the scaling functions. Because scaling functions are low-frequency components, only the wavelet (high-frequency) components of R_s (B_s) contribute to the estimation of G .

IV. DENOISING METHOD

We are left with the task of designing a denoising algorithm that will fulfill the constraints outlined in section III. There are many existing image denoising algorithms that are compatible with these constraints, offering flexibilities and choices in the design of an image processing pipeline. For example, a simple image denoising method, such as bilateral filtering [28], may be combined with demosaicing procedure when the computational resource is limited. On the other hand, combining a more sophisticated image denoising method, such as [22] and [12], with demosaicing procedure may yield improved image quality.

In this section, a demosaicing algorithm based on a total least squares (TLS) image denoising method is briefly described [12]. The following discussions are intended as a proof-of-concept case study *only*. That is, the choice of denoising method is not unique, and we remind the reader that the main goal of this paper is to establish a theoretical framework for combining demosaicing with denoising.

A. TLS Denoising Problem

TLS is a natural way to approximate the data as a linear combination of the basis vectors when the basis vectors are contaminated by errors or noise. In general, the data does not live in the subspace spanned by these basis vectors, and therefore perturbations are introduced in the system to account for the inexactness of the approximation model. While the least squares methods allow perturbation in the data only, total least squares allow perturbations in the noisy basis vectors also. Given that the noise is present in the basis vectors, allowing perturbation in the basis vectors makes intuitive sense [5] [12]. The weights to the linear combination can be chosen such that the *overall* perturbation is minimized [5] [7].

Let π be an arbitrary variable with arbitrary dimensions. As a notational convention, let $\vec{\pi}$ (a variable with an arrow) indicate a column vector, whose elements are the elements of π , bijectively reordered into a vector. For the rest of this paper, let α have $n \times n$ compact support, also. Define $\vec{x}_0, \vec{y}_0 \in \mathbb{R}^{n^2}$ as $n \times n$ image patches from X and Y (i.e. $n \times n$ vector cropped from an image), respectively, whose center value corresponds to the pixel location $(0, 0)$ (the pixel-of-interest). In this section, we are interested in designing a filter α such that $\hat{G}(0, 0) = \sum \alpha(i, j)[G(2i, 2j) + (k_0 + k_1 x_0(i, j))\delta(i, j)]$ is an optimal estimate of $G(0, 0)$ in the TLS sense (the same techniques are used to estimate $R(0, 0)$ and $B(0, 0)$). Complete details of the denoising algorithm can be found in [12].

Let Y_1, Y_2, Y_3, Y_4 be the four noisy single-color images obtained from downsampling Y by 2 in both horizontal and vertical directions (see figure 5 for illustration). Define $\{\vec{y}_1, \dots, \vec{y}_m\}$ as a set of vectorized $n \times n$ image patches cropped from Y_1, \dots, Y_4 , let $\{\vec{x}_1, \dots, \vec{x}_m\}$ be the ideal red, green, and blue image patches (i.e. noise free) corresponding to $\{\vec{y}_1, \dots, \vec{y}_m\}$, respectively, and

$$\vec{z}_k = \vec{x}_k + k_0 \delta_k + k_1 \text{diag}(\vec{x}_0) \delta_k,$$

where $\delta_k \sim \mathcal{N}([0, \dots, 0]^T, I)$ is a noise vector. In general, images patches are taken from the spatial vicinity of the pixel of interest by spatially displacing an $n \times n$ window by a few pixels (thus the pixel of interest does not necessarily lie in the center of the patch). Suppose $\vec{\alpha} \in \mathbb{R}^{n^2}$ is chosen such that for all k , $\vec{z}_k^T \vec{\alpha}$ is an optimal estimate for the center value in x_k . If this family of image patches is *similar* to y_0 then it is reasonable to assume that α will be a good filter for (4), also. A measure of similarity is introduced later.

Define $X^d = [\vec{x}_1, \dots, \vec{x}_m]^T$, $Y^d = [\vec{y}_1, \dots, \vec{y}_m]^T$, $Z^d = [\vec{z}_1, \dots, \vec{z}_m]^T$, and let \vec{x}^d be the column in X^d that corresponds to the center pixels of $\{\vec{x}_1, \dots, \vec{x}_m\}$. Suppose we allow a small perturbation in Z^d and \vec{x}^d called E and e_0 , respectively, and $\Phi \in \mathbb{R}^{m \times m}$ and $\Theta \in \mathbb{R}^{n^2-1 \times n^2-1}$ are weighting matrices. In order that $Z^d \vec{\alpha}$ be an optimal estimate for \vec{x}^d in the weighted TLS sense, $\vec{\alpha}$ solves the following criterion:

