

Second-Generation Color Filter Array and Demosaicking Designs

Keigo Hirakawa and Patrick J. Wolfe

School of Engineering and Applied Sciences
Department of Statistics, Harvard University
Oxford Street, Cambridge, MA 02138 USA

Abstract

Recent developments in spatio-spectral sampling theory for color imaging devices show that the choice of color filter array largely determines the spatial resolution of color images achievable by subsequent processing schemes such as demosaicking and image denoising. This paper highlights the cost-effectiveness of a new breed of color filter array patterns based on this sampling theory by detailing an implementation of the demosaicking method consisting of entirely linear elements and comprising a total of only ten add operations per full-pixel reconstruction. With color fidelity that rivals the state-of-the-art interpolation methods and an order of complexity near to that of the bilinear interpolation, this joint sensor-demosaicking solution to digital camera architectures can fulfill the image quality and complexity needs of future digital multimedia simultaneously.

Keywords: Color filter array, demosaicking, color imaging, digital multimedia

1. INTRODUCTION

The growing ubiquity of digital multimedia drives simultaneous demand for throughput and improved image quality in color imaging devices, underscoring the recent trend toward increasing spatial resolution via shrinking pixel sensor sizes. Although this can enhance the overall fidelity of the image, the excess sensor hardware and computational complexity spent for extra resolution in turn drive the overall cost and power consumption of imaging systems. Thus, a new design paradigm that permits a more favorable complexity-resolution trade-off represents a major step toward meeting the needs of future digital multimedia.

As described below, the authors' recent introduction of spatio-spectral sampling theory shows how it is possible to increase the spatial resolution of the color image sensor with *no* additional hardware and *less* computation. The color filter array (CFA) is part of an interface between the real world and the digital hardware, and hence it determines to a great extent the maximal resolution achievable by subsequent processing schemes. Though much industry and academic research is devoted to *demosaicking*—the process of reconstructing a spatially undersampled vector field whose components correspond to particular color tristimulus values—the analytical framework afforded by this new sampling theory exposed the inherent loss of resolution in the contemporary CFA designs.² Consequently, it comes as no surprise that increasing complexity in the camera pipeline, while using existing CFA patterns, often yields diminishing returns in terms of image quality.

In a break from the traditional approach, the spatio-spectral CFA design paradigm aims to maximally preserve resolution during acquisition, thereby retaining the integrity of the color image signal in the sensor data.^{2,3} Image data acquired in this way are therefore easily accessible, enjoy simple reconstruction schemes, and admit enhanced color fidelity when compared to the traditional approaches. The throughput gains to be had from a large computational savings—combined with state-of-the-art color fidelity owing to the significant increase of the spatial resolution in the image sensor—make the *joint sensor-demosaicking* solution an extremely cost-effective alternative to the current practices in portable digital still and video camera designs.

Patent Pending. Further author information: (Send correspondence to K.H.)

K.H.: E-mail: hirakawa@stat.harvard.edu

P.J.W.: E-mail: patrick@seas.harvard.edu

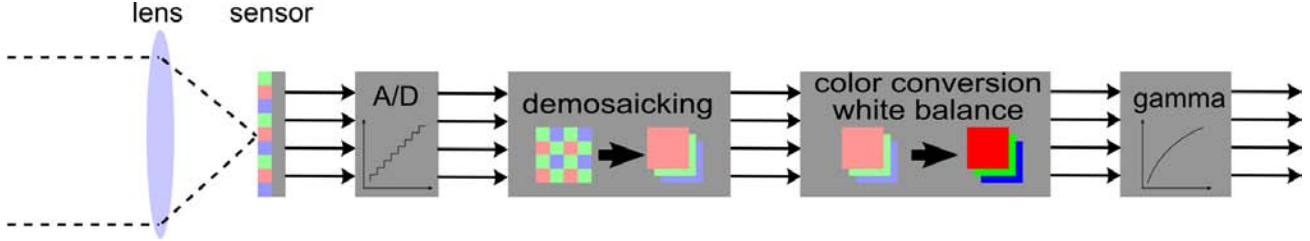


Figure 1. Illustration of a typical camera pipeline

The contribution of this paper is to provide full implementational details for a demosaicking method for spatio-spectral CFAs requiring a total of only ten add operations per full-pixel reconstruction—nearly comparable to the complexity seen in bilinear interpolation. This demosaicking method also contains no nonlinear elements, such as muxing or *if-then* operations that are costly to realize in both DSP and ASIC implementations. As described below, this computational savings comes in part from exploiting existing structures in the digital camera pipeline; the proposed architecture does *not* alter the standard interface, already established in the industry, between the camera and computers or printers.

This paper is organized as follows. A review of the current digital camera pipeline is first provided in Section 2. The theory of spatio-spectral CFA is reviewed in Section 3, and the design and implementation of the corresponding demosaicking algorithm are detailed in Section 4. Sections 5 and 6 comprise results and a concluding discussion, respectively.

2. CAMERA PIPELINE REVIEW

It is instructive to first review the internal workings of a digital camera, as the demosaicking method proposed in Section 4 takes advantage of pre-existing structure within the camera architecture. A number of signal processing steps or modules comprise a camera pipeline after the acquisition of data; for an excellent coverage of the digital camera pipeline, refer to Reference 4.

A typical pipeline is shown in Figure 1. Let $\mathbf{x}(\mathbf{n}) = [x_r(\mathbf{n}), x_g(\mathbf{n}), x_b(\mathbf{n})]^T \in \mathbb{R}^3$ be the RGB tristimulus value of the desired color image signal at pixel location indexed by $\mathbf{n} \in \mathbb{Z}^2$. Then, the key components of the camera pipeline are as follows:

1. CFA and Sensor

Once the light enters into the camera through the lens, it is focused onto the surface of sensor array. Each pixel sensor measures the intensity of the light that penetrates through the color filter, whose value is recorded digitally via the analog-to-digital (A/D) converter. This is effectively a spatio-spectral subsampling procedure implemented as a color filter array, whereby each pixel location measures only a portion of the visible spectrum selected from amongst a chosen “color partition” of that spectrum. That is, let

$$\mathbf{c}(\mathbf{n}) = [c_r(\mathbf{n}), c_g(\mathbf{n}), c_b(\mathbf{n})]^T \in \mathbb{R}^3, \quad 0 \leq c_r(\mathbf{n}), c_g(\mathbf{n}), c_b(\mathbf{n}) \leq 1$$

indicate the corresponding CFA color combination. Then the sensor measurement $y(\mathbf{n})$ is an inner product between the color filter and the image signal:

$$y(\mathbf{n}) = \mathbf{c}(\mathbf{n})^T \mathbf{x}(\mathbf{n}). \quad (1)$$

Detailed analysis of $y(\mathbf{n})$ is provided in Reference 2 and it is briefly summarized below in Section 3. See Figure 2 for example CFA patterns.

