

Rethinking Color Cameras

Ayan Chakrabarti
 Harvard University
 Cambridge, MA

ayanc@eecs.harvard.edu

William T. Freeman
 Massachusetts Institute of Technology
 Cambridge, MA

billf@mit.edu

Todd Zickler
 Harvard University
 Cambridge, MA

zickler@seas.harvard.edu

Abstract

Digital color cameras make sub-sampled measurements of color at alternating pixel locations, and then “demosaick” these measurements to create full color images by up-sampling. This allows traditional cameras with restricted processing hardware to produce color images from a single shot, but it requires blocking a majority of the incident light and is prone to aliasing artifacts. In this paper, we introduce a computational approach to color photography, where the sampling pattern and reconstruction process are co-designed to enhance sharpness and photographic speed. The pattern is made predominantly panchromatic, thus avoiding excessive loss of light and aliasing of high spatial-frequency intensity variations. Color is sampled at a very sparse set of locations and then propagated throughout the image with guidance from the un-aliased luminance channel. Experimental results show that this approach leads to significant reductions in noise and aliasing artifacts, especially in low-light conditions.

panchromatic: sensitive to all colors

1. Introduction

The standard practice for one-shot digital color photography is to include a color filter array in front of the sensor that samples the three color channels in a dense alternating mosaic. The most common alternating mosaic is the [Bayer pattern \[1\]](#) shown in the top of Fig. 1. Full color images are then reconstructed from these samples through an up-sampling process known as [demosaicking](#). In general, this approach is quite functional, and with support from advances in demosaicking algorithms [16], Bayer-like color cameras have certainly stood the test of time.

By adopting this approach to color imaging, photographers have accepted the fact that color cameras are substantially slower and blurrier than grayscale cameras of the same resolution. This is because any camera that is Bayer-like—in the sense of densely sampling all three color channels—is subject to two fundamental limitations. First,

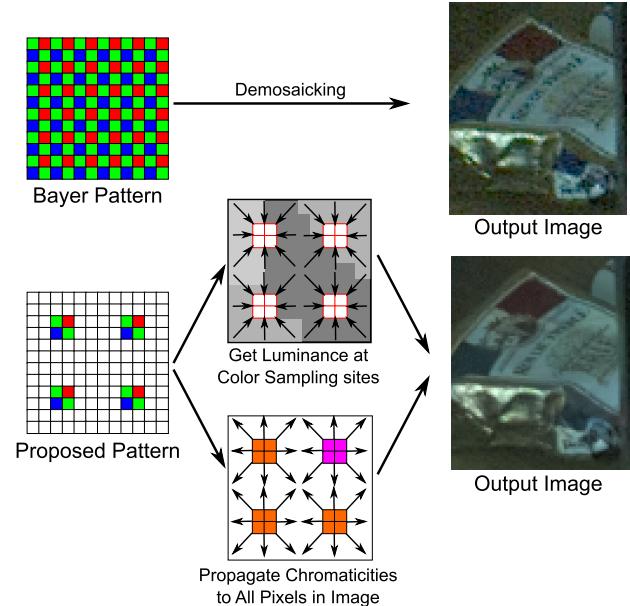


Figure 1. A computational color camera. *Top:* Most digital color cameras use the Bayer pattern (or something like it) to sub-sample color alternately; and then they *demosaick* these samples to create full-color images by up-sampling. *Bottom:* We propose an alternative that samples color very sparsely and is otherwise panchromatic. Full color images are reconstructed by filling missed luminance values and propagating color with guidance from the luminance channel. This approach is much more light-efficient, and it reduces aliasing.

since every color filter blocks roughly two-thirds of the incident light, and since every element of the array is occupied by a color filter, the color camera is roughly three times slower than its grayscale counterpart. Changes to the mosaic pattern, such as including [panchromatic elements in each repeating color block \[6\]](#), can improve efficiency to a certain degree, but there is still substantial loss of light.

The second limitation relates to aliasing and sharpness. When all of the colors are alternated densely, they are all under-sampled, and this induces aliasing artifacts in the re-

construction. Advances in demosaicking algorithms can reduce these artifacts significantly [16], but color cameras still rely on optical blur to avoid them completely, and this makes color cameras much less sharp than grayscale cameras of the same resolution.