$$\min_{\alpha} \|\Phi[E, e_0]M\Theta\|_F^2, \quad (5)$$

subject to

$$(Z^d + E)\alpha = \vec{x}^d + e_0$$

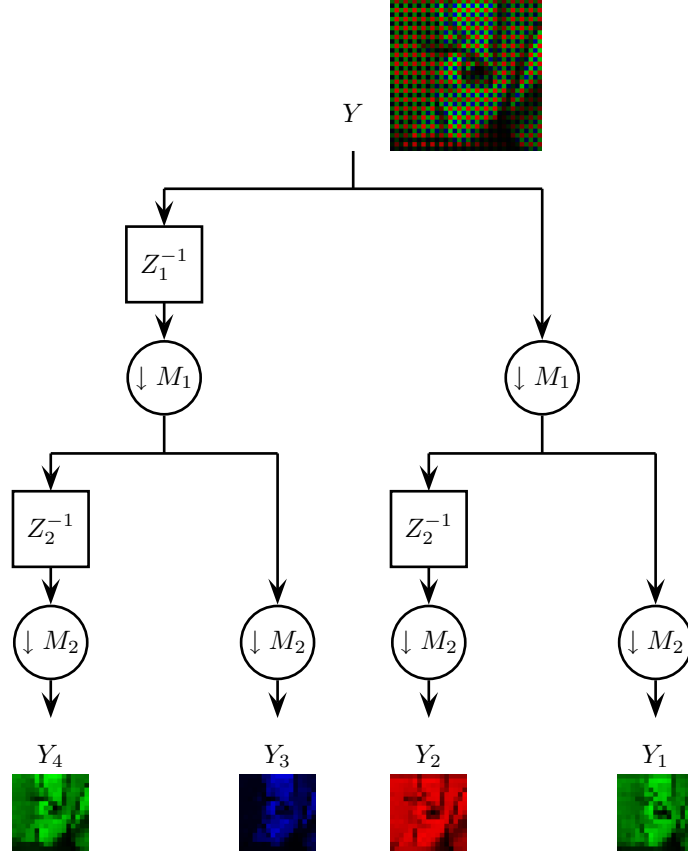


Fig. 5. Illustration of downsampled Bayer pattern image. Z_1^{-1} and Z_2^{-1} represent delay lines in the vertical and horizontal directions, respectively. $\downarrow M_1$ and $\downarrow M_2$ represent downsampling by 2 in vertical and horizontal directions, respectively.

where $\vec{\alpha}$ takes the form

$$\vec{\alpha} = M\vec{\beta}. \quad (6)$$

In other words, we are interested in choosing α such that it minimizes the *overall* perturbation, *weighted* by matrices Φ and $M\Theta$, as in (5). Perturbations E and e_0 are needed because $x^d \notin \text{span}\{z_1^d, \dots, z_m^d\}$ in general. Weighting matrices Φ and Θ controls the amount of perturbation allowed in each image patch and pixel location, respectively [12]. Our choice of weighting matrices are outlined in section IV-C.

Note that $M \in \mathbb{R}^{n^2+1 \times n^2-1}$ restricts $\vec{\alpha}$ to the subspace spanned by the columns of M . This is convenient for constraining $\vec{\alpha}$ such that coefficients corresponding to red and blue pixels add up to zero, respectively. An example of M matrix is (assumes \vec{y}_k is vectorized such that like colors are grouped together):

$$M = \begin{bmatrix} M_R & & & \\ & M_G & & \\ & & M_B & \\ & & & 1 \end{bmatrix}, \quad (7)$$

where $M_G = I$, $M_R, M_B = [I; -1, \dots, -1]$, and I is an identity matrix. Our strategy is to solve for optimal $\vec{\beta}$, and set $\vec{\alpha} = M\vec{\beta}$.

A variation of the TLS problem (5) using an affine approximation model was solved by de Groen [5]. He showed that the cost function (or the weighted perturbation), $\|\Phi[E, e_0]M\Theta\|_F^2$, is reduced greatly when the column-means of $\Phi[Z^d, \bar{x}^d]M\Theta$ are subtracted from their respective columns first, suggesting a better model fit. In this paper, we modify the approach outlined in section III to take advantage of the affine approximation technique.

More specifically, instead of (5), we solve for $\vec{\alpha}$ in the system that minimizes $\|\Phi[E, e_0]M\Theta\|_F^2$ subject to $(\tilde{Z}^d + E)\alpha = \tilde{x}^d + e_0$. Here, $\tilde{Z}^d = Z^d - [1, \dots, 1]^T \bar{z}$ and $\tilde{x}^d = \bar{x}^d - [1, \dots, 1]^T \bar{x}^d$, where $\bar{z} = \frac{1}{m} \sum_{k=1}^m \vec{z}_k$ and \bar{x}^d is the average value of \bar{x}^d . \tilde{X}^d and \tilde{Y}^d are defined similarly. Note that the average of the column in Y^d corresponding to the center pixel is a good approximation for \bar{x}^d (see (1)). Once $\vec{\alpha}$ is solved, our optimal estimate for \bar{x}^d is:

$$\hat{x}^d = \tilde{Z}^d \vec{\alpha} + \bar{x}^d.$$

More importantly, let $\{\vec{r}_1, \dots, \vec{r}_m\}$ be a set of $n \times n$ image patches cropped from the noisy sensor output Y , whose center value corresponds to the pixel location $(2i, 2j)$, $-\frac{\sqrt{m}-1}{2} \leq i, j \leq \frac{\sqrt{m}-1}{2}$. Note that the relative locations of the red and blue pixels in the CFA pattern is the same for y_0 and r_k . Define $\bar{y}_0 = \frac{1}{m} \sum_{k=1}^m \vec{r}_k$. Then, our best estimate for $G(0, 0)$ is

$$\hat{G}(0, 0) = \tilde{y}_0 \vec{\alpha} + \bar{x}^d,$$

where $\tilde{y}_0 = \bar{y}_0 - \bar{y}_0$.

B. Solution to TLS

Solving the TLS system above is straightforward [12]. Let $N = n^2 - 1$, and assume that the weighting matrices from (5) are diagonal: $\Phi = \text{diag}(\phi_1, \dots, \phi_m)$ and $\Theta = \text{diag}(\theta_1, \dots, \theta_N)$. Using singular value decomposition of matrix $\Phi[\tilde{Z}^d, \tilde{x}^d]\Theta = U\Sigma V^T$, where $\Sigma = \text{diag}(\sigma_1, \dots, \sigma_N)$ and $\sigma_k^2 > \sigma_{k+1}^2$, $\vec{\beta}$ that solves (5) is [7]:

$$\vec{\beta} = -\frac{\theta_N}{v_{N,N}} \begin{pmatrix} \theta_1 & & \\ & \ddots & \\ & & \theta_{N-1} \end{pmatrix} \begin{bmatrix} v_{1,N} \\ \dots \\ v_{N-1,N} \end{bmatrix}, \quad (8)$$

where $[v_{1,N}, \dots, v_{N,N}]^T$ is the right singular vector corresponding to σ_N . However, \tilde{x}^d is not available in the denoising problem, thus making it difficult to compute V from singular value decomposition. To work around this problem, we define the matrix P :

$$\begin{aligned} P &= (\Phi[\tilde{Z}^d, \tilde{x}^d]M\Theta)^T (\Phi[\tilde{Z}^d, \tilde{x}^d]M\Theta) \\ &= (U\Sigma V^T)^T (U\Sigma V^T) = V\Sigma^2 V^T. \end{aligned}$$