2. Demosaicking

The spatial subsampling is approximately inverted through demosaicking—the subject of this paper, which we elaborate on in Section 4—yielding a *complete* tristimulus value at each pixel location. For now, let $\hat{\mathbf{x}}_y$ denote the demosaicking estimate of the complete color image \mathbf{x} given y .

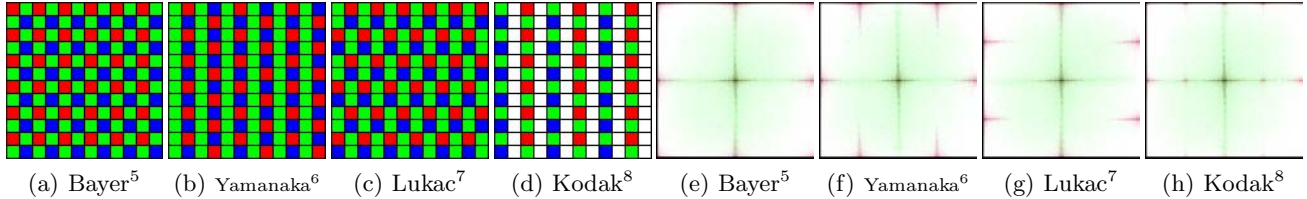


Figure 2. Examples of existing CFA patterns (a–d) and corresponding log-magnitude spectra of typical sensor data (e–h)

3. Color Space Conversion/White Balance

Because the color coordinates defined by the sensitivity of the color filters \mathbf{c} may not correspond exactly with the *standardized* color space (such as sRGB space), the resulting tristimulus values undergo a color space conversion (change of basis) via pixel-wise multiplication by a predetermined matrix $\mathbf{M} \in \mathbb{R}^{3 \times 3}$:

$$\mathbf{z}(\mathbf{n}) = \mathbf{M}\hat{\mathbf{x}}_y(\mathbf{n}). \quad (2)$$

Additional color space conversion may be required for image compression, which usually operates in an opponent color space. Linear white-balance correction—adjustment to the color to account for variations in illuminant and the environment—is typically made either before demosaicking or concurrently with the color space conversion in (2).

4. Gamma Correction

A point-wise nonlinearity termed the inverse gamma function $\Gamma^{-1} : \mathbb{R} \rightarrow \mathbb{R}$ is applied to the color-corrected tristimulus value \mathbf{z} to yield the display stimulus $\mathbf{u} = [u_1, u_2, u_3]^T$:

$$u_i(\mathbf{n}) = \Gamma^{-1}\{z_i(\mathbf{n})\}.$$

This gamma correction step will *undo* the effects of nonlinearity Γ inherent in display devices (i.e., $\Gamma(\mathbf{u})$ is linear with respect to \mathbf{z}).

Note that demosaicking is typically the only module that deals with the spatial aspects of the image data, thus making it by far the most computationally intensive module; other operations are point-wise or pixel-wise.

The processing pipeline for a video sequence requires extra care, as it must handle temporal information. To meet the required frame rate and increased data volume, it is not atypical to use less sophisticated demosaicking methods such as bilinear interpolation, and/or implement the processing pipeline using dedicated hardware (ASIC). Although signal distortions can be masked by motion blur and compression to a degree (in contrast to the case of still-image reconstruction), the loss of image quality due to simple demosaicking is unambiguously noticeable. Moreover, nonlinear demosaicking steps such as edge detection may introduce a pixel flickering problem, a type of inter-frame oscillatory effect caused by the data-dependent detection variable’s instability and susceptibility to noise. In such cases, a pixel (often at the edges or textural regions) appears to flicker or alternate between two values, and causes a loss in quality noticeable even to the casual observer. Two common ways to fix this problem are to incorporate redundancies across frames, or to eliminate nonlinearity—with the latter solution preferred as the former requires a frame buffer. Furthermore, the frame rate determines the transistor reset interval; reducing the number of photons encountered by the sensor in each frame compromises the signal-to-noise ratio. To improve their robustness to noise, color filter arrays in video cameras are often panchromatic, or sensitive to wideband electromagnetic spectra to allow more light to reach the sensor.

In an ASIC architecture where signal processing steps are performed in tandem, each module is synchronized with the rate of the processing at the slowest module. Therefore, in most cases throughput is increased by introducing hardware redundancy at this slowest module. In DSP chip solutions, each module contributes to the overall execution time—throughput is therefore increased by speeding up the clock rate. In either implementation paradigm, reduction of complexity in demosaicking will therefore speed the overall system, reduce the clock rate to save power, lower gate counts, increase the video frame rate, and allow for larger frame sizes.

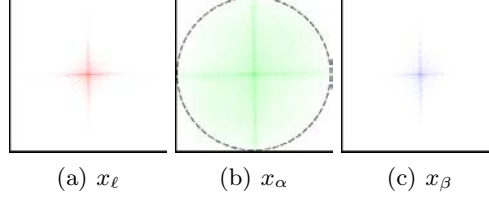


Figure 3. Log-magnitude spectra of a typical color image, with the circle of (b) indicating full sensor resolution

3. REVIEW OF SPATIO-SPECTRAL SAMPLING THEORY

In this section, we briefly review the requisite tools needed to identify aliasing structures at the acquisition interface that preclude full sensor resolution—full details are disclosed in a prior publication.² We first summarize the spatio-spectral analysis framework and then use it to identify principal differences between existing CFA patterns and our new panchromatic designs.