We propose an alternative approach to color imaging, one that is well-suited to computational cameras that are freed from the restrictions of traditional digital signal processor-based architectures. As depicted in the bottom of Fig. 1, our filter array leaves the majority of pixels unaltered, and therefore measures luminance without subsampling. Color is sampled only sparsely, by placing copies of the standard 2×2 Bayer block (or something like it) on a relatively coarse grid. This sort of sparse color sampling has been previously considered in the context of recognition and military surveillance, where the goal is to improve luminance resolution at the expense of color fidelity [11]. In contrast, we show here that with an appropriate computational reconstruction process, high color fidelity need not be sacrificed with this arrangement, and the resulting system is in fact preferable to a Bayer-like approach for general color photography.

Our computational reconstruction process is depicted in the bottom of Fig. 1. We use a two-step process in which: 1) missed luminance samples are inferred by hole-filling; and 2) chromaticity is propagated by colorization with a spatio-spectral image model. This approach simultaneously provides luminance images and color images that have fewer aliasing artifacts than Bayer-like approaches, because sequential hole-filling and colorization are more reliable than trivariate up-sampling. At the same time, the filter array is significantly more light-efficient, because most pixels collect measurements without attenuation. The result is a color camera that is both faster and sharper.

2. Related Work

The Bayer pattern was introduced in 1976 [1] and it has persisted almost universally during the four decades since. By including twice as many green samples as red or blue samples, it provides a good balance between sensitivity and color fidelity, and the associated demosaicking process can be implemented using low-power digital signal processors. Subsequent research and development of color imaging with dense alternating mosaics has explored modifications to both filter patterns and demosaicking algorithms, and we discuss each in turn.

Demosacking. Reconstruction algorithms are designed to up-sample color measurements from the sensor’s dense alternating mosaic, and rely on local correlations between color channels [2, 5, 10, 12, 16, 19]. In particular, it is common to define *luminance* as the spectral mean of local color samples—or for convenience, simply as the green

channel—and *chrominance* as the difference between each color sample and its luminance, and then to jointly up-sample by assuming that chrominance channels have lower spatial frequency content than luminance (e.g., [10]).

While working from the assumption of chrominance being relatively “low-pass” may be a good strategy when up-sampling from a dense alternating mosaic, is not entirely accurate from a physical point of view. Empirical analyses of image statistics [4] have shown that chrominance channels have the same energy distribution across spatial frequencies as the luminance channel. Moreover, while analyzing color in terms of luminance and chrominance can provide better decorrelation than linear color space, the luminance and chrominance channels are not statistically independent, because large changes in luminance often coincide with large changes in chrominance.

The natural presence of high-frequency chrominance variation and correlations between luminance and chrominance often leads to images that contain disturbing aliasing artifacts near edges in an image. Modern demosaicking algorithms seek to reduce these artifacts, for example, by explicitly allowing for large chrominance changes across edges [12, 19], or by using more sophisticated image models [2, 5]. However, these methods have all been designed for dense alternating mosaics like the Bayer pattern, and they are therefore subject to the fundamental limitations of light-inefficiency and under-sampling.

Color filter arrays. While the Bayer pattern remains ubiquitous in commercially-available digital color cameras, researchers have explored many other patterns that can make color cameras more light-efficient. Notable examples include interspersing unfiltered pixels among the standard red, green and blue ones [6], and employing broadband filters whose spectral responses still span the RGB color space [13]. These patterns and others like them seem to have been conceived within the context of digital signal processors, where the space of possible patterns is restricted to repetitions of small blocks that each act separately to recover full color. Consequently, they have offered relatively modest improvements to light efficiency.

An interesting exception to this trend is the pattern proposed by Heim *et al.* [11] for visible-plus-infrared cameras used in recognition and military surveillance applications. Like us, their filter array samples color at only a sparse set of locations, allowing the remaining sensor elements to collect high-quality, un-aliased measurements of visible-plus-infrared luminance in low-light conditions. Since high-fidelity color is not required for these applications, Heim *et al.* simply create crude color channels by naive interpolation. In the following sections, we will show that an appropriate computational process can infer high-fidelity color from such sparse color samples, thereby opening the door to greater sharpness and speed in general color photography.

Computational photography. Our reconstruction approach is inspired by the success of computer vision and graphics techniques for in-painting and colorization. In-painting techniques use generic image models and/or exemplar patches from elsewhere in the image to fill in pixels that are missing, for example, because a large region has been manually cut out [7, 17, 20]. Colorization methods use spatio-spectral image models to add color to a grayscale image based on a small number of colored scribbles provided by a user [15]. Our reconstruction process contains these ideas, but implemented in ways that are appropriate for regular patterns and a much finer spatial scale.