Our strategy is to estimate P and obtain the right singular vector V through its eigen decomposition. Note that $\mathcal{E}\{\delta_k\} = 0$ and $\mathcal{E}\{\delta_k \delta_l^T\} = I$ if $k = l$ and 0 if otherwise. When $m \gg N$, $P = \mathcal{E}\{P\}$, and

$$P = \mathcal{E}\{P\} = \Theta^T M^T \begin{bmatrix} \mathcal{E}\{\tilde{Z}^{dT} \Phi^2 \tilde{Z}^d\} & \tilde{X}^{dT} \Phi^2 \tilde{x}^d \\ \tilde{x}^{dT} \Phi^2 \tilde{X}^d & \tilde{x}^{dT} \Phi^2 \tilde{x}^d \end{bmatrix} M\Theta. \quad (9)$$

With some manipulations, $\mathcal{E}\{\tilde{Z}^{dT}\Phi^2\tilde{Z}^d\}$ simplifies to:

$$\mathcal{E}\{\tilde{Z}^{dT}\Phi^2\tilde{Z}^d\} = \tilde{X}^{dT}\Phi^2\tilde{X}^d + \text{diag}(k_0 + k_1\bar{x}_0)^2 \left(\sum_{i=1}^m \phi_i^2 \right). \quad (10)$$

Given Y , P can be estimated. Let $P_{YY} = \tilde{Y}^{dT}\Phi^2\tilde{Y}^d$ and $\tilde{x}_i = \vec{x}_i - \bar{x}$, $\tilde{y}_i = \vec{y}_i - \bar{y}$. For $m \gg N$, $P_{YY} = \mathcal{E}\{P_{YY}\}$, and it simplifies to:

$$\begin{aligned} P_{YY} &= \tilde{X}^{dT}\Phi^2\tilde{X}^d + \sum_{i=1}^m \phi_i^2 \text{diag}(k_0 + k_1\bar{x})^2 + \sum_{i=1}^m \phi_i^2 k_1^2 \text{diag}(\tilde{x}_i)^2 \\ &\quad + 2 \sum_{i=1}^m \phi_i^2 k_1 \text{diag}(k_0 + k_1\bar{x}) \text{diag}(\tilde{x}_i). \end{aligned}$$

Using the substitution $\mathcal{E}\{\sum_i \phi_i^2 \tilde{y}_i\} = \sum_i \phi_i^2 \tilde{x}_i$ and the fact that the diagonal entries of $\tilde{X}^{dT}\Phi^2\tilde{X}^d$ and $\sum_i \phi_i^2 \text{diag}(\tilde{x}_i)^2$ are identical, $\tilde{X}^{dT}\Phi^2\tilde{X}^d$ can be estimated using the following procedure:

- 1) Compute $P_{YY} = \tilde{Y}^{dT}\Phi^2\tilde{Y}^d$.
- 2) Compute $P_{YY} - \sum \phi_i^2 [\text{diag}(k_0 + k_1\bar{x})^2 + 2k_1 \text{diag}(k_0 + k_1\bar{x}) \text{diag}(\tilde{y}_i)]$.
- 3) Multiply the diagonal entries of step 2 by $(1 + k_1^2)^{-1}$.

Let us call this estimate P_{XX} . The estimates of $\tilde{X}^{dT}\Phi^2\tilde{x}^d$, $\tilde{x}^{dT}\Phi^2\tilde{X}^d$, and $\tilde{x}^{dT}\Phi^2\tilde{x}^d$ are obtained by taking appropriate rows and columns of P_{XX} . $\mathcal{E}\{\tilde{Z}^{dT}\Phi^2\tilde{Z}^d\}$ is computed from P_{XX} using (10) and exchanging \bar{y}_0 in lieu of \bar{x}_0 (this substitution is justified in [12]). Thus, matrix P is fully computable.

To summarize, the strategy to solving the TLS system consists of four major steps:

- 1) Estimate P_{XX} from $P_{YY} = \tilde{Y}^{dT}\Phi^2\tilde{Y}^d$.
- 2) Estimate $P = (\Phi[\tilde{Z}^d, \tilde{x}^d]M\Theta)^T(\Phi[\tilde{Z}^d, \tilde{x}^d]M\Theta)$ using P_{XX} .
- 3) Compute V using eigen decomposition: $P = V\Sigma^2V^T$.
- 4) Solve for optimal $\vec{\alpha}$ using (8) and (6).

This $\vec{\alpha}$ solves (5) subject to $(\tilde{Z}^d + E)\alpha = \tilde{x}^d + e_0$. Our best estimate for $G(0, 0)$ is

$$\hat{G}(0, 0) = \tilde{y}_0^T \vec{\alpha} + \bar{x}^d.$$

Same technique is used to estimate $R(0, 0)$ and $B(0, 0)$ from noisy sensor data, Y .

C. Weighting Matrices

Above, $\Phi = \text{diag}(\phi_1, \dots, \phi_m)$ and $\Theta = \text{diag}(\theta_1, \dots, \theta_N)$ are weighting matrices. In this paper, the $n \times n$ image patches $\{\vec{y}_1, \dots, \vec{y}_m\}$ are taken from Y_1, \dots, Y_4 in the spatial vicinity of $G(0, 0)$ [12]. Though not all image patches may include the pixel of interest, there are many image patches that are similar to \vec{y}_k . On the other hand, because natural images contain discontinuous signals, not every \vec{y}_k shares the same image attributes with \vec{y}_0 . To prioritize $\{\vec{y}_1, \dots, \vec{y}_m\}$ in the order of similarity, larger weight is given (i.e. larger ϕ_k) if $H^T \vec{y}_k$ is similar to $H^T \vec{y}_0$. More specifically,

$$\phi_k = \exp(-(\vec{y}_0 - \vec{y}_k)^T H H^T (\vec{y}_0 - \vec{y}_k) / k_\phi)$$

where $k_\phi \in \mathbb{R}$ is a constant that controls how quickly ϕ_k rolls off as $(\vec{y}_0 - \vec{y}_k)^T H H^T (\vec{y}_0 - \vec{y}_k)$ becomes large, and a H is a weighting matrix. In this paper, H is M from (7) and the experimental results have $\theta_1, \dots, \theta_{N-1} = 1$ and $\theta_N = 0.5$.