3.1. Analysis of Sensor Data

Recalling the sensor measurement formulation of (1), we have that in the case of pure-color (i.e., RGB) color filter arrays, $\mathbf{c}(\mathbf{n}) \in \{[1, 0, 0]^T, [0, 1, 0]^T, [0, 0, 1]^T\}$, whereas in the case of panchromatic or spatio-spectral designs, each value represents a *mixture* of colors. Suppose further that $c_r + c_g + c_b = 1$ (true by default for pure-color CFAs); define $c_\alpha(\mathbf{n}) = c_r(\mathbf{n}) - k_\alpha$ and $c_\beta(\mathbf{n}) = c_b(\mathbf{n}) - k_\beta$, where k_α and k_β are the (constant) DC components of c_r and c_b , respectively. Then (1) simplifies as follows:

$$\begin{aligned}
 \mathbf{y}(\mathbf{n}) &= \mathbf{c}(\mathbf{n})^T \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 1 \end{bmatrix} \begin{bmatrix} 1 & -1 & 0 \\ 0 & 1 & 0 \\ 0 & -1 & 1 \end{bmatrix} \mathbf{x}(\mathbf{n}) \\
 &= [c_r(\mathbf{n}) \quad 1 \quad c_b(\mathbf{n})] \begin{bmatrix} x_\alpha(\mathbf{n}) \\ x_g(\mathbf{n}) \\ x_\beta(\mathbf{n}) \end{bmatrix} \\
 &= [c_r(\mathbf{n}) \quad 1 \quad c_b(\mathbf{n})] \begin{bmatrix} 1 & 0 & 0 \\ -k_\alpha & 1 & -k_\beta \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ k_\alpha & 1 & k_\beta \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_\alpha(\mathbf{n}) \\ x_g(\mathbf{n}) \\ x_\beta(\mathbf{n}) \end{bmatrix} \\
 &= [c_\alpha(\mathbf{n}) \quad 1 \quad c_\beta(\mathbf{n})] \begin{bmatrix} x_\alpha(\mathbf{n}) \\ x_\ell(\mathbf{n}) \\ x_\beta(\mathbf{n}) \end{bmatrix}, \tag{3}
 \end{aligned}$$

where $x_\alpha = x_r - x_g$ and $x_\beta = x_b - x_g$ are difference images, and $x_\ell = x_g + k_\alpha x_\alpha + k_\beta x_\beta$ represents a baseline sum image. The advantage of the $\{x_\alpha, x_\ell, x_\beta\}$ representation is that these difference images can serve as a proxy for chrominance, and enjoy rapid decay in the frequency domain, whereas x_ℓ can be taken to represent luminance, which embodies edge and texture information. Figure 3, which illustrates the spectral support of $x_\alpha, x_\ell, x_\beta$ for a typical color image, verifies empirically that the bandwidth of x_ℓ is much higher than that of the modulated difference images x_α, x_β . In fact, the physical spatial resolution of the image sensor is precisely defined by the radius around the origin in Figure 3(b).

A point-wise multiplication between two signals in the time domain is equivalent to a convolution in the two-dimensional Fourier domain. Let us assume c_α and c_β are the finite sums of sinusoids (equivalent to assuming that the CFA has a periodic pattern); the Fourier transforms of c_α and c_β can thus be written as:

$$\mathcal{F}c_\alpha = \sum_{i=1}^I s_i \delta(\boldsymbol{\omega} - \boldsymbol{\lambda}_i), \quad \mathcal{F}c_\beta = \sum_{i=1}^I t_i \delta(\boldsymbol{\omega} - \boldsymbol{\lambda}_i), \tag{4}$$

where $\{\lambda_i, s_i, t_i\}$ is determined by the choice of CFA pattern. Then $c_\alpha \cdot x_\alpha$ and $c_\beta \cdot x_\beta$ represent “modulations” (or shifts) in frequency by the “carrier frequencies” λ_i :

$$\mathcal{F}\{c_\alpha \cdot x_\alpha\} = \sum_i s_i \mathcal{F}x_\alpha(\omega - \lambda_i), \quad \mathcal{F}\{c_\beta \cdot x_\beta\} = \sum_i t_i \mathcal{F}x_\beta(\omega - \lambda_i).$$

The acquisition process in (3) hence is a sum of baseline signal x_ℓ and the modulated difference images $c_\alpha \cdot x_\alpha$ and $c_\beta \cdot x_\beta$, where c_α and c_β determine the carrier frequencies λ_i .

3.2. Aliasing in Pure-Color CFA Designs

Aliasing occurs whenever the spectral supports of the modulated difference images $c_\alpha \cdot x_\alpha$ and $c_\beta \cdot x_\beta$ overlap with the spectral support of the baseline signal x_ℓ —in which case one signal is “contaminating” the other. More concretely, sensor data suffers from aliasing when, for some fixed ω , we have that both $\mathcal{F}x_\ell(\omega) \neq 0$ and $\mathcal{F}x_\alpha(\omega - \lambda_i) \neq 0$ for some i (and likewise with $\mathcal{F}x_\ell(\omega)$ and $\mathcal{F}x_\beta(\omega - \lambda_i)$). Barring any additional assumptions about the structure of the color image signal, aliasing prevents perfect reconstruction of the full-color image. Because the spectral support of x_ℓ is centered around the origin (DC), it is then obvious that we would like the modulation frequency λ_i to be as “high-frequency” as possible to reduce this risk of aliasing. Unfortunately, it has been proven that λ_i as determined by *any* pure-color CFA patterns includes low frequencies—see Reference 2 for a precise statement of this fact and its proof. In fact, well-studied patterns such as Bayer,⁵ Yamanaka,⁶ Lukac,⁷ and Kodak⁸ are inherently sub-optimal in this regard, because they are subject to severe aliasing. As illustrated in Figure 2, spectra of simulated sensor data y captured with these CFA patterns bear out this claim.

In light of these shortcomings, a digital camera designed with pure-color CFA patterns must resort to one or more of the following three measures, precisely corresponding to the complexity-resolution trade-offs indicated earlier:

1. Reduce Camera Resolution

Limiting the spatial resolution of the camera reduces the risk of aliasing. Known commonly as “anti-aliasing,” attenuating the high frequency components in x_ℓ will result in shrinking the radius of the largest concentration of energy centered about the origin in Figure 2—thereby allowing sufficient separation between $\mathcal{F}x_\ell(\omega)$ and $\mathcal{F}x_\alpha(\omega - \lambda_i)$ or $\mathcal{F}x_\beta(\omega - \lambda_i)$. Although implementing anti-aliasing is as simple as tuning the optics slightly, the output images are effectively lowpassed and are unacceptable for many applications.