3. Computational Color Imaging

Like traditional color cameras, our sensor design and reconstruction algorithms are based on multiplexing color filters spatially, and then reconstructing from these samples by exploiting redundancies in the spatio-spectral content of natural images. A key difference is the model we use for these spatio-spectral redundancies.

We begin by observing that material boundaries occur sparsely in a natural scene, and therefore give rise to images that contain many small, contiguous regions, each corresponding to the same physical material. Within each region, variations in color are primarily due to shading, and this manifests as a spatially-varying scalar factor that multiplies all color channels equally, causing all color three-vectors to be more or less equal up to scale. This implies that the luminance values within each region will vary—as will, in fact, chrominance values—but the chromaticities, or relative ratios between different channels, will stay nearly constant. At the boundaries between regions, both the luminance and chromaticities will change abruptly, but since these boundaries are rare, we expect that a local image patch will feature only a small number of distinct chromaticities.

According to this simple model of the world, every color channel, and any linear combination thereof (including chrominance), can and will exhibit high-frequency spatial variation. They will all be affected by aliasing if subsampled. To avoid this as much as possible, our measurement strategy is as follows (see Fig. 1). We leave a majority of sensor elements unfiltered to measure luminance, and at a coarse grid of locations, we embed 2×2 Bayer blocks that are separated by K pixels in the horizontal and vertical directions ($K = 6$ in the figure).

This design ensures that at least one color channel, the luminance channel, is measured (almost) without subsampling, and can therefore be reconstructed with minimal aliasing. A full color image requires estimating chromaticities as well, and for this we rely on our simple model of the world, according to which a sparse set of color measurements are sufficient for identifying the chromaticity in each small material region. As long as the Bayer blocks are

dense enough to include at least one chromaticity sample per material region, they can be properly propagated under the guidance of the (minimally-aliased) luminance channel.

Intuitively, the reconstruction process requires estimating the missing luminance values at the Bayer block sites, collating the measurements at these sites to compute a chromaticity vector for each block, and then propagating these chromaticities across the image plane. These steps are described in detail next. A reference implementation will be made available on publication.

3.1. Recovering Luminance

The first step in the reconstruction process is recovering the missing luminance values at the sites that sample color. There are a number of in-painting algorithms to consider for this task, but experimentally we find that a simple wavelet-based approach suffices (similar to that of Selesnick *et al.* [17]), since the holes are only small 2×2 blocks.

Let $m[n]$ denote the sensor measurements where $n \in \Omega$ indexes pixel location, and let Ω_L correspond to the subset of locations where luminance is directly measured. We reconstruct the luminance values $l[n]$ everywhere as:

$$l = \arg \min_l \sum_i |(f_i * l)[n]|_{\text{TV}}, \quad (1)$$

such that $l[n] = m[n], \forall n \in \Omega_L$. Here, $*$ denotes convolution, $\{f_i\}_i$ are high-pass wavelet decomposition filters, and $|\cdot|_{\text{TV}}$ corresponds to a total-variations norm on wavelet coefficients in local spatial neighborhoods defined in terms of a kernel k :

$$|\omega[n]|_{\text{TV}} = \sum_{n \in \Omega} [(k * \omega^2)[n]]^{\frac{1}{2}}. \quad (2)$$

The minimization in (1) is implemented using an iterative approach based on variable splitting. At each iteration t , we update the current estimate $l_t[n]$ of $l[n]$ as:

$$l_{t+1}[n] = \begin{cases} m[n], & \text{if } n \in \Omega_L, \\ l_t^*[n], & \text{otherwise,} \end{cases} \quad (3)$$

where,

$$l_t^*[n] = \arg \min_{l^*} \sum_n (l_t[n] - l^*[n])^2 + \beta^t \sum_i |(f_i * l^*)[n]|_{\text{TV}}. \quad (4)$$

This can be interpreted as denoising luminance estimate l^t based on the model (1), with a decreasing value of noise variance defined by β^t , $\beta < 1$. The minimization in (4) can be carried out in closed form by: computing a wavelet decomposition of $l_t[n]$ to obtain coefficients $\omega_{i,t}[n]$; shrinking these coefficients according to

$$\omega_{i,t}^*[n] = \left[\frac{\max(0, [(k * \omega_{i,t}^2)[n]]^{\frac{1}{2}} - \beta^t)}{[(k * \omega_{i,t}^2)[n]]^{\frac{1}{2}}} \right] \omega_{i,t}[n]; \quad (5)$$

and then reconstructing l_t^* from the coefficients $\omega_{i,t}^*[n]$.