D. Pre-Processing

The effectiveness of the TLS denoising algorithm depends on our ability to estimate P matrix accurately. Given $\delta(i, j) \sim N(0, 1)$, there will be one or two pixels occasionally that stand out because the value of δ at that pixel position is far greater than its standard deviation. This is problematic because such entries in Y appear more than once, degrading our estimate for P greatly. To work around this problem, we propose to prune the outliers. The following pre-processing procedure was used. For each pixel location in Y ,

- 1) Let w be a set of pixels in Y that fall within the $L \times L$ neighborhood of the pixel of interest, and whose color is the same as the pixel of interest.
- 2) Find the k th largest and k th smallest pixel values in w .
- 3) If the pixel of interest is larger (smaller) than the k th largest (smallest) value in w , replace it with the k th largest (smallest) pixel value in w .

The proposed pre-processing procedure is a particularly good match for working with image sensors. Due to variabilities in manufacturing processes, the image sensors often contain a few defective pixel sensors, sometimes referred to as *hot-pixels* or *dead-pixels*. Because the output values from these pixel sensors do not have any relevance to the image (i.e. an outlier), and because most demosaicing algorithms contain filtering, these pixels can potentially degrade the quality of the output image significantly. Although improvements in the manufacturing process is desirable, designing a digital system to remove the defective pixels can increase the yield. Assuming that the defective pixel sensors do not cluster, the pre-processing procedure above should effectively remove them.

V. IMPLEMENTATION AND RESULTS

Our TLS algorithm is implemented by taking 5×5 image patches from a 25×25 neighborhood. Pre-processing had a window size of 11×11 , and we picked the 4th smallest and largest pixel values. Parameters k_0 and k_1 were available *a priori*. When $k_1 = 0$, algorithms to estimate the noise variance have been proposed [14]. In real-world CMOS and CCD sensors, however, k_0 and k_1 depend closely on the programmable amplifiers in the A/D converter. We may assume that the gain for the amplifier is provided, and that relationship between the gain and k_0, k_1 is understood from the calibration experiments.

Experiments were performed on natural images (24-bit/pixel color images) that are widely used: *parrot*, *lena*, and *clown* (see figure 6). A color image is sampled according to Bayer CFA pattern and pseudo-random noise is added using the CMOS/CCD noise model (1) to simulate what the image sensor sees. The full color representation of the color image is then reconstructed using the proposed method. The output image is evaluated by comparing it to the original image using mean squared error (MSE), mean absolute error (MAE), and SCIELAB [30]. While MSE is a popular method to evaluate the performance of imaging algorithms, SCIELAB is a perceptually sensitive error