2. Increase Pixel Count

Densely populated pixel sensors yield a higher spatial sampling rate. The resiliency to aliasing is improved without compromising spatial resolution (afforded by the increased distance between $\mathcal{F}x_\ell(\omega)$ and $\mathcal{F}x_\alpha(\omega - \lambda_i)$ or $\mathcal{F}x_\beta(\omega - \lambda_i)$). However, the overall cost of the camera system will rise in proportion to the required extra sensor hardware and computational complexity.

3. Reconstruct with Nonlinear Methods

Reconstruction from aliased sensor is an ill-posed problem, and stronger assumptions about the signal are needed to recover the full-color image. Here the key idea is that conditioned on the “correct” identification of this structure—for example, the “smooth along edges, rough across edges” principle—aliasing can be avoided in some cases. Such reconstruction methods are data-driven and inherently nonlinear, adding extra complexity to the demosaicking method (though this is typically regarded as less expensive compared to increasing the pixel count). Moreover, violations of the underlying assumptions and sensitivities of the resultant decision variables both contribute to gross artifacts and numerical instabilities, in addition to the flickering effects in video described earlier. The difficulty of modeling the effects of nonlinearity on noise also complicates a post-demosaicking stage of image denoising.

In each case it may be seen that avoiding or reducing the risk of aliasing involves loss of resolution and/or added complexity.

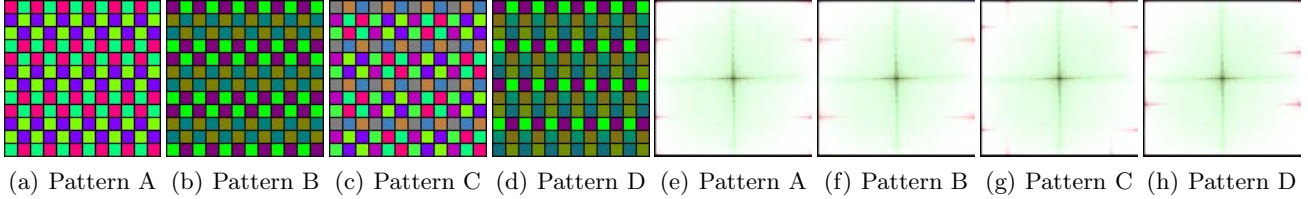


Figure 4. New CFA patterns² (a–d) and corresponding log-magnitude image spectra (e–h)

3.3. Spatio-Spectral CFA Design

The objective of the new spatio-spectral CFA design paradigm is to reduce the risk of aliasing without increasing the number of pixels, precisely through the use of new CFA patterns specified directly in the frequency domain. Recall (4), and suppose that $\{\lambda_i, s_i, t_i\}$ are interpreted as “design parameters” to be chosen directly—so that we may then derive the color filter array in its final form via the inverse Fourier transform. When the carrier frequencies λ_i correspond to high frequencies, we expect to obtain a partitioning of the captured color image data in the frequency domain, by forcing maximal distance between the baseband and the modulated difference images. The design aspects of choosing $\{\lambda_i, s_i, t_i\}$ to minimize aliasing are discussed in detail in References 2,3 and will not be covered here. Note that the resultant CFA patterns are necessarily panchromatic,² as shown in Figure 4.

Using these designs, the digital camera pipeline benefits from alias-free color image sensing in addition to a number of other advantages:

1. High Spatial Resolution

An alias-free color image sensor device has high spatial resolution because no anti-aliasing measures need to be taken. As can be confirmed from Figure 4, the radius of the spectral support around the baseband remains large, and consequently the color fidelity is improved significantly (see also Section 5).

2. Low Computational Complexity

An important benefit to alias-free acquisition is the low complexity of the reconstruction method. A straightforward *completely linear* demodulation technique can decode the color image data partitioned in the Fourier domain. In particular, nonlinear estimation techniques are unnecessary because the new sensor is far less sensitive to the directionality of image features.

3. Linearity of Reconstruction Method

A linear reconstruction method is numerically stable because it lacks data-dependent decision variables (e.g., to determine directionality), and video data processing no longer requires frame buffers. Furthermore, linear processing gives rise to the possibilities for post-demosaicking image denoising.

4. Improved Signal-to-Noise Ratio

Light intensity measurements are subject to signal-dependent noise, and this phenomenon becomes more apparent in photon-limited environments (e.g., indoor photography). The panchromaticity of these new CFA patterns helps to let more light through, and therefore increases robustness to noise.

Overall, the complexity-resolution trade-off appears highly favorable, because neither nonlinear reconstruction methods nor anti-aliasing measures need be taken in the absence of aliasing at the image sensor.

4. EFFICIENT LINEAR DEMOSAICKING

We now turn to the implementational details of a linear demosaicking methodology to accompany spatio-spectral CFA designs. The most general reconstruction method is presented first, followed by strategies for particular CFA patterns that further simplify the computational complexity. As a special case, we show that 4×4 CFA patterns admit an extremely efficient reconstruction method, requiring a total of only ten add operations per full-pixel reconstruction.

4.1. Demosaicking for General Case

The key observation in the previous section was that spatio-spectral CFA designs significantly reduce the risk of aliasing. Assuming a perfect partitioning of the color image data in the frequency domain, a modulated signal can be recovered by a multiplication by the carrier frequency followed by an ideal lowpass filter. Thus we expect an exact reconstruction via a standard demodulation (so-called ‘‘AM demodulation’’):

$$\hat{\mathbf{x}}_y(\mathbf{n}) = \begin{bmatrix} \hat{x}_r(\mathbf{n}) \\ \hat{x}_g(\mathbf{n}) \\ \hat{x}_b(\mathbf{n}) \end{bmatrix} = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ -k_\alpha & 1 & -k_\beta \\ 0 & 0 & 1 \end{bmatrix} \left(\begin{bmatrix} s_1 & 0 & t_1 \\ \vdots & \vdots & \vdots \\ s_I & 0 & t_I \\ 0 & 1 & 0 \end{bmatrix}^\# \right) \begin{bmatrix} h_\alpha * \{\phi_{\lambda_1} y\} \\ \vdots \\ h_\alpha * \{\phi_{\lambda_I} y\} \\ h_\ell * y \end{bmatrix}, \quad (5)$$