In our implementation, we use a single-level, undecimated Daubechies-2 wavelet decomposition and a 3×3 box filter as the neighborhood kernel k , and execute fifty iterations with $\beta = 2^{-1/4}$, initialized with $l_0[n] = m[n]$. As we shall see in Sec. 4, this process yields a high-quality reconstruction of scene luminance.

3.2. Estimating Chromaticities at Bayer Blocks

Our next step is to estimate the red, green, and blue chromaticities at each of the four sites within every 2×2 Bayer block, where chromaticity is defined as the ratio between a color channel value and the luminance value at the same pixel. We use notation $r[n], g[n], b[n]$ for the (yet unknown) red, green, and blue (RGB) color channel values at each pixel,¹ and $c_r[n], c_g[n], c_b[n]$ for the associated chromaticities (e.g., $c_r[n] = r[n]/l[n]$).

We reconstruct the chromaticities within each Bayer block independently and, at first, we assume that the Bayer block does not span a material boundary. This means that our task is to estimate a single chromaticity vector that is shared by all four sites within a block. Let $n \in \{n_r, n_b, n_{g_1}, n_{g_2}\}$ be the sites corresponding to the red, blue, and two green filters within a Bayer block. According to our definition of chromaticity we have

$$\begin{aligned} m[n_r] &= c_r l[n_r], \quad m[n_b] = c_b l[n_b], \\ m[n_{g_1}] &= c_g l[n_{g_1}], \quad m[n_{g_2}] = c_g l[n_{g_2}], \end{aligned} \quad (6)$$

where c_r, c_g, c_b are the desired RGB chromaticities of the material at that block. We also assume that the color and luminance channels are related as:

$$\gamma_r r[n] + \gamma_g g[n] + \gamma_b b[n] = l[n], \quad (7)$$

where $\gamma_r, \gamma_g, \gamma_b$ are determined by calibrating the spectral filter and sensor responses.

Accordingly, we compute a regularized least-squares estimate for the chromaticities as:

$$c_i = \arg \min_{c_i} (l[n_i]c_i - m[n_i])^2 + \sigma_z^2(c_i - \bar{\gamma})^2, \quad (8)$$

for $i \in \{r, b\}$, and

$$\begin{aligned} c_g &= \arg \min_{c_g} (l[n_{g_1}]c_g - m[n_{g_1}])^2 + (l[n_{g_2}]c_g - m[n_{g_2}])^2 \\ &\quad + \sigma_z^2(c_g - \bar{\gamma})^2, \end{aligned} \quad (9)$$

subject to $\sum_i \gamma_i c_i = 1$ as per (7). Here, σ_z^2 is the observation noise variance, and $\bar{\gamma} = (\sum_i \gamma_i)^{-1}$. This regularization

¹Without loss of generality, we use terms red, green, and blue to refer to the colors being measured by the spectral filters within the Bayer block. Since there are fabrication constraints on spectral filters, they may differ from the matching functions of standard RGB, and a 3×3 linear transform will be required to correct for this difference. Such transforms should be applied to the output of our reconstruction algorithm.

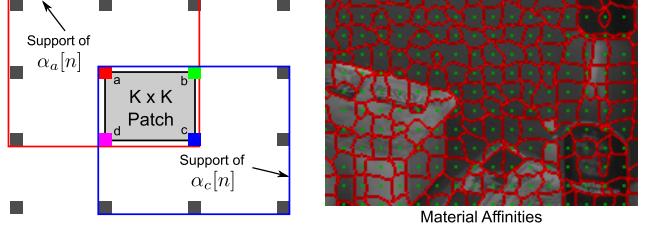


Figure 2. Propagating chromaticity with material affinities. *Left*: Chromaticities at pixels within each $K \times K$ patch are computed as convex combinations of chromaticities measured by the Bayer blocks at its corners. The combination weights are determined by four affinity maps $\alpha_j[n]$, one from each corner Bayer block j , that encode luminance edge information. *Right*: Affinity map showing regions of pixels with highest affinity to each block (marked in green), super-imposed on the corresponding luminance image.

can be interpreted as biasing the chromaticity estimates at dark pixels toward “gray”, or $c_r = c_g = c_b$.

