Video and Image Bayesian Demosaicing with a Two Color Image Prior

Eric P. Bennett[1], Matthew Uyttendaele[2], C. Lawrence Zitnick[2], Richard Szeliski[2], and Sing Bing Kang[2]

[1] The University of North Carolina at Chapel Hill, Chapel Hill, NC*
[2] Microsoft Research, Redmond, WA

Abstract. The demosaicing process converts single-CCD color representations of one color channel per pixel into full per-pixel RGB. We introduce a Bayesian technique for demosaicing Bayer color filter array patterns that is based on a statistically-obtained two color per-pixel image prior. By modeling all local color behavior as a linear combination of two fully specified RGB triples, we avoid color fringing artifacts while preserving sharp edges. Our grid-less, floating-point pixel location architecture can process both single images and multiple images from video within the same framework, with multiple images providing denser color samples and therefore better color reproduction with reduced aliasing. An initial clustering is performed to determine the underlying local two color model surrounding each pixel. Using a product of Gaussians statistical model, the underlying linear blending ratio of the two representative colors at each pixel is estimated, while simultaneously providing noise reduction. Finally, we show that by sampling the image model at a finer resolution than the source images during reconstruction, our continuous demosaicing technique can super-resolve in a single step.

1 Introduction

Most digital cameras use a single sensor to record images and video. They use color filter arrays (CFAs) to capture one color band per pixel, and interpolate colors to produce full RGB per pixel. This interpolation process is known as demosaicing.

The Bayer filter is the most popular type of CFA used today. Demosaicing a raw Bayer image requires an underlying image model to guide decisions for reconstructing the missing color channels. At every pixel only one color channel is sampled, so we must use that information, combined with that of nearby samples, to reconstruct plausible RGB triples. An image model provides a prior for reconstructing the missing colors based on patterns of the surrounding samples. Demosaicing algorithms differ in how local spatial changes in a single color channel are used to propagate information to the other channels.

Demosaicing is inherently underspecified because there are no complete RGB triples anywhere in the image to learn an image specific prior from. Even worse,

* Research performed while an intern at Microsoft Research, Redmond.

A. Leonardis, H. Bischof, and A. Pinz (Eds.): ECCV 2006, Part I, LNCS 3951, pp. 508–521, 2006.

if every pixel in an image has a random ratio of red, green, and blue, there is no hope of reconstructing the image. Only by assuming some local coherence between channels can we reasonably reconstruct the original image.

In this paper, we reduce the problem's complexity by developing an underlying statistical image model that treats all colors in a local area as a linear combination of no more than two representative colors. To use this model, we estimate the two representative colors for the local area centered at each pixel and find the linear blending for each pixel that determines its color.

Our two color model is motivated by the need to reconstruct accurate colors at edges. Current demosaicing algorithms can accurately reconstruct colors in image areas where only low frequencies are present. However, yellow or purple color fringing can appear at high frequency edges due to the edges in multiple color channels not being aligned in the reconstruction. By constraining the system to interpolate between fully specified RGB colors, there is less risk of misalignment. This constraint also provides noise reduction in smooth areas.

The underlying two colors at each pixel are estimated using K-Means clustering. The RGB colors used for clustering can come from any existing demosaicing algorithm. The final color at each pixel results from discovering the proper linear blending coefficient between the two representative colors.

Based on knowledge of a small set of CFA samples around each pixel, our problem is posed using Bayesian probabilities. Stating the problem statistically allows the model to include non-grid-aligned samples from multiple images or temporally adjacent video frames to increase color accuracy. Also, by sampling the demosaicer's output at an increased resolution, information from these additional samples exposes details between pixels, providing super-resolution in a single step.

2 Previous Work

There are many approaches to demosaicing. A simple technique for demosaicing a Bayer color filter array [1] (shown in Figure 1) is bilinear interpolation, which is able to reconstruct smooth and smoothly varying image areas. At the edges, bilinear interpolation risks creating aliasing or "zippering" artifacts where every other pixel along an edge alternates between being considered on or off the edge. Color fringing is the other significant artifact, where yellows, purples, and cyans appear along or on sharp edges. These artifacts result from bilinear interpolation incorrectly placing an edge in a color channel one pixel offset from the same edge in a different channel.