where ‘‘*’’ is a discrete convolution operator; the passbands of the ideal lowpass filters h_α and h_ℓ match the respective bandwidths of the signals x_α and x_ℓ (assuming the spectral supports of x_α and x_β are identical); I denotes the total number of carrier frequencies and $\cdot^\#$ the pseudo-inverse; and ϕ_{λ_i} represents a pure sinusoid with frequency λ_i :

$$\phi_{\lambda_i}(\mathbf{n}) = \mathcal{F}^{-1} \delta(\boldsymbol{\omega} - \boldsymbol{\lambda}_i) = (2\pi)^{-1} e^{j\boldsymbol{\lambda}_i^T \mathbf{n}}.$$

In practice, a crude lowpass filter with a modest cutoff frequency is sufficient for demodulation in (5) when the modulated signals are *well partitioned* in the frequency domain—we will revisit this point in Section 4.3.

4.2. Simplified Demosaicking for Orthogonal Color Filter Array

Suppose the conjugate carrier frequencies $\bar{c}_\alpha(\mathbf{n}) = c_\alpha^{-1}(\mathbf{n})$ and $\bar{c}_\beta(\mathbf{n}) = c_\beta^{-1}(\mathbf{n})$ exist (i.e., no zero elements). When the carrier frequencies are orthogonal, a modulated signal can be recovered by a multiplication by the conjugate carrier frequency followed by a lowpass filter:

$$\hat{\mathbf{x}}_y(\mathbf{n}) = \begin{bmatrix} \hat{x}_r(\mathbf{n}) \\ \hat{x}_g(\mathbf{n}) \\ \hat{x}_b(\mathbf{n}) \end{bmatrix} = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ -k_\alpha & 1 & -k_\beta \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} h_\alpha * \{\bar{c}_\alpha y\} \\ h_\ell * y \\ h_\beta * \{\bar{c}_\beta y\} \end{bmatrix}, \quad (6)$$

Because of the assumed partitioning, and by extension the mutual exclusivity of the signals in the Fourier domain, we may assume $c_\alpha h_\alpha + h_\ell + c_\beta h_\beta = \delta$, where $\delta(\mathbf{n})$ is the Kronecker delta function. Then the convolution $h_\ell * y$ can be rewritten as:

$$\begin{aligned} h_\ell * y &= (\delta - c_\alpha h_\alpha - c_\beta h_\beta) * y \\ &= y - \{c_\alpha h_\alpha\} * y - \{c_\beta h_\beta\} * y \\ &= y - c_\alpha \{h_\alpha * \{\bar{c}_\alpha y\}\} - c_\beta \{h_\beta * \{\bar{c}_\beta y\}\}. \end{aligned}$$

The demodulation in (6) simplifies to:

$$\begin{aligned} \hat{\mathbf{x}}_y(\mathbf{n}) &= \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ -k_\alpha & 1 & -k_\beta \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ -c_\alpha(\mathbf{n}) & 1 & -c_\beta(\mathbf{n}) \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} h_\alpha * \{\bar{c}_\alpha y\} \\ y \\ h_\alpha * \{\bar{c}_\beta y\} \end{bmatrix} \\ &= \begin{bmatrix} 1 - k_\alpha - c_\alpha(\mathbf{n}) & 1 & -k_\beta - c_\beta(\mathbf{n}) \\ -k_\alpha - c_\alpha(\mathbf{n}) & 1 & -k_\beta - c_\beta(\mathbf{n}) \\ -k_\alpha - c_\alpha(\mathbf{n}) & 1 & 1 - k_\beta - c_\beta(\mathbf{n}) \end{bmatrix} \begin{bmatrix} h_\alpha * \{\bar{c}_\alpha y\} \\ y \\ h_\alpha * \{\bar{c}_\beta y\} \end{bmatrix} \\ &= \begin{bmatrix} 1 - c_r(\mathbf{n}) & 1 & -c_b(\mathbf{n}) \\ -c_r(\mathbf{n}) & 1 & -c_b(\mathbf{n}) \\ -c_r(\mathbf{n}) & 1 & 1 - c_b(\mathbf{n}) \end{bmatrix} \begin{bmatrix} h_\alpha * \{\bar{c}_\alpha y\} \\ y \\ h_\alpha * \{\bar{c}_\beta y\} \end{bmatrix}, \quad (7) \end{aligned}$$

and the computational savings comes from having to perform the convolution operation only twice.

4.3. Efficient Demosaicking for 4×4 CFA Patterns

Pattern Analysis and Reconstruction

Decoding sensor data captured with 4×4 CFA patterns is surprisingly efficient. Owing to the restricted choice of sinusoids— $(\pi, \pm\frac{\pi}{2})$, $(\pm\frac{\pi}{2}, \pi)$, (π, π) comprise all possible carrier frequencies—the space of all possible color filters c_α and c_β are spanned by an orthogonal basis $\{\psi_1, \dots, \psi_5\}$ comprised of sequences of ± 1 :

$$\{\psi_1(\mathbf{n}) = -1^{n_1+n_2}, \psi_2(\mathbf{n}) = -1^{\lfloor n_1/2 \rfloor + n_2}, \psi_3(\mathbf{n}) = -1^{\lceil n_1/2 \rceil + n_2}, \psi_4(\mathbf{n}) = -1^{n_1 + \lfloor n_2/2 \rfloor}, \psi_5(\mathbf{n}) = -1^{n_1 + \lceil n_2/2 \rceil}\},$$

where $\lfloor \cdot \rfloor$ and $\lceil \cdot \rceil$ operators denote rounding down and up, respectively. Thus, we can rewrite c_α and c_β as:

$$c_\alpha = \sum_{i=1}^5 p_i \psi_i(\mathbf{n}), \quad c_\beta = \sum_{i=1}^5 q_i \psi_i(\mathbf{n}).$$