This is an equality-constrained least-squares problem and can be solved in closed form to yield a chromaticity vector at each Bayer block. We then apply a post-processing step to remove isolated mis-estimates, which typically occur in dark regions or where significant luminance variation exists within a Bayer block. For this we use median filtering, which is common in removing color-interpolation outliers (e.g., [8, 12]). Since the Bayer blocks are arranged on a coarse grid of points, we interpret the spatial collection of estimated chromaticities as a low-resolution chromaticity image that is about K^2 times smaller than the sensor resolution. We apply 5-tap, one-dimensional median-filters on this coarse image at multiple orientations to generate multiple proposals for the filtered result at each site, and then we choose from these proposals the value that is most similar to the pre-filtered value at that site. This process has the benefit of eliminating isolated errors while preserving thin structures, and we apply it multiple times till convergence.

3.3. Propagating Chromaticities

The final step in the reconstruction process involves propagating the chromaticity information from the Bayer block locations to all sites, based on features in the luminance channel that are likely to correspond to material boundaries. We do this in two steps. First, we obtain an initial estimate of the chromaticity at each site by computing a convex combination of the four chromaticities at its neighboring Bayer blocks, with the convex weights determined by local luminance features. This step is efficient and highly parallelized. In the second step, we refine these initial estimates through non-local filtering.

To begin, we partition the image into $K \times K$ patches such that the four corners of each patch include one site from a Bayer block (see Fig. 2). Let a, b, c, d index the

four corners, with n_a, n_b, n_c, n_d being the corresponding corner pixel locations, and $c_{i,a}, \dots, c_{i,d}$, $i \in \{r, g, b\}$ the chromaticities as recovered in Sec. 3.2. The chromaticities within the patch are then computed as:

$$c_i[n] = \kappa_a[n]c_{i,a} + \kappa_b[n]c_{i,b} + \kappa_c[n]c_{i,c} + \kappa_d[n]c_{i,d}, \quad (10)$$

where $\kappa_a[n], \dots, \kappa_d[n]$ are scalar combination weights.

To compute these weights, we introduce the intermediate concept of **material affinity** affinity maps that encode the affinity $\alpha_j[n] \in [0, 1]$ between the Bayer block j and the sites n that surround it. The maps are computed independently (and in parallel) within overlapping $(2K+1) \times (2K+1)$ regions centered at the Bayer blocks (see Fig. 2) as:

$$\alpha_j[n] = \arg \min_{\alpha} \sum_n \sum_{n' \in \mathbb{N}_1(n)} c[n, n'](\alpha[n] - \alpha[n'])^2, \quad (11)$$

typo?

idea, if luminance ~same, then alpha should not change by much,
since if they do differ by a lot, would be penalized more severely by the coefficients

with the constraint that $\alpha_j[n] = 1$ at the four sites in the Bayer block j , and 0 at the sites in the remaining eight Bayer blocks in the region, and $\mathbb{N}_1(n)$ corresponds to a 3×3 neighborhood around site n . The scalar weights $c[n, n']$ come from edges in the luminance image. For example, when n, n' are horizontal neighbors, we define $c[n, n']$ as:

$$c[n, n'] = \exp(-\max(e_h[n], e_h[n'])) , \quad (12)$$

small, if exists an edge over luminance space
large, if luminance uniform ?

where

$$e_h[n] = \frac{\sum_s |(G_s^h * l)[n]|^2}{\sum_{n^\dagger} \sum_s |(G_s^h * l)[n^\dagger]|^2}, \quad (13)$$

is the normalized sum of horizontal gradient magnitudes of $l[n]$ —computed across multiple scales s using horizontal derivative of Gaussian filters G_s^h . The summation on n^\dagger in the denominator is over the $K \times K$ patch that contains n .

The minimization in (11) can be carried out efficiently using the conjugate-gradient method. Moreover, due to the symmetry of the cost function and the constraints, we need only solve (11) for three-quarters of the sub-regions j . This is because each site n has four maps α_j associated with it, one for each of the four nearest Bayer blocks. The values of these maps must sum to one at every site n , so by indexing the $(2K+1) \times (2K+1)$ sub-regions appropriately, we need only compute three maps out of every four. Figure 2 illustrates an affinity map computed using this approach.

Having computed the affinity maps $\alpha_j[n]$, we set the combination weights in (10) as:

$$\kappa_j[n] \propto \alpha_j^2[n] l[n_j], \quad j \in \{a, b, c, d\}, \quad (14)$$

with $\kappa_a[n] + \dots + \kappa_d[n] = 1$. Note that in addition to the affinity maps, the weights are also proportional to $l[n]$ to promote contributions from brighter blocks when the affinities are roughly equal, as is the case in homogeneous regions without edges.