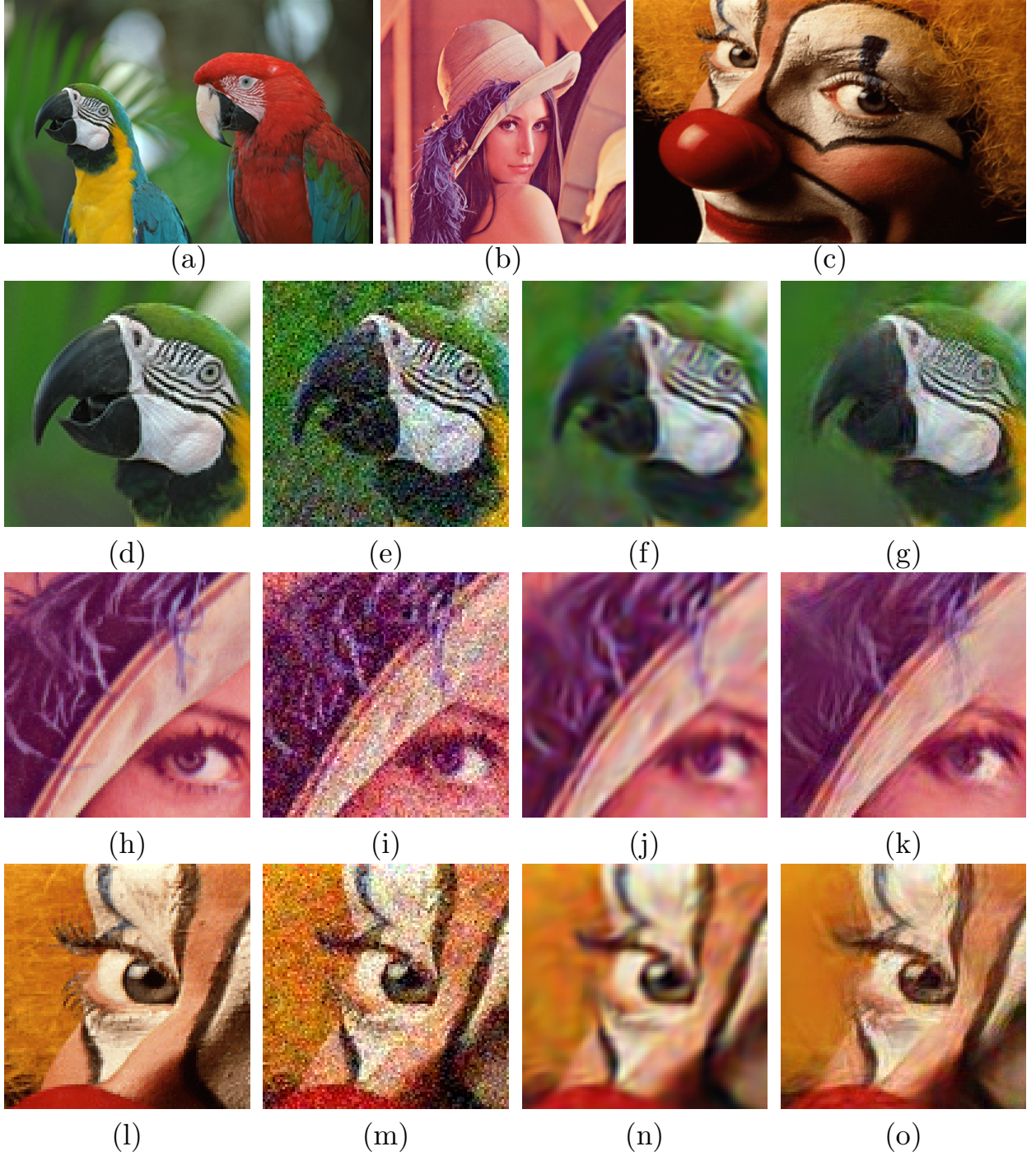


Fig. 6. Example images. (a) Original “parrot” image (256×384). (b) Original “lena” image (512×512). (c) Original “clown” image (200×320). (d)-(o) Zoomed parts of the test images. (d)(h)(l) Original images. (e)(i)(m) Reconstructed from noisy sensor data using method in [8]. (f)(j)(n) Reconstructed from noisy sensor data using methods in [8] and [25]. (g)(k)(o) Reconstructed from noisy sensor data using the proposed method. Noisy sensor data contained signal-independent noise, $(k_0, k_1) = (25, 0)$.

TABLE I

COMPARISON OF DEMOSAICING AND DENOISING ALGORITHMS ON THE “PARROT” IMAGE, EVALUATED USING MSE, MAE, AND AVERAGE SCIELAB ERROR [30]. NOISE LEVELS CONSIDERED WERE $(k_0, k_1) = (0, 0)$, $(25, 0)$, $(25, 0.1)$, AND $(25, 0.2)$. SEE TEXT.

noise parameter (k_0, k_1)	error metric		MSE			MAE			SCIELAB		
			demosaicing method			demosaicing method			demosaicing method		
			[9]	[11]	proposed	[9]	[11]	proposed	[9]	[11]	proposed
(0,0)			13.28	11.10	29.70	1.59	1.42	2.31	1.66	1.01	3.44
(25,0)	denoising method	none	560.65	600.32	101.91	18.83	19.49	6.75	58.10	69.18	22.27
		[25]	110.67	118.69	N/A	7.14	7.76	N/A	26.11	29.28	N/A
		[24]	117.05	126.28	N/A	7.24	7.88	N/A	26.73	30.00	N/A
		[12]	85.82	92.52	N/A	6.52	6.71	N/A	22.65	24.38	N/A
(25,0.1)	denoising method	none	982.89	1057.36	147.90	24.73	25.69	8.03	86.68	104.67	29.17
		[25]	165.06	176.45	N/A	8.63	9.44	N/A	35.77	40.24	N/A
		[24]	161.69	179.94	N/A	8.65	9.51	N/A	36.76	41.51	N/A
		[12]	126.53	150.02	N/A	7.85	8.84	N/A	31.86	36.68	N/A
(25,0.2)	denoising method	none	1490.89	1599.20	192.78	30.18	31.33	9.09	122.49	149.82	36.33
		[25]	229.64	253.41	N/A	10.18	11.21	N/A	47.48	53.27	N/A
		[24]	227.34	254.21	N/A	10.24	11.23	N/A	49.77	55.11	N/A
		[12]	173.09	205.74	N/A	9.13	10.25	N/A	42.23	48.55	N/A

metric that takes human visual system models into account. Therefore, SCIELAB error values are more reliable and meaningful than MSE as a way to evaluate output images when it is established *a priori* that human eye is the end user [30]. MAE is reported in [20] also.

To the best of knowledge of the authors, this is the first study of combining demosaicing and denoising algorithms. We therefore compare our results to the state-of-the-art demosaicing algorithms [8] [11] followed by denoising algorithms [24] [25] [12]. All denoising algorithms are provided with *a priori* parameter, k_0 , and denoising is performed on each color plane separately. Note that the methods in [25] and [24] are not equipped to handle the case when the noise is signal-dependent. In order to compare the proposed algorithms to the existing the methods fairly, the methods in [25] and [24] are combined with generalized homomorphic operator [6]. This operator approximately decouples the noise from the signal. See [6] for details.

Tables I-III summarizes the MSE, MAE, and average SCIELAB errors for the test images in figure 6. All three tables have the same format, but each use different image data that are corrupted by noise: $(k_0, k_1) = (0, 0)$, $(25, 0)$, $(25, 0.1)$, $(25, 0.2)$. The images are reconstructed using different demosaicing and denoising methods. The output images are compared to the original images using MSE (in dB), MAE (in pixel magnitudes), and average SCIELAB errors (in CIELAB magnitudes). For example, to read the MSE value for the output image reconstructed using demosaicing method in [11] and denoising method in [12] when the sensor data is corrupted by the noise $(25, 0.1)$, locate the row marked with “ [12]” in the second column immediately to the right of “(25,0.1)”. Also,

TABLE II

COMPARISON OF DEMOSAICING AND DENOISING ALGORITHMS ON THE “LENA” IMAGE, EVALUATED USING MSE, MAE, AND AVERAGE SCIELAB ERROR [30]. NOISE LEVELS CONSIDERED WERE $(k_0, k_1) = (0, 0)$, $(25, 0)$, $(25, 0.1)$, AND $(25, 0.2)$. SEE TEXT.

	error metric		MSE			MAE			SCIELAB		
noise parameter (k_0, k_1)			demosaicing method			demosaicing method			demosaicing method		
			[9]	[11]	proposed	[9]	[11]	proposed	[9]	[11]	proposed
(0,0)			26.47	27.40	36.17	2.99	2.96	3.54	1.45	1.47	2.87
(25,0)	denoising method	none	578.82	619.02	95.00	19.12	19.73	6.99	46.02	53.88	19.75
		[25]	92.13	109.25	N/A	7.17	7.87	N/A	20.26	22.12	N/A
		[24]	93.99	109.10	N/A	7.19	7.80	N/A	21.18	22.97	N/A
		[12]	83.86	101.68	N/A	6.77	7.50	N/A	19.44	21.97	N/A
(25,0.1)	denoising method	none	1181.57	1249.97	139.08	27.16	27.87	8.54	76.49	88.49	26.99
		[25]	161.27	193.38	N/A	9.66	10.63	N/A	30.44	33.15	N/A
		[24]	166.04	198.85	N/A	9.77	10.70	N/A	32.33	35.13	N/A
		[12]	134.46	165.10	N/A	8.73	9.78	N/A	28.93	32.11	N/A
(25,0.2)	denoising method	none	1941.33	2059.54	194.62	34.57	35.49	10.23	114.87	134.10	36.01
		[25]	284.30	357.29	N/A	12.92	14.40	N/A	44.60	50.16	N/A
		[24]	294.84	365.71	N/A	13.12	14.48	N/A	47.86	53.38	N/A
		[12]	206.98	267.17	N/A	10.89	12.42	N/A	41.40	47.18	N/A