In this section, we take advantage of the sequences of ± 1 to derive efficient demodulation scheme. Suppose the new CFA pattern is designed with an arithmetic average of two distinct ψ_i 's:²

$$\begin{aligned} \theta_\alpha(\mathbf{n}) &= \frac{\psi_{i_1}(\mathbf{n}) + v_\alpha \psi_{i_2}(\mathbf{n})}{2} \\ \theta_\beta(\mathbf{n}) &= \frac{\psi_{j_1}(\mathbf{n}) + v_\beta \psi_{j_2}(\mathbf{n})}{2} \\ c_\alpha &= p\theta_\alpha, \quad c_\beta = q\theta_\beta, \end{aligned}$$

where p, q are constants, $i_1 \neq i_2$ and $j_1 \neq j_2$, and $v_\alpha, v_\beta \in \{\pm 1\}$. It is easy to verify the following properties:

- $\theta_\alpha, \theta_\beta \in \{0, \pm 1\}$, where the entries with 0 comprise half of the samples.
- $\theta_\alpha^2, \theta_\beta^2 \in \{0, 1\}$, where the entries with 0 comprise half of the samples.

If c_α and c_β are orthogonal, an efficient alternative to (7) is:

$$\hat{\mathbf{x}}_y(\mathbf{n}) = \underbrace{\begin{bmatrix} (2 - 2c_r(\mathbf{n}))/p & 1 & -(2c_b(\mathbf{n}))/q \\ -(2c_r(\mathbf{n}))/p & 1 & -(2c_b(\mathbf{n}))/q \\ -(2c_r(\mathbf{n}))/p & 1 & (2 - 2c_b(\mathbf{n}))/q \end{bmatrix}}_{\text{pixel-wise operation}} \underbrace{\begin{bmatrix} h_\alpha * \{\theta_\alpha y\} \\ y \\ h_\alpha * \{\theta_\beta y\} \end{bmatrix}}_{\text{spatial processing}}. \quad (8)$$

The spatial processing of demosaicking in its entirety is now contained in the second term of (8). The first term in (8) is a 3×3 matrix multiplication, which is a completely pixel-wise operation that can be combined with \mathbf{M} in (2) *at no cost*:

$$\mathbf{z}(\mathbf{n}) = \mathbf{M} \hat{\mathbf{x}} = \mathbf{M} \underbrace{\begin{bmatrix} (2 - 2c_r(\mathbf{n}))/p & 1 & -(2c_b(\mathbf{n}))/q \\ -(2c_r(\mathbf{n}))/p & 1 & -(2c_b(\mathbf{n}))/q \\ -(2c_r(\mathbf{n}))/p & 1 & (2 - 2c_b(\mathbf{n}))/q \end{bmatrix}}_{\text{can be combined}} \begin{bmatrix} h_\alpha * \{\theta_\alpha y\} \\ y \\ h_\alpha * \{\theta_\beta y\} \end{bmatrix}.$$

The combined matrix can be precomputed offline—and so the computational cost of the first term in (8) is absorbed into the color conversion module. Therefore, the complexity of the demosaicking module is the cost of implementing the second term in (8).

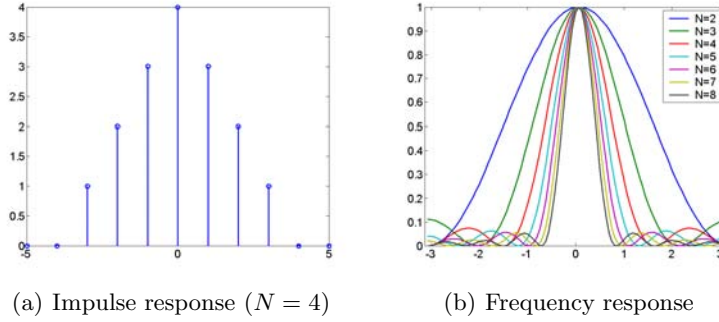


Figure 5. Responses of the triangle filters used for efficient linear demosaicking

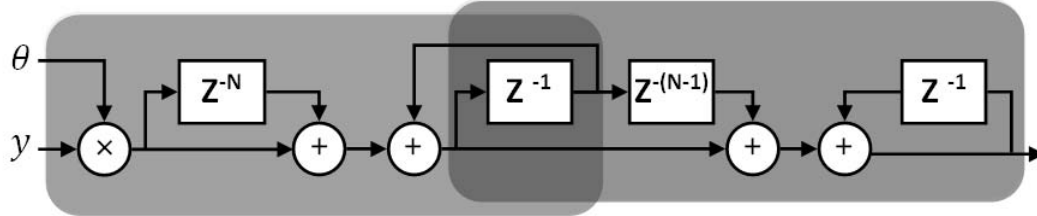


Figure 6. Implementation of a one-dimensional structure $h_{\text{tri}} * \{\theta y\}$, with shaded regions corresponding to boxcar filters

Triangle Lowpass Filter

We are now left with the task of designing an efficient lowpass filter h_α . With sufficient separation between the baseband and modulated signals in the frequency domain, a crudely designed lowpass filter h_α suffice for the task. Here we propose to implement h_α as a separable two-dimensional filter, where a “triangle filter” is applied both horizontally and vertically. As illustrated in Figure 5, the triangle filter is linear-phase filter with a modest cutoff in the frequency domain—and is sufficient for demodulation in (8) when the modulated signals are well partitioned in the frequency domain.

A length $2N - 1$ triangle filter can be implemented by cascading two boxcar filters, each of which has the following one-dimensional Z transform:

$$H_{\text{box}}(z) = \sum_{n=0}^{N-1} z^{-n} = \frac{1 - z^{-N}}{1 - z^{-1}}.$$

The numerator and denominator of this Z transform can each be implemented with a single adder. Thus, the computational complexity of the triangle filter $H_{\text{tri}}(z) = H_{\text{box}}(z)^2$ —pictured in Figure 6—is 4 adders per pixel, and the required “delay lines” are of length $2N + 1$. In two dimensions, the separable triangle filter h_α has a multidimensional Z transform given by:

$$\begin{aligned} H_\alpha(\mathbf{z}) &= \left(\frac{1 - z_1^{-N}}{1 - z_1^{-1}} \right) \left(\frac{1 - z_1^{-N}}{1 - z_1^{-1}} \right) \left(\frac{1 - z_2^{-N}}{1 - z_2^{-1}} \right) \left(\frac{1 - z_2^{-N}}{1 - z_2^{-1}} \right) \\ &= H_{\text{tri}}(z_1)H_{\text{tri}}(z_2), \end{aligned} \quad (9)$$

where z_1 and z_2 correspond to delay lines in horizontal and vertical directions. Realization of (8) based on a literal implementation of (9) therefore requires application of the filter structure in Figure 6 four times—so that the overall complexity is 16 adders per full-pixel reconstruction and $8N + 4$ delay lines.