This process gives us an initial estimate of chromaticities at each pixel. We then apply a non-local kernel to further refine these chromaticity estimates as:

$$c_i^+[n] = \sum_{n' \in \mathbb{N}_2(n)} w[n, n'] c_i[n], \quad i \in \{r, g, b\}, \quad (15)$$

where $w[n, n']$ are positive scalar weights computed by matching luminance patches centered at n and n' :

$$w[n, n'] = \frac{1}{Z[n]} \exp \left(-\frac{\|G * (l[n'] - l[n])^2\|^2}{2h^2} \right) l[n'],$$

$$Z[n] = \sum_{n' \in \mathbb{N}_2(n)} \exp \left(-\frac{\|G * (l[n'] - l[n])^2\|^2}{2h^2} \right) l[n'], \quad (16)$$

where G is a Gaussian filter with standard deviation $K/8$, and $h = \lambda \sigma_z$ is set proportional to the standard deviation of observation noise (with $\lambda = 40$ in our implementation).

An important decision here is choosing the neighborhood \mathbb{N}_2 within which to look for matching patches in (15). A larger neighborhood has the obvious benefit of averaging over a broader set of candidate chromaticities. However in addition to an increase in computational cost, larger neighborhoods also make it more likely that candidate patches will be mis-classified as a match. This is because our approach differs from standard non-local means [3] in one key aspect—since our goal is to estimate chromaticities and not denoising, we use the luminance channel for computing the kernel weights $w[n, n']$, but use these weights to update the chromaticities.

As a compromise, we restrict $\mathbb{N}_2(n)$ to a $(K+1) \times (K+1)$ window centered at n , but apply the update in (15) multiple times. Therefore while we effectively average chromaticities across a larger region, at each update, we only borrow estimates from pixels in a restricted neighborhood defined by the color sampling rate. This approach is also computationally efficient, since the updates do not affect the luminance channel and the weights $w[n, n']$ need only be computed once.

The final output RGB image is then simply given by the product of the chromaticity and luminance estimates:

$$r[n] = c_r[n] l[n], \quad g[n] = c_g[n] l[n], \quad b[n] = c_b[n] l[n]. \quad (17)$$

4. Experimental Results

We evaluate our approach by simulating sensor measurements using ten captured images of indoor and outdoor scenes from the database of Gehler et al. [9]. We use the linear version of this database generated by Shi and Funt [18],

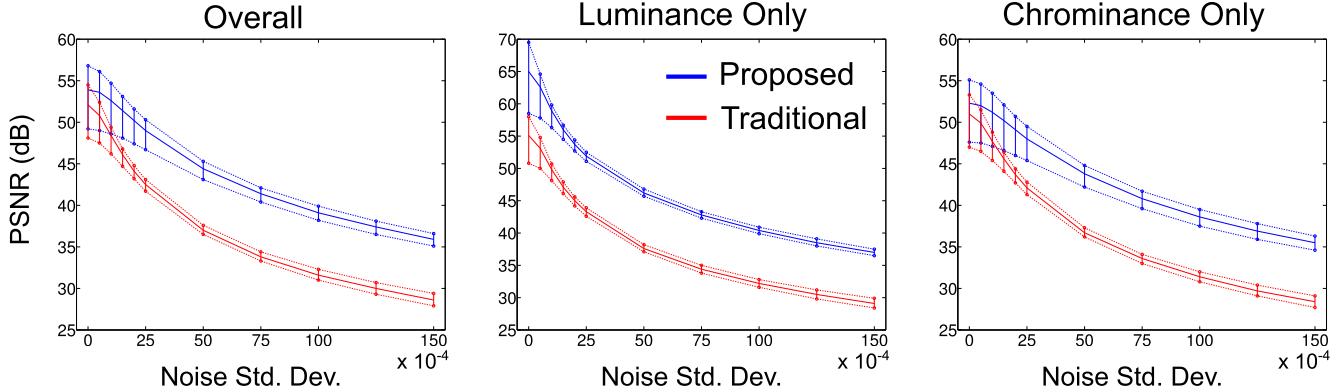


Figure 3. Quantitative performance of proposed method over traditional Bayer-based demosaicking. Shown here are quantiles (25%, median, 75%) of PSNR computed over all overlapping 10×10 patches in our ten test images, for different levels of noise, and for the entire reconstructed RGB image as well as for the luminance channel and chrominance channels separately. We see that our method outperforms traditional Bayer demosaicking even in the noiseless case, but the degree of improvement increases dramatically as we start considering higher noise levels (corresponding to low-light capture). We also see that at relatively low noise levels, most of the improvement comes from improvements in the luminance channel.

who replaced every 2×2 Bayer block in the original camera measurements with a single trichromatic pixel. Since there was no up-sampling during their replacement, these are full-color, low-noise, alias-free images at half the resolution (2.7 megapixels) of the original camera measurements; and they are representative of the spatio-spectral statistics that digital cameras are likely to encounter.