TABLE III

COMPARISON OF DEMOSAICING AND DENOISING ALGORITHMS ON THE “CLOWN” IMAGE, EVALUATED USING MSE, MAE, AND AVERAGE SCIELAB ERROR [30]. NOISE LEVELS CONSIDERED WERE $(k_0, k_1) = (0, 0)$, $(25, 0)$, $(25, 0.1)$, AND $(25, 0.2)$. SEE TEXT.

	error metric		MSE			MAE			SCIELAB		
noise parameter (k_0, k_1)			demosaicing method			demosaicing method			demosaicing method		
			[9]	[11]	proposed	[9]	[11]	proposed	[9]	[11]	proposed
(0,0)			50.56	63.77	105.95	3.91	4.27	5.04	14.52	16.29	31.14
(25,0)	denoising method	none	492.08	544.20	194.00	16.95	17.82	10.21	242.97	266.28	156.82
		[25]	203.92	210.90	N/A	10.74	11.04	N/A	218.19	221.16	N/A
		[24]	195.73	211.46	N/A	10.57	11.04	N/A	218.42	221.78	N/A
		[12]	184.88	216.84	N/A	10.03	10.64	N/A	210.55	221.56	N/A
(25,0.1)	denoising method	none	814.11	887.26	252.06	21.27	22.23	11.49	283.61	306.08	176.41
		[25]	273.78	278.10	N/A	12.23	12.45	N/A	243.33	239.11	N/A
		[24]	274.02	276.91	N/A	12.20	12.43	N/A	246.50	240.67	N/A
		[12]	236.21	278.12	N/A	11.45	12.16	N/A	233.04	235.52	N/A
(25,0.2)	denoising method	none	1216.72	1311.51	314.41	25.31	26.36	12.70	331.96	355.38	195.77
		[25]	362.20	371.97	N/A	13.83	14.07	N/A	273.37	261.21	N/A
		[24]	362.27	371.98	N/A	13.83	14.06	N/A	276.92	263.41	N/A
		[12]	313.51	409.05	N/A	12.98	14.09	N/A	266.07	279.74	N/A

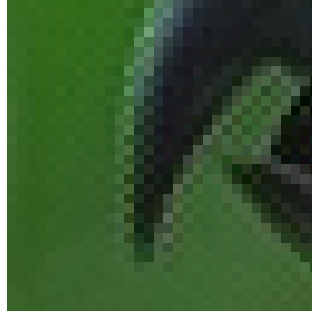


Fig. 7. Zoomed part of an output image from the proposed algorithm. In the absence of noise ($(k_0, k_1) = (0, 0)$), the algorithm occasionally suffers from zippering artifacts.

locate the column marked with “ [11]” immediately under “MSE”. The intersection of this row and this column is the MSE value we are interested in. In the absence of noise (i.e. $(k_0, k_1) = (0, 0)$), no image denoising methods were applied. In the case that the pixel values were estimated using the proposed method, the denoising algorithm is not necessary (therefore marked by “N/A”).

In the absence of noise ($(k_0, k_1) = (0, 0)$), the proposed method did not perform as well as demosaicing methods in [8] and [11]. In the presence of noise, the tables clearly show the benefits of considering demosaicing and denoising as a single operation. The performance according to MSE and MAE is comparable for the proposed algorithm and the other reconstruction methods. However, SCIELAB error values indicate that the output from the proposed algorithm is noticeably closer to the original image than that of other methods. The differences are especially pronounced when $k_1 \neq 0$. The performance evaluation according to SCIELAB error values is consistent with the visual quality assessment of the output images. Figure 6 shows zoomed output images, reconstructed from noisy sensor data $(k_0, k_1) = (25, 0)$. The amplification of noise is evident due to demosaicing (figures 6(e), (i), and (m)). While applying a denoising algorithm after demosaicing algorithm helps the overall image quality (figures 6(g), (j), and (n)), the proposed algorithm is clearly sharper (see feathers in figure 6(k), eye in figure 6(l)). The differences are especially pronounced in the smooth regions of the image, where the proposed algorithm suppresses the noise more effectively, without introducing extraneous colors (see solid color regions in figures 6(f) and (g)). However, in the absence of noise ($(k_0, k_1) = (0, 0)$), the proposed algorithm occasionally suffers from zippering artifacts (see an example in figure 7). This agrees with the observations we made in tables I-III that the other demosaicing methods perform better in absence of noise.

Experiments were also performed on 8-bit sensor data taken from Agilent Technologies’ CMOS APS digital camera in low light. Images were captured in a raw-data format with the same programmable gain amplifier setup used during the calibration process. The parameters for the proposed algorithm were $(k_0, k_1) = (3, 0.02)$. After demosaicing, the images were processed with color space conversion and gamma correction ($\gamma = 1.8$). The illuminant was known *a priori*, and it was considered in the color space conversion step. Figure 8 shows examples comparing the proposed method to other demosaicing methods. The demosaicing methods in [8] and [11] maintain