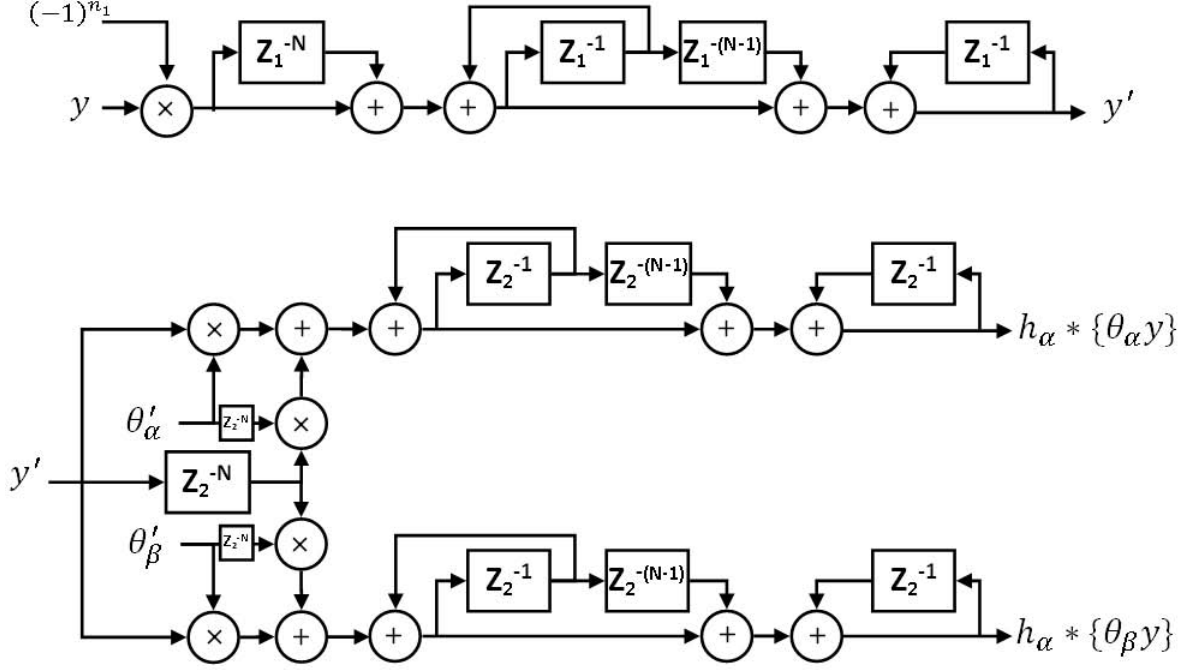


Figure 7. Block diagram of the proposed demosaicking method for a 4×4 CFA pattern. The structure at the top represents (10), whereas the structure on the bottom shows (11) and (12). In an ASIC implementation, Z_1 corresponds to a line buffer and Z_2 is a register. The signal $\theta'_\alpha y'$ and $\theta'_\beta y'$ contain many zero entries—eliminating the need to perform several of the “add” operations above—and the delay line on θ'_α and θ'_β can be hard-coded

Implementation of Demosaicking

Further hardware optimization is often possible by exploiting details specific to the CFA design. While it is impractical to show details for every configuration in this paper, we study the demosaicking method for CFA pattern in Figure 4(a) as an illustrative example, where $p = q = 1/4$, $(i_1, i_2) = (j_1, j_2) = (4, 1)$, and $v_\alpha = 1$, $v_\beta = -1$. Isolating the redundancies between $h_\alpha * \{\theta_\alpha y\}$ and $h_\alpha * \{\theta_\beta y\}$, we perform demosaicking according to (8) in stages:

$$y' = h_{\text{tri}}(n_1) * \{(-1)^{n_1} y\} \quad (10)$$

$$h_\alpha * \{\theta_\alpha y\} = h_{\text{tri}}(n_2) * \{\theta'_\alpha(n_2) y'(\mathbf{n})\} \quad (11)$$

$$h_\beta * \{\theta_\beta y\} = h_{\text{tri}}(n_2) * \{\theta'_\beta(n_2) y'(\mathbf{n})\}, \quad (12)$$

where

$$\theta'_\alpha(n) = \frac{(-1)^{\lfloor n/2 \rfloor} + (-1)^n}{2} = [\dots, 0, 0, 1, -1, 0, 0, 1, -1, \dots]$$

$$\theta'_\beta(n) = \frac{(-1)^{\lfloor n/2 \rfloor} - (-1)^n}{2} = [\dots, 1, -1, 0, 0, 1, -1, 0, 0, \dots].$$

The computational complexity of the triangle filter in (10) is 4 adds per full-pixel reconstruction and $2N + 1$ delay lines, whereas the complexities of (11) and (12) are 3 adds and $3N/2 + 1$ delay lines each—with the extra savings coming from the “0 entries” in $\theta'_\alpha y'$ and $\theta'_\beta y'$ that eliminate the need to compute or store them during filtering. In total, the overall cost of this implementation is 10 add operations per full-pixel reconstruction and $5N + 3$ delay lines. In ASIC designs, line buffers are typically more expensive than the registers; (10-12) represent $2N + 1$ line buffers and $3N + 2$ flip-flop registers.