We treat these full-color images as ground truth, and we evaluate our approach by sampling these images according to our filter pattern, simulating noise, and then reconstructing an approximation to the ground truth as described in the previous section. Performance is measured by the difference between the ground truth full-color image and the approximated reconstruction. As a baseline for comparison, we repeat the protocol using a standard Bayer pattern and a state-of-the-art demosaicking algorithm [19]. For the elements of our pattern that are panchromatic, we simulate samples simply by summing the three color color values at the corresponding pixel in the ground-truth image, *i.e.*, $l[n] = r[n] + g[n] + b[n]$. This corresponds to measurements from a hypothetical spectral filter that is the sum of the camera’s native RGB filters, and is a conservative estimate of the light-efficiency of unfiltered measurements taken by the underlying sensor [14].

The choice of the color-sampling frequency K in our pattern involves a trade-off between light-efficiency and sharpness on one hand, and the ability to detect fine variations in chromaticity (material boundaries) on the other. After trying different values on test images at various scene scales, we set $K = 6$ because it provides a reasonable balance between efficiency, sharpness, and accurate chromaticity variation.

We generate measurements with different levels of ad-

ditive Gaussian noise, where higher values of noise variance simulate low-SNR measurements corresponding to low-light capture. Figure 3 summarizes the performance of our approach and the baseline Bayer approach across noise levels in terms of PSNR—defined as $10 \log_{10}(1/\text{MSE})$ for intensities in the range $[0, 1]$, where MSE is the mean square error between the ground truth and reconstructed intensities. We show quantiles of this metric computed over all overlapping 10×10 patches from all ten images (avoiding patches within 15 pixels of the image boundary), which is about 2.7×10^7 patches in total. We plot PSNR values for errors summed over all color channels, as well as in luminance and chrominance channels separately.

We see that even in the noiseless case, when light-efficiency is not a factor, our computational approach yields more accurate reconstructions. For this case, the performance gain is primarily in the luminance channel, which is measured directly by our sensor and reconstructed without aliasing. With increasing amounts of noise (*i.e.*, lower light levels), the difference in the two approaches becomes more significant, and we see that our design offers substantially improved estimates of both chrominance and luminance. The improvements in chrominance are due to the fact that our sparse and noisy (*i.e.*, Bayer-block attenuated) measurements of chromaticity are multiplied by high-SNR, un-aliased luminance values during propagation, which is better than everywhere collecting color measurements that are aliased and higher in noise.

Figure 4 contains examples of reconstructed image regions by both methods for different noise levels (please see the supplementary material for more reconstruction results). The benefits at low light levels are apparent, and we readily find that our design produces reasonable reconstructions at



Figure 4. Sample reconstructions. Shown here are regions (of various sizes) cropped from images reconstructed from traditional Bayer-based demosaicking and our approach, at different noise levels—increasing from left to right, with the first pair corresponding to the noiseless case. The results from our algorithm usually have fewer aliasing artifacts in the noiseless case, and are reasonable at noise levels where traditional demosaicking outputs are significantly degraded. Please zoom in for a detailed view of reconstruction quality.

higher noise-levels where traditional demosaicked images would become dominated by noise. With low-noise too, our approach generally offers better reconstructions with fewer aliasing artifacts and high color-fidelity. However, it is worth noting that we occasionally incorrectly estimate the hue of very fine image structures, when they fall within the gaps of our color sampling sites (*e.g.*, fourth row in Fig. 4). Therefore, our design differs from Bayer-based demosaicking in the way it reacts to high-frequency spatial variation in color. While the latter is susceptible to aliasing at all boundaries, our design is only affected when changes in chromaticity happen at a rate than the color sampling frequency K . And even in these regions, we recover edges and textures accurately but fail to detect changes in hue.

A reference implementation of our reconstruction process is implemented in MATLAB and C. It can be distributed across up to three processor cores, and requires roughly one hundred seconds to reconstruct each 2.7 megapixel test image. It will be made publicly available upon publication.

5. Discussion

Material boundaries in natural scenes are relatively rare, so sparse samples of color can be sufficient for color photography. This approach makes sense in the context of computational cameras, where reconstruction can leverage general computational resources, either on-board or in the cloud. It is preferable to traditional Bayer-like designs in this context, because it enables faster capture with fewer aliasing artifacts.