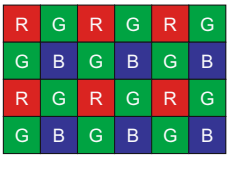


Fig. 1. The Bayer color filter array pattern

Solving the color fringing and zippering issues was the focus of much subsequent research. One approach is to bilinearly interpolate the green channel and then interpolate the red:green and blue:green ratios for the remaining samples [2]. This assumes there exists no green detail smaller than two pixels and that red and green locally vary in a fixed ratio with green. Median interpolation [3] assumes that bilinear interpolation can be repaired by median filtering the red-green and blue-green spaces. Both of these methods target fringing artifacts but can result in over-smoothing. Comparisons can be found in [4].

The approach for Vector Color Filter Array Demosaicing [5] uses filtering in a different way. It first generates pseudo-colors using the local combinations of R, G, and B CFA samples. The chosen color is the "median color" whose total distance to the other pseudo-colors is minimized.

Techniques sensitive to gradients were introduced to reduce over-smoothing by performing color interpolation only along sharp edges and not across them. Laroche and Prescott [6], Hamilton and Adams [7], and Chang *et al.* [8] presented algorithms with a chronologically increasing number of gradient directions evaluated and interpolated. Kimmel [9] modeled images as smooth surfaces separated by edge discontinuities which were enhanced with inverse diffusion.

There are also grid-based techniques that learn statistical image models. Malvar *et al.* [10] presents a fast linear interpolation scheme with color-specific kernels which were learned using a Wiener approach from the popular Kodak data set [11]. Another approach, independent of color filter array pattern, is Assorted Pixels [12] which constructs a kernel based on any multi-spectral array.

Hel-Or [13] modeled correlation between channels using Canonical Correlation Analysis. Bayesian statistical methods were applied to the demosaicing problem by Brainard [14] who modeled the base image priors as a set of sinusoids. Closer to our two color approach is Bayesian Matting [15], which finds the foreground and background colors from the surrounding areas. Using a Bayesian system, it finds a linear blending ratio between these two colors.

To perform super-resolution enhancement, additional resolution information must be acquired from somewhere. Zomet and Peleg [16] use information in multiple images taken from different sensors while Freeman *et al.* [17] inferred resolution from different resolution scalings of the same image. Demosaicing while providing super-resolution from multiple images was investigated by Fung and Mann [18]. Their approach places samples into a regularly spaced grid and each output pixel component is found using a nearest neighbor search of the registered inputs. Gotoh and Okutomi [19] generalized earlier super-resolution approaches to directly process Bayer samples from many frames. Their results used primarily synthetic frames ($>$20), and did not give quantitative results or consider single-image demosaicing.

To evaluate our results, we are interested in using a more perceptually valid measure than merely SNR or MSE. The S-CIELAB [20, 21] model provides an extension to the $L\alpha\beta$ color space that is aware of local contrast and can hint if the human visual system (HVS) cannot detect errors due to masking.

Also of interest is the iCAM [22] model which predicts the color the HVS perceives in the presence of nearby colors. Although we do not use such a complex model, there are methods that measure perceptual error considering more aspects of the HVS, such as the Multiscale Adaptation Model [23] and Visible Differences Predictor [24]. These and other models have proved useful in detecting HDR compression error and in allocating rendering tasks based on contrast [25, 26].

3 The Two Color Model

Central to our processing is the assumption that at most two representative colors exist within a local neighborhood. Every pixel within that neighborhood is either one of the representative colors or is a linear combination of both. This assumption is violated in areas where more than two different colors meet, but such occurrences are relatively rare. Assuming a Gaussian noise model, this distribution represents a cylindrical volume in color space which spans the two representative colors, as shown in Figure 2.

The two color model serves multiple purposes. Primarily, it serves as a constraint to the ill-conditioned demosaicing problem. With only a single channel reported from the sensor at each pixel, we rely on local color combinations from nearby samples to create RGB triples. To use information from neighboring pixels, an assumption of local coherence is made, as in Section 2. Our process, clustering Bayer samples into RGB triples, is discussed in Section 3.1.

Secondly, by snapping values to a consistent local, edge-preserving model, the amount of local variation is decreased, and therefore noise reduction is provided at no additional computational cost. Furthermore, model outliers can be readily identified and appropriately attenuated or