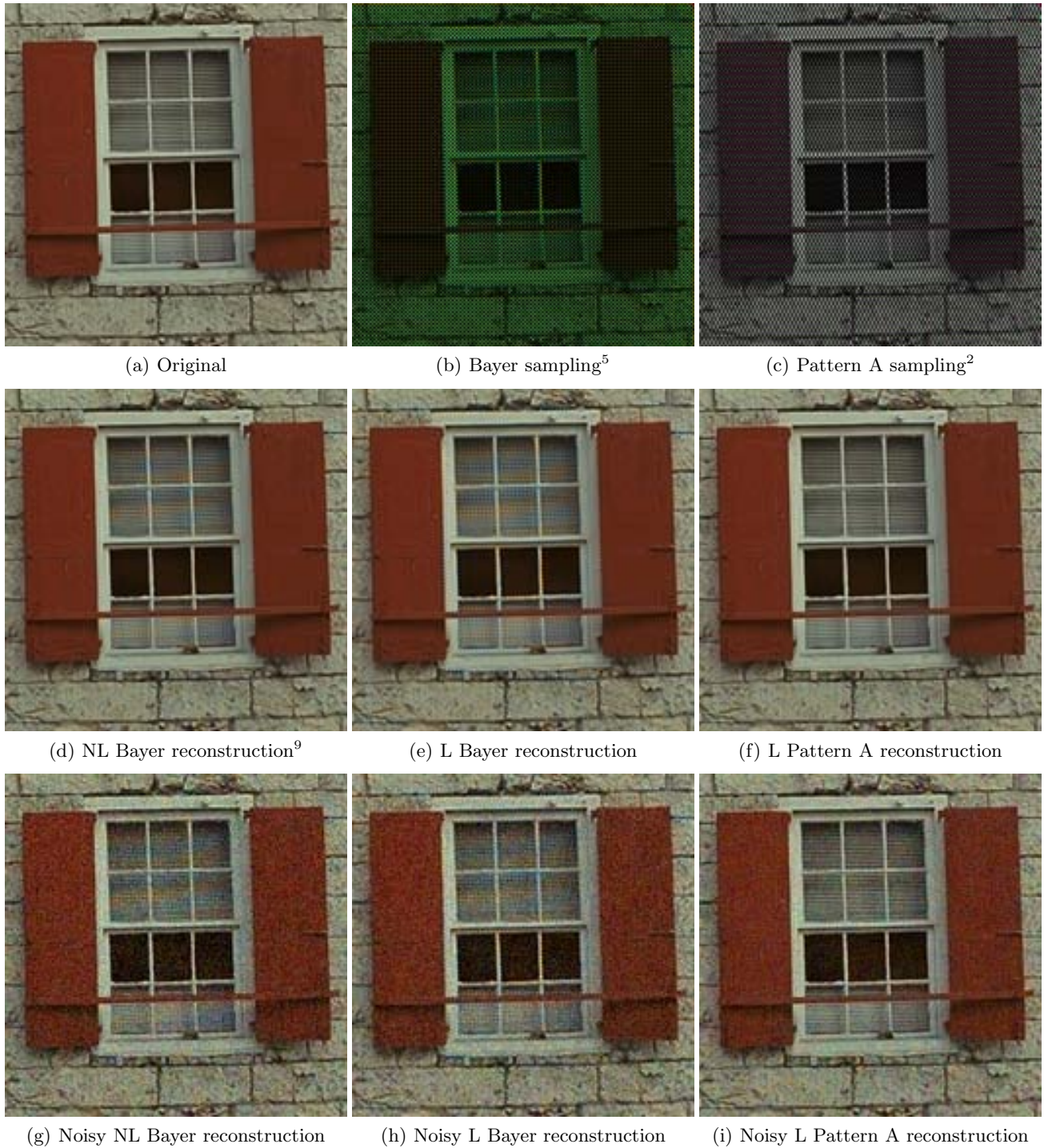


Figure 8. Sampled sensor data (top row) corresponding to a Kodak test image, with nonlinear⁹ (NL) and linear (L) reconstruction methods shown for the case of clean (middle row) and noisy (bottom row) sensor data

5. RESULTS

Example output images are shown in Figure 8—the proposed scheme used $N = 4$, which was observed empirically to work well. The reconstructions in Figures 8(d-f) indicate that the spatio-spectral CFA designs are aliasing-free, and therefore that a demosaicking scheme with the order of complexity near that of the bilinear interpolation can easily outperform state-of-the-art demosaicking methods. Figures 8(g-i) also demonstrate the resiliency of the spatio-spectral CFA patterns to noise. More experimental results are found in References 2, 3.

6. DISCUSSION

Our proposed joint sensor-demosaicking solution to digital camera architecture assigns carrier frequencies away from the origin in the spatial Fourier plane, thereby exploiting the regions of the spectrum unoccupied by pure-color CFA patterns. Furthermore, the separation of spatial and pixel-wise operation in demosaicking admits significant complexity savings by combining matrix multiplication with the color conversion module. As a result, the proposed scheme achieving state-of-the-art output image quality with nominal complexity totalling just ten add operations per full pixel reconstruction. This joint sensor-demosaicking solution is thus well positioned to be the foundation for the next generation digital cameras.

REFERENCES

1. K. Hirakawa and P. J. Wolfe, “Spatio-spectral color filter array design for optimal image recovery,” *To be submitted for review: IEEE Trans. Image Process.* , 2007.
2. K. Hirakawa and P. J. Wolfe, “Spatio-spectral color filter array design for enhanced image fidelity,” *Proc. IEEE Int. Con. Image Process.* , September 2007.
3. K. Hirakawa and P. J. Wolfe, “Fourier domain display color filter array design,” *Proc. IEEE Int. Con. Image Process.* , September 2007.
4. K. Parulski and K. E. Spaulding, “Color image processing for digital cameras,” in *Digital Color Imaging Handbook*, G. Sharma, ed., pp. 728–757, CRC Press, Boca Raton, FL, 2003.
5. B. E. Bayer, “Color imaging array.” US Patent 3 971 065, 1976.
6. S. Yamanaka, “Solid state color camera.” US Patent 4 054 906, 1977.
7. R. Lukac and K. N. Plataniotis, “Color filter arrays: Design and performance analysis,” *IEEE Transactions on Consumer Electronics* **51**, pp. 1260–1267, 2005.
8. T. Kijima, H. Nakamura, J. Compton, and J. Hamilton, “Image sensor with improved light sensitivity.” US Patent 20 070 177 236, 2007.
9. B. K. Gunturk, J. Glotzbach, Y. Altunbasak, R. W. Schafer, and R. M. Mersereau, “Demosaicking: Color filter array interpolation in single chip digital cameras,” *IEEE Signal Processing Magazine* **22**, pp. 44–54, 2005.