A case for denoising before demosaicking color filter array data

Sung Hee Park, Hyung Suk Kim, Steven Lansel, Manu Parmar, and Brian A. Wandell

***Abstract*—Denoising algorithms are well developed for grayscale and color images, but not as well for color filter array (CFA) data. Consequently, the common color imaging pipeline demosaics CFA data before denoising. In this paper we explore the noise-related properties of the imaging pipeline that demosaics CFA data before denoising. We then propose and explore a way to transform CFA data to a form that is amenable to existing grayscale and color denoising schemes. Since CFA data are a third as many as demosaicked data, we can expect to reduce processing time and power requirements to about a third of current requirements.**

***Index Terms*—color imaging pipeline, demosaicking, denoising, CFA denoising**

I. Introduction

The color imaging pipeline in a typical digital camera starts with sensor data acquired through a color filter array (CFA). The CFA mosaic allows the measurement of the intensity of only one waveband at a particular pixel location; the image processing pipeline must have a demosaicking stage to interpolate the subsampled CFA data to form a full-color image. Since the Bayer CFA [1] is most prevalent, in this paper we confine our attention to the Bayer CFA. Figure 1 shows the essential pieces of the imaging chain.

Sensor data are inherently noisy; the sources of noise include an array of electronic noise (thermal, readout, variance in amplifier gains, etc.), and photon shot noise due to the physics of the light measuring process. In modern imaging pipelines, the pre-processing stage reduces the most extreme sensor defects like variations in column and pixel gains, dead-pixels, etc. A separate denoising stage is typically included after the demosaicking stage [2].

Applying denoising after demosaicking is convenient because denoising operations are well developed in the context of grayscale and color images. On the other hand, CFA data do not conform to usual assumptions about images, e.g., smoothness, piecewise constant nature, etc. Denoising CFA data before demosaicking is an attractive alternative since CFA data are a third as many as three-color image data and the computation required for many denoising algorithms scales at least linearly with the number of pixels. If we are able to apply an existing denoising method directly to CFA data, we stand to reduce the computation time and power requirements by a factor of three. Recently, several researchers have proposed methods designed to denoise CFA data directly [3]–[5].

S.-H. Park, H.-S. Kim, S. Lansel and M. Parmar are with the Electrical Engineering Dept., Stanford University, Stanford CA-94305, email: shpark7@stanford.edu; B. A. Wandell is with the Psychology Dept., Stanford University.

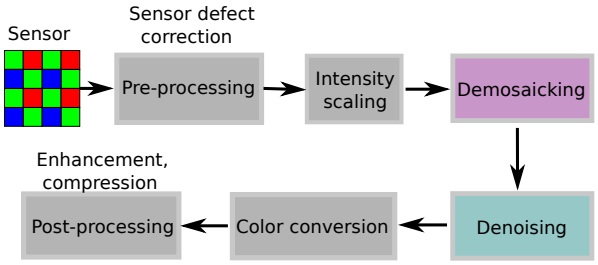


Fig. 1. Essential stages of the color imaging pipeline. Usually, a denoising stage follows the demosaicking stage.

In this paper we first motivate denoising before demosaicking by analyzing issues related to denoising demosaicked data. We then explore a novel method to denoise subsampled CFA data prior to demosaicking. This new approach requires only a small change to the pipeline. It offers the advantage of reduced computation with little change in the overall performance.

II. Demosaicking before denoising

A. Error terms – notation

Let x be an RGB image and A be an operator that samples x according to the Bayer CFA pattern; CFA data is

$$y = Ax + n, \tag{1}$$

where n is the sensor noise. We assume n is signal independent i.i.d. Gaussian with mean zero and variance σ^2. Let $D(.)$ denote the demosaicking operation, which may be nonlinear. The error in the demosaicked image found from noisy subsampled data is