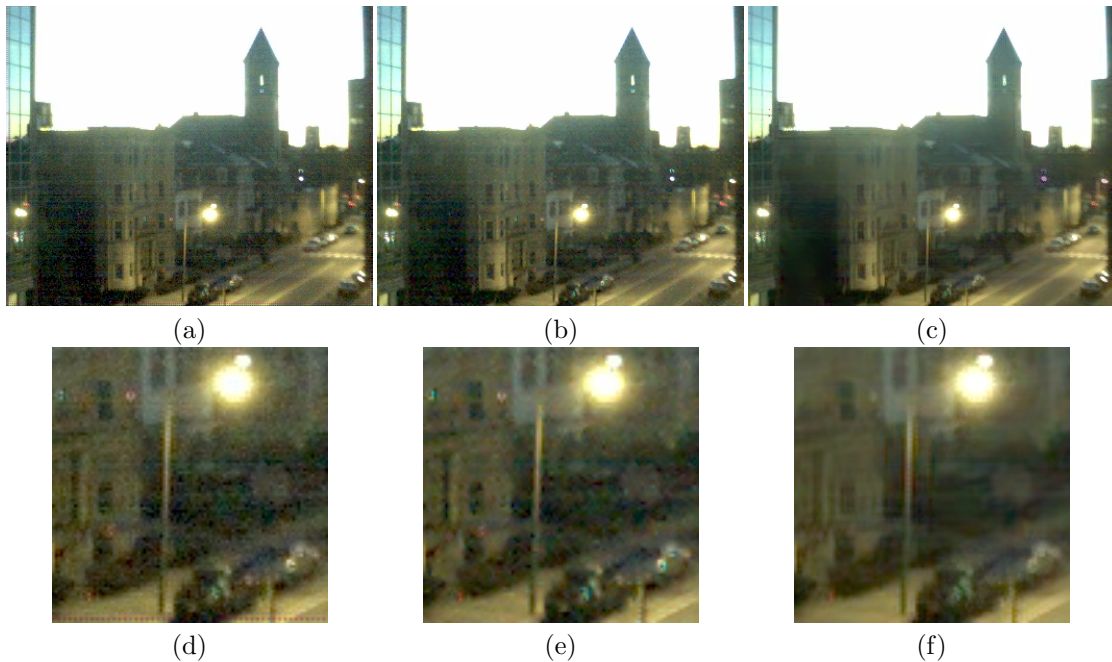


Fig. 8. Example using an image captured with Agilent Technologies' CMOS APS digital camera. (a) Method in [8] (b) Method in [11] (c) Proposed method. (d) Zoomed part of (a). (e) Zoomed part of (b). (f) Zoomed part of (c)

high contrast, but the *grainy* noise is highly visible in the dark regions of the image. While the proposed algorithm suffers from occasional zippering artifacts, the *grainy* noise is eliminated well by the proposed algorithm.

Finally, we note that the computational cost of the method proposed in section IV is high. The eigen decomposition of symmetric positive-definite matrix $P \in \mathbb{R}^{n \times n}$ requires $O(n^2)$ operations [12]. However, it is reminded that method in section IV is only an example for ideas developed in section III. The choice of denoising methods should be made according to the application's specific computational and/or image quality needs.

VI. CONCLUSION

Noting that image interpolation and image denoising are both estimation problems, this paper presented a unified method to combine demosaicing and image denoising procedures. The filtering coefficients were restricted such that only the high frequency components of the image signals contribute to the estimation of pixel values of different colors. With substitutions, the multi-colored demosaicing/denoising problem was simplified to a single-color denoising problem. Using this theoretical framework, an example algorithm was constructed, and it was tested on color images with pseudo-random noise and on raw sensor data from a real CMOS digital camera. The experimental results verify that performing demosaicing and denoising simultaneously is far more effective than treating the demosaicing and denoising problems separately.

Theoretical framework in section III introduces a powerful and flexible technique for developing new demosaicing-denoising algorithms to fit the computational and/or image quality needs of various applications. While the example

method developed in section IV is computationally expensive, it suggests that superior image quality is achieved by combining a state-of-the-art image denoising technique with demosaicing. As the research in image denoising methods advances (in terms of image quality or computational efficiency), similar developments can be made in demosaicing by using the proposed framework. The future research in this area include design and implementation of wavelet-based demosaicing-denoising method using the generalized strategy outlined at the end of section III.

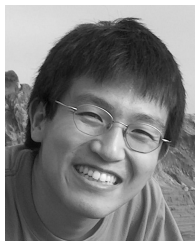
ACKNOWLEDGMENT

The authors would like to thank Agilent Technologies for making their CMOS digital camera available to us. We appreciate the help and the suggestions from Texas Instruments, Qijia Quyang, and James Prudhomme. We also thank Dr. B. Gunturk for providing the code to the demosaicing method in [8].