Our reconstruction process is designed to demonstrate that high-quality reconstruction is possible from sparse color measurements, but it provides only one example of such a process, and it should be interpreted as just one point in the space of trade-offs between computational complexity and reconstruction quality. Based on the computational hardware available and intended application, one could choose to replace the different steps of our algorithm with counterparts that are either more or less expensive computationally. For example, one could use a more sophisticated dictionary-based algorithm for luminance inpainting, or omit the non-local filter-based refinement step for chromaticity propagation. Indeed, it may be desirable to have a cheaper on-camera reconstruction algorithm to generate image previews, while retaining the original sensor measurements for more sophisticated offline processing.

Our use of Bayer blocks at the sparse color sampling sites is also just one of many possible choices. Previous work on designing broadband, light-efficient spectral filters [13] is complimentary to the proposed approach, and our design's light-efficiency can likely be improved further by using such filters at the color sampling sites. Another direction for future work is to consider hyperspectral imag-

ing with similarly sparse spatial multiplexing. The freedom to sample color sparsely suggests that we may be able to make additional spectrally-independent measurements at each sampling site, without incurring the loss in spatial resolution associated with traditional alternating mosaics.

References

- [1] B. E. Bayer. Color imaging array. US Patent 3,971,065, 1976.
- [2] E. P. Bennett, M. Uyttendaele, C. L. Zitnick, R. Szeliski, and S. B. Kang. Video and image bayesian demosaicing with a two color image prior. In *Proc. ECCV*. 2006.
- [3] A. Buades, B. Coll, and J.-M. Morel. A non-local algorithm for image denoising. In *Proc. CVPR*. IEEE, 2005.
- [4] A. Chakrabarti and T. Zickler. Statistics of real-world hyperspectral images. In *Proc. CVPR*, 2011.
- [5] K.-H. Chung and Y.-H. Chan. Color demosaicing using variance of color differences. *IEEE Trans. IP*, 2006.
- [6] J. Compton and J. Hamilton. Image sensor with improved light sensitivity. US Patent 8,330,839, 2012.
- [7] A. Criminisi, P. Perez, and K. Toyama. Object removal by exemplar-based inpainting. In *Proc. CVPR*, 2003.
- [8] W. Freeman. Median filter for reconstructing missing color samples. US Patent 4,724,395, 1988.
- [9] P. V. Gehler, C. Rother, A. Blake, T. Minka, and T. Sharp. Bayesian color constancy revisited. In *Proc. CVPR*, 2008.
- [10] B. K. Gunturk, Y. Altunbasak, and R. M. Mersereau. Color plane interpolation using alternating projections. *IEEE Trans. IP*, 2002.
- [11] G. B. Heim, J. Burkepile, and W. W. Frame. Low-light-level EMCCD color camera. In *Proc. SPIE, Airborne Intelligence, Surveillance, Reconnaissance (ISR) Systems and Applications III*, 2006.
- [12] K. Hirakawa and T. W. Parks. Adaptive homogeneity-directed demosaicing algorithm. *IEEE Trans. IP*, 2005.
- [13] K. Hirakawa and P. Wolfe. *IEEE Trans. IP*, 2008.
- [14] J. Jiang, D. Liu, J. Gu, and S. Ssstrunk. What is the Space of Spectral Sensitivity Functions for Digital Color Cameras? In *Proc. Workshop on the Applications of Computer Vision (WACV)*, 2013.
- [15] A. Levin, D. Lischinski, and Y. Weiss. Colorization using optimization. *ACM SIGGRAPH*, 2004.
- [16] X. Li, B. Gunturk, and L. Zhang. Image demosaicing: A systematic survey. In *Proc. SPIE*, 2008.
- [17] I. W. Selesnick, R. Van Slyke, and O. G. Guleryuz. Pixel recovery via ℓ_1 minimization in the wavelet domain. In *Proc. ICIP*, 2004.
- [18] L. Shi and B. Funt. Re-processed version of the Gehler color constancy dataset of 568 images. 2010. Accessed from <http://www.cs.sfu.ca/~colour/data/>.
- [19] L. Zhang and X. Wu. Color demosaicking via directional linear minimum mean square-error estimation. *IEEE Trans. IP*, 2005.
- [20] M. Zhou, H. Chen, L. Ren, G. Sapiro, L. Carin, and J. W. Paisley. Non-parametric bayesian dictionary learning for sparse image representations. In *NIPS*, 2009.