$$\eta_{\mathrm{D}} = D(Ax + n) - x. \tag{2}$$

We can express η_{D} as the sum

$$\eta_{\mathrm{D}} = \eta_{\mathrm{F}} + \eta_{\mathrm{R}}, \tag{3}$$

where η_{F} is defined to be the component of error attributed to the demosaicking process; this error is fixed and corresponds to the error in the demosaicked image when the input data is noise-free. We call this component of the error the *fixed* demosaicking error

$$\eta_{\mathrm{F}} = D(Ax) - x. \tag{4}$$

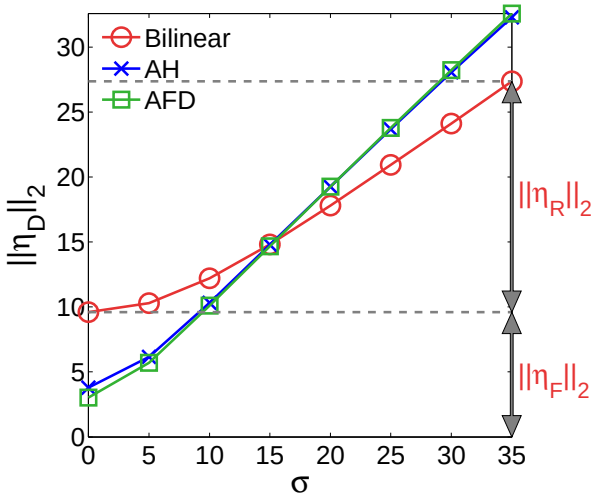


Fig. 2. Average RMS error over the 24 Kodak test images for three different demosaicking methods shown as a function of standard deviation of noise in the input data. RMS error is between a true three-color image and a demosaicked image found from noisy, subsampled data.

That leaves the component of error caused by noise. We call this the *random* demosaicking error

$$\eta_R = D(Ax + n) - D(Ax). \quad (5)$$

If the demosaicking method is linear, the random demosaicking error is simply the result of demosaicking the noise term: $\eta_R = D(n)$.

B. The relative magnitude of random and fixed demosaicking error varies with input SNR

We compare the fixed and random demosaicking errors as a function of noise-level for two adaptive demosaicking algorithms and bilinear interpolation (Fig. 2). The two adaptive methods are (a) Hirakawa and Parks' demosaicking algorithm based on adaptive homogeneity [6] denoted by AH, and (b) an adaptive frequency domain filtering scheme proposed by Dubois [7] denoted by AFD. The graph shows the total error ($||\eta_D||_2$) between a true three-color image and an image demosaicked after sampling the true image with the Bayer CFA arrangement. The average value is found over the images (8 bit per color) in Kodak's PhotoCD PCD0992 [8].

We can identify the fixed and random error components from these curves. The fixed error can be found at the zero input noise level. As the input noise increases, the increase in the total error is due to the random demosaicking error. For example, consider the fixed demosaicking error (η_F) and the random demosaicking error (η_R) for the bilinear demosaicking method when the input noise standard deviation is 35. Under noisy conditions the random demosaicking error is much higher than the fixed error. Further, note that the adaptive algorithms perform extremely well under low sensor noise conditions. However, their performance relative to the bilinear method degrades significantly as noise increases.