REFERENCES

- [1] D. Alleysson, S. Ssstrunk, J. Hraut, "Linear demosaicing inspired by the human visual system," accepted for publication in *IEEE Transactions on Image Processing*, 2004, Available online at <http://lcavwww.epfl.ch/sabines/ASH04.pdf>
- [2] B. E. Bayer, "Color imaging array," US Patent 3 971 065, 1976.
- [3] M. Borg, R. Mentzer, K. Singh, "Digital Imaging using CMOS sensors," *Proc. International IC-China Conference and Exhibition, Electronic Engineering Times*, pp. 37-47, 2001.
- [4] L. Chang, Y. P. Tang, "Effective use of spatial and spectral correlations for color filter array demosaicking," *IEEE Transactions on Consumer Electronics*, vol. 50, no. 2, pp. 355-365, May 2004.
- [5] P. de Groen, "An Introduction to Total Least Squares", *Nieuw Archief voor Wiskunde*, Vierde serie, deel 14, 1996.
- [6] R. Ding, A. N. Venetsanopoulos, "Generalized Homomorphic and Adaptive Order Statistic Filters for the Removal of Impulsive and Signal-Dependent Noise," *IEEE Trans. Circuits and Systems*, vol. 34, no. 8, pp. 948-955, August 1987.
- [7] G. H. Golub, C. F. Van Loan, "Matrix Computations", The Johns Hopkins University Press, 3rd ed., 1996.
- [8] B. K. Gunturk, Y. Altunbasak, R. M. Mersereau, "Color Plane Interpolation Using Alternating Projections," *IEEE Trans. Image Processing*, vol. 11, No. 9, pp.997-1013, September 2002.
- [9] B. K. Gunturk, J. Glotzbach, Y. Altunbasak, R. W. Schafer, R. M. Mersereau, "Demosaicking: Color filter array interpolation in single chip digital cameras," *IEEE Signal Processing Magazine* (Special Issue on Color Image Processing), vol. 22, No. 1, pp.44-54, January 2005.
- [10] G. E. Healey, R. Kondepudy, "Radiometric CCD Camera Calibration and Noise Estimation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 16, No. 3, pp. 267-276, March 1994.
- [11] K. Hirakawa, T. W. Parks, "Adaptive Homogeneity-Directed Demosaicing Algorithm," *IEEE Trans Image Processing*, vol. 14, no. 3, pp.360-369, March 2005.
- [12] K. Hirakawa, T. W. Parks, "Image Denoising for Signal-Dependent Noise," *Proc. Acoustics, Speech, and Signal Processing*, 2005, vol. 2, pp.29-32, March 2005.
- [13] International Commission on Illumination, "Recommendations on uniform color spaces, color difference equations, psychometric color terms," Supplement No.2 to CIE publication No.15 (E.-1.3.1), TC-1.3., 1971.
- [14] I. M. Johnstone, B. W. Silverman, "Wavelet threshold estimators for data with correlated noise," *Journal of Royal Statist. Soc.*, vol. B 59, 1997.
- [15] R. Kakarala, Z. Baharav, "Adaptive demosaicing with the principal vector method," *IEEE Trans. Consumer Electronics*, vol. 48, no. 4, pp.932-937, November 2002.
- [16] W. Lu, Y.-P. Tan, "Color Filter Array Demosaicking: New Method and Performance Measures," *IEEE Trans. Image Processing*, vol. 12, no. 10, pp.1194-1210, October 2003.
- [17] R. Lukac, K. N. Plataniotis, D. Hatzinakos, M. Aleksic, "A novel cost effective demosaicing approach," *IEEE Trans. Consumer Electronics*, vol. 50, no. 1, pp.256-261, February 2004.

- [18] R. Lukac and K.N. Plataniotis, "Normalized color-ratio modelling for CFA interpolation," *IEEE Transactions on Consumer Electronics*, vol. 50, no. 2, pp. 737-745, May 2004.
- [19] R. Lukac, K. Martin, K. N. Plataniotis, "Digital camera zooming based on unified CFA image processing steps," *IEEE Transactions on Consumer Electronics*, vol. 50, no. 1, pp. 15-24, February 2004.
- [20] R. Lukac, K. Martin, K. N. Plataniotis, "Demosaicked image postprocessing using local color ratios," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 14, no. 6, pp. 914-920, June 2004.
- [21] R. Lukac, B. Smolka, K. Martin, K.N. Plataniotis, A. N. Venetsanopoulos, "Vector filtering for color imaging," *IEEE Signal Processing Magazine, Special Issue on Color Image Processing*, vol. 22, no. 1, pp. 74-86, January 2005.
- [22] D. D. Muresan, T. W. Parks, "Adaptive Principal Components and Image Denoising," *Proc. Int. Conf. Image Processing*, vol. 1, pp.101-104, September 2003.
- [23] D. D. Muresan, T. W. Parks, "Demosaicing using optimal recovery," *IEEE Transactions on Image Processing*, vol. 14, no. 2, pp. 267-278, February 2005.
- [24] A. Pizurica, W. Philips, I. Lemahieu, M. Acheroy, "A joint inter- and intrascale statistical model for Bayesian wavelet based image denoising," *IEEE Trans. Image Processing*, vol. 11, no. 5, pp.545-557, May 2002.
- [25] J. Portilla, V. Strela, M. J. Wainwright, E. P. Simoncelli, "Image Denoising Using Scale Mixture of Gaussians in the Wavelet Domain," *IEEE Trans. Image Processing*, vol. 12, no. 11, pp.1338-1351, November 2003.
- [26] R. Ramanath, W. E. Snyder, "Adaptive demosaicking," *Journal of Electronic Imaging* 12(4), pp.633-642, October 2003.
- [27] H. Tian, B. Fowler, A. E. Gamal, "Analysis of Temporal Noise in CMOS Photodiode Active Pixel Sensor," *IEEE Jnl. Solid-State Circuits*, vol. 36, no. 1, pp.92-101, 2001.
- [28] C. Tomasi, R. Manduchi, "Bilateral filtering for gray and color images," *Sixth International Conference on Computer Vision*, pp. 839-46, New Delhi, India, 1998.
- [29] X. Wu, N. Zhang, "Primary-consistent soft-decision color demosaicking for digital cameras," *IEEE Transactions on Image Processing*, vol. 13, no. 9, pp. 1263-1274, September 2004.
- [30] X. Zhang, D. A. Silverstein, J. E. Farrell, B. A. Wandell, "Color Image Quality Metric S-CIELAB and its application on Halftone Texture Visibility," *COMPCON97 Digest of Papers*, IEEE, pp. 44-48, 1997.



Keigo Hirakawa Keigo Hirakawa received M.S. (2003) and Ph.D (2005) degree in electrical and computer engineering from Cornell University, Ithaca, NY, and the B.S. degree in electrical engineering from Princeton University, Princeton, NJ, in 2000. He is currently pursuing M.M. degree at New England Conservatory of Music in Jazz Piano Performance while also serving as a research assistant at the department of statistics at Harvard University, Cambridge, MA, under the direction of Prof. Xiao-li Meng. His research interests include image modeling, color representation, image denoising, demosaicing, image interpolation, and the statistical treatment of missing data.



Thomas W. Parks B.E.E., M.S., Ph.D.(Cornell) -1967. From 1967 until 1986 he was with the Department of Electrical and Computer Engineering at Rice University, Houston, TX. In 1986 he joined Cornell as a Professor of Electrical Engineering. He is a Fellow of the IEEE, a recipient of the IEEE Third Millennium Medal, and a recipient of the 2004 IEEE Jack S. Kilby Medal. He received the Humboldt Foundation Senior Scientist Award, and has been a Senior Fulbright Fellow. He has co-authored several books on digital signal processing and published a number of research papers. His research interests are signal theory and digital signal processing.