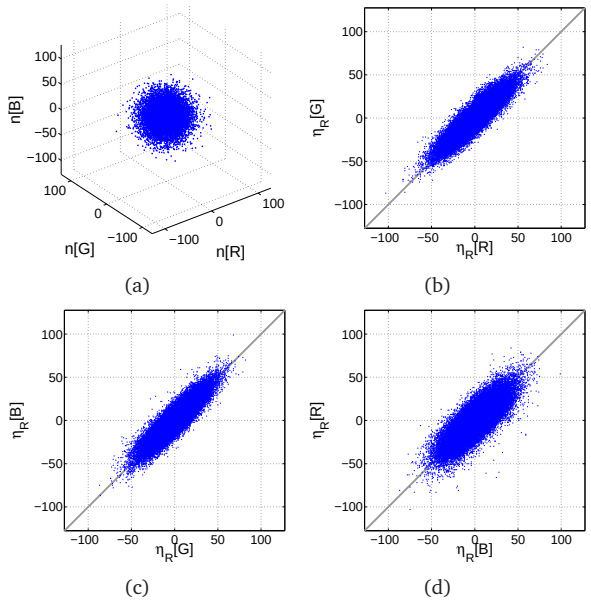
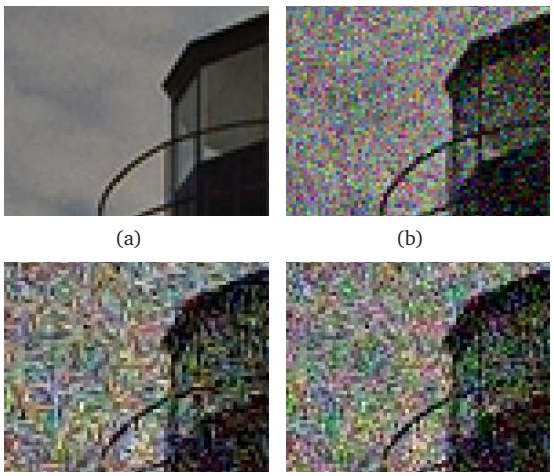


Fig. 3. Scatter diagrams of image noise for an experiment with image # 19 (Lighthouse) from the Kodak set of test images. AWGN with $\sigma = 35$ was added to the image. (a) Sensor noise before demosaicking; $n[k]$, $k = \mathrm{R,G,B}$ are the components of n in the k channel. (b-d) Random demosaicking error projected on to the RG, GB, and RB planes respectively; $\eta_R[k]$, $k = \mathrm{R,G,B}$ are the components of η_R in the k channel.

At low noise levels the fixed demosaicking error determines the pipeline quality. As input noise increases, the random demosaicking error becomes dominant. Hence, it is important to consider the interaction between input noise and the demosaicking algorithm.

C. Demosaicking introduces spatio-chromatic correlations into the sensor noise

It is convenient to study the effect of image-dependent demosaicking algorithms on the input noise via simulations rather than analysis. We simulated sensor noise according to the signal model in Eq. 1. Fig. 3(a) shows a scatter diagram of the noise in the RGB channels. Figures 3(b-d) show the RGB pair-wise distributions of the random demosaicking error. The independent sensor noise becomes correlated after demosaicking. In fact, the distribution aligns along the $(1,1,1)$ direction in RGB space. This implies that the random demosaicking error terms in the RGB channels have similar values. This is a property of most state-of-the-art Bayer demosaicking algorithms because they use high frequency information from the green channel to estimate high frequency information in the red and blue channels [9]. While this reduces visible artifacts at edges, it also introduces chromatic correlations.



(a) (b) (c) (d)

Fig. 4. Illustration of noise with different statistics. (a) Noise-free image (b) Image with AWGN with standard deviation 35 (c) Image in (b) sampled with the Bayer CFA and demosaicked with the AH method (d) Image in (b) sampled with the Bayer CFA and demosaicked with the AFD method.

In addition to the chromatic correlation, demosaicking introduces spatial correlations into the random demosaicking error. These arise because demosaicking algorithms use nearby CFA measurements to estimate missing values. For noisy images, demosaicking algorithms may confuse noise with signal and introduce false edges and patterns. Fig. 4(a) shows a cropped region from image # 19 from the Kodak set. Fig. 4(b) shows the result of adding AWGN with $\sigma = 35$ to this image. Corresponding images sampled with the Bayer CFA and demosaicked with the AH and AFD methods are shown in images 4(c,d). The noise in both demosaicked images has spatial structure (streaks in the AH image and low frequency-blotches in the AFD image).

D. Structured noise is harder to denoise

The spatio-chromatic noise structure limits the performance of the denoising algorithm. To illustrate this effect, we simulated noisy images of the form

$$z = D(Ax) + \varepsilon_i, \tag{6}$$

where

- $\varepsilon_1 = \eta_R$; so $z = D(Ax + n)$, the result of demosaicking noisy CFA data, and contains noise with spatio-chromatic correlations.
- ε_2 is spatially independent zero mean Gaussian with the same chromatic covariance as η_R. z contains noise with chromatic correlations but no spatial correlations.
- ε_3 is AWGN with variance $||\eta_R||_2^2$; z contains noise with no spatial or chromatic correlations.

Figure 5 compares the denoising performance of these three types of images at different noise levels. The denoiser is most successful in removing AWGN noise, followed by

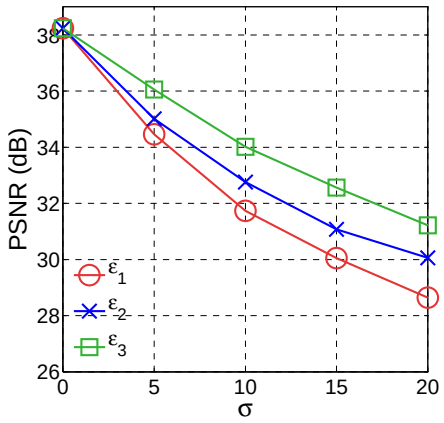


Fig. 5. Performance of the BLS-GSM [10] denoiser on images corrupted with noise of different natures – independent, correlated among color channels, and correlated among the color channels and in space. Results are for image # 19 from the Kodak set demosaicked with the AH method.

spatially uncorrelated but color correlated noise, and finally, the noise correlated in both color and space. Since demosaicking algorithms will add spatio-chromatic correlations to the AWGN sensor noise, these results motivate us to denoise before demosaicking.

III. CFA DENOISING

CFA data can not be denoised with prevalent grayscale denoising algorithms since adjacent pixels of the CFA image represent different color measurements. The CFA image has a block structure not compatible with standard assumptions of smoothness and piecewise constancy made by denoising methods. Color image denoising algorithms usually use a color space transformation to transform RGB values to a luminance-chrominance representation; the signal in transformed space is then denoised using a grayscale denoiser (Fig. 6(a)). CFA data cannot be adequately denoised using color image denoising algorithms because each pixel only contains one color measurement, and pixel-wise color space transformations are not possible.

One possible approach to denoise Bayer CFA data is to decompose the CFA image into four smaller images with the R, G_1, G_2, and B measurements corresponding to the four locations in the CFA block. The four single channel images can then be denoised with a grayscale denoising algorithm and the resulting images can be rearranged into the denoised CFA image. This method performs poorly because important information in the chromatic correlations is ignored. CFA denoising algorithms can be improved by using the spatial and chromatic correlations in the CFA data.

Another approach is to form a lower resolution RGB image by extracting the red and blue values from each CFA block and averaging the green values. This full color image can then be processed with a color denoising algorithm. This approach takes advantage of the color correlation in the

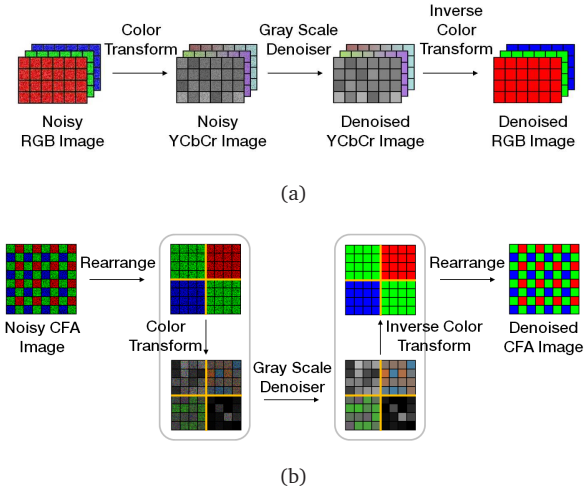


Fig. 6. (a) Color image denoising using grayscale denoisers. (b) Illustration of the proposed CFA denoising approach.

CFA image, but it fails to preserve the high frequency spatial information in the green channel of the original CFA.

Our approach is summarized in Fig. 6(b). We first rearrange the CFA image into the four RG_1G_2B channels. Then, we apply a color transformation to find a new representation much like the color space transformation used in common color denoising methods (Fig. 6(a)). The transformed signal can then be denoised using existing grayscale algorithms.

A. Selecting a color representation designed for CFA denoising

First, we choose an orthonormal color transformation to ensure the sensor noise has the same distribution in the original and transformed representations. The orthonormal property also guarantees the errors that remain in the denoised images in the transformed color space are not amplified when the data are converted back to the RG_1G_2B representation.

Second, we choose the axes of the representation from a principal components analysis (PCA) of the pixel RGB values in the Kodak data set. In the PCA basis, the data have the largest variance along the first principal component direction; the second principal component is in the direction of maximum variance of all vectors that are orthonormal to the first principal component, and so forth.

To create a transformation that can be applied to the RG_1G_2B values from the Bayer pattern, we split the energy in the green channel between G_1 and G_2. We add one more dimension to the transformation that captures the small difference between the two green coefficients G_1 and G_2. The transformation from RG_1G_2B to the proposed $C_0C_1C_2\Delta G$ representation is

$$\begin{bmatrix} C_0 \\ C_1 \\ C_2 \\ \Delta G \end{bmatrix} = \begin{bmatrix} 0.541 & 0.436 & 0.436 & 0.572 \\ -0.794 & 0.107 & 0.107 & 0.588 \\ -0.276 & 0.546 & 0.546 & -0.572 \\ 0 & 0.707 & -0.707 & 0 \end{bmatrix} \begin{bmatrix} R \\ G_1 \\ G_2 \\ B \end{bmatrix}.$$

This transform is helpful for denoising because it effectively compacts the signal energy while the noise is distributed equally in all dimensions. The signal dominates the noise in the first dimension and thus can be more easily extracted; the last few dimensions can be aggressively filtered to remove the noise. Doing so does not penalize PSNR significantly because there is less signal variance in these dimensions.

IV. Experiments

To compare the performance of the demosaic-first and denoise-first pipelines, we performed simulations with several combinations of state-of-the-art demosaicking and denoising algorithms. The results are given in Table I; images are available online at http://www.stanford.edu/~shpark7/demden/demden.html. The results show that with our CFA denoising method, the demosaic-first pipeline slightly outperforms the denoise-first pipeline. However, the difference is small.

A. Denoiser implementation

We used two different denoising methods, BLS-GSM [10] and CBM3D [11]. BLS-GSM is a grayscale denoising algorithm which we used to separately denoise the four $C_0C_1C_2\Delta G$ image channels. CBM3D is a color denoising algorithm based on the grayscale BM3D algorithm. Both CBM3D and BM3D use two stages - block matching and estimation. CBM3D improves the result of applying BM3D independently on the RGB channels by operating in luminance-chrominance space. Block matching is performed only in the luminance channel. These matched blocks are also used to process the chrominance channels. Similarly, when applying CBM3D to CFA images in our framework, we perform block matching only on the C_0 channel of the transformed image.

B. Computational Complexity

BLS-GSM has two stages, a pyramid transform that scales as $N\log_2(N)$ and an estimation stage that scales approximately as N, where N is the number of pixels. The denoise-first imaging pipeline requires at most $1/3^{\text{rd}}$ the computation as the demosaic-first pipeline.

CBM3D involves two operations - block matching and estimation. The algorithm scales linearly with the number of pixels for RGB images. When denoising before demosaicking, block matching is performed once on an image with $N/4$ pixels, which results in $1/4^{\text{th}}$ the computation for block matching compared to denoising after demosaicking. The estimation stage of the algorithm is performed on four images images with $N/4$ pixels instead of on three images with N pixels. This which results in $1/3^{\text{rd}}$ as much computation for denoise-first pipeline than the demosaic-first pipeline.

V. Conclusions

The demosaicking stage in the image processing pipeline introduces chromatic and spatial correlations to sensor noise.

TABLE I

PSNR VALUES BETWEEN THE NOISE-FREE IMAGES IN THE KODAK TEST SET AND THE FINAL OUTPUT OF THE PIPELINE. WE SHOW RESULTS FOR TWO DENOISING METHODS, CBM3D [11] AND BLS-GSM [10], AND TWO DEMOSAICKING METHODS, (AH) [6], AFD [7] FOR TWO DIFFERENT NOISE LEVELS ($\sigma = 10, 20$). DM DENOTES THE DEMOSAIC-FIRST PIPELINE AND DN DENOTES THE DENOISE-FIRST PIPELINE.

Image #	BLS-GSM								CBM3D							
	$\sigma = 10$				$\sigma = 20$				$\sigma = 10$				$\sigma = 20$			
	AH		AFD		AH		AFD		AH		AFD		AH		AFD	
	DM	DN	DM	DN	DM	DN	DM	DN	DM	DN	DM	DN	DM	DN	DM	DN
1	29.6	29.5	30.7	29.9	26.6	26.4	27.3	26.5	29.9	29.8	31.0	30.2	26.9	26.8	27.5	26. 9
2	32.0	32.6	32.7	32.6	29.3	30.1	30.1	30.2	32.9	32.7	33.3	32.6	30.3	30.7	30.7	30. 6
3	33.6	34.1	34.3	34.0	30.2	31.0	31.4	30.9	35.1	34.8	35.6	34.7	31.8	31.9	32.4	31. 9
4	32.2	32.3	32.9	32.5	29.4	29.7	30.2	29.8	33.1	32.9	33.8	33.0	30.5	30.5	30.9	30. 6
5	29.5	29.6	30.5	29.8	26.3	26.2	27.1	26.2	30.5	29.8	31.3	30.1	27.1	26.4	27.7	26. 5
6	30.8	30.8	31.8	31.0	27.6	27.6	28.3	27.6	31.3	31.0	32.1	31.2	28.0	28.0	28.6	28. 0
7	33.0	33.1	33.7	33.3	29.6	29.6	29.8	29.7	34.5	33.7	35.1	33.9	31.0	30.4	31.5	30. 5
8	29.2	29.0	29.6	29.3	26.1	25.7	26.0	25.9	30.0	29.3	30.8	29.7	26.9	26.3	27.5	26. 4
9	33.6	33.8	33.9	33.8	30.2	30.5	30.1	30.5	34.9	34.4	35.3	34.3	31.8	31.4	32.2	31. 4
10	33.3	33.3	33.6	33.4	29.9	30.3	29.9	30.3	34.6	33.8	35.1	34.0	31.5	31.0	31.9	31 .1
11	31.1	31.2	31.9	31.5	28.2	28.4	28.1	28.4	31.8	31.5	32.8	31.8	28.9	28.8	29.5	28 .9
12	33.5	33.6	33.7	33.6	30.4	30.7	30.1	30.7	34.4	34.0	35.0	34.0	31.6	31.5	31.9	31 .5
13	27.7	27.7	29.2	28.7	24.9	24.8	25.8	25.1	27.9	27.8	29.4	28.7	25.0	24.9	25.8	25 .2
14	29.5	29.9	30.5	30.0	26.9	27.1	27.6	27.1	30.5	30.0	31.0	30.0	27.6	27.2	28.0	27 .2
15	32.0	32.3	32.9	32.6	29.4	29.6	30.1	29.7	32.9	32.9	33.8	33.1	30.3	30.3	30.8	30 .4
16	32.7	32.7	33.4	32.7	29.4	29.5	30.3	29.6	33.3	33.0	34.0	33.1	30.2	30.1	30.6	30 .1
17	32.7	32.7	33.7	32.9	29.6	29.7	30.6	29.8	33.7	33.0	34.3	33.2	30.4	30.1	30.9	30 .2
18	29.7	29.7	30.9	30.2	26.9	26.9	27.8	27.1	30.2	29.7	31.2	30.3	27.3	27.1	28.0	27 .2
19	31.7	31.6	32.5	31.7	28.6	28.8	29.5	28.8	32.4	32.2	33.2	32.3	29.8	29.5	30.1	29 .5
20	32.2	32.5	32.8	32.4	28.7	28.8	29.1	28.7	33.0	32.7	33.4	32.6	29.3	29.3	29.5	29 .1
21	31.2	31.2	32.2	31.5	28.1	28.1	28.9	28.1	31.6	31.4	32.5	31.7	28.6	28.4	29.1	28 .5
22	30.9	30.9	31.7	31.2	28.2	28.4	29.0	28.4	31.4	31.1	32.2	31.3	28.9	28.7	29.3	28 .7
23	33.7	34.3	34.4	34.2	30.5	31.4	31.7	31.4	35.1	34.8	35.4	34.7	32.0	32.1	32.5	32 .0
24	29.3	29.1	30.6	29.8	26.5	26.6	27.5	26.8	29.7	29.3	30.9	29.9	27.1	26.9	27.8	27 .1
Avg.	31.4	31.6	32.3	31.8	28.4	28.6	29.0	28.6	32.3	31.9	33.0	32.1	29.3	29.1	29.8	29.2

Such structured noise is harder to remove than independent noise. We propose a method to directly denoise CFA data before demosaicking. Images from the denoise-first pipeline using our CFA denoising method are of similar quality to traditional demosaic-first pipelines although the denoising stage in the denoise-first pipeline requires at most $1/3^{\text{rd}}$ the computation of the denoising stage in the demosaic-first pipeline.

REFERENCES

[1] B. Bayer, "Color imaging array," U.S. Patent 3971065, July 1976.

[2] W.-C. Kao, S.-H. Wang, L.-Y. Chen, and S.-Y. Lin, "Design considerations of color image processing pipeline for digital cameras," *IEEE Transactions on Consumer Electronics*, vol. 52, no. 4, pp. 1144–1152, 2006.

[3] K. Hirakawa, X.-L. Meng, and P. Wolfe, "A framework for wavelet-based analysis and processing of color filter array images with applications to denoising and demosaicing," in *Acoustics, Speech and Signal Processing, 2007. ICASSP 2007. IEEE International Conference on*, vol. 1, April 2007, pp. I–597–I–600.

[4] L. Zhang, R. Lukac, X. Wu, and D. Zhang, "PCA-based spatially adaptive denoising of cfa images for single-sensor digital cameras," *IEEE Trans. Image Process.*, vol. 18, no. 4, pp. 797–812, 2009.

[5] A. Danielyan, M. Vehvilainen, A. Foi, V. Katkovnik, and K. Egiazarian, "Cross-color BM3D filtering of noisy raw data," in *Proc. International Workshop on Local and Non-Local Approximation in Image Processing LNLA 2009*, 2009, pp. 125–129.

[6] K. Hirakawa and T. W. Parks, "Adaptive homogeneity-directed demosaicing algorithm," *IEEE Transactions on Image Processing*, vol. 14, no. 3, pp. 360–369, 2005.

[7] E. Dubois, "Frequency-domain methods for demosaicking of Bayer-sampled color images," *IEEE Signal Process. Lett.*, vol. 12, no. 12, pp. 847–850, 2005.

[8] Eastman Kodak Company, "PhotoCD PCD0992," (http://r0k.us/graphics/kodak/).

[9] B. Gunturk, Y. Altunbasak, and R. Mersereau, "Color plane interpolation using alternating projections," *IEEE Trans. Image Process.*, vol. 11, no. 9, pp. 997–1013, 2002.

[10] J. Portilla, V. Strela, M. Wainwright, and E. Simoncelli, "Image denoising using scale mixtures of Gaussians in the wavelet domain," *IEEE Trans. Image Process.*, vol. 12, no. 11, pp. 1338–1351, 2003.

[11] K. Dabov, A. Foi, V. Katkovnik, and K. O. Egiazarian, "Image denoising by sparse 3-D transform-domain collaborative filtering," *IEEE Transactions on Image Processing*, vol. 16, no. 8, pp. 2080–2095, 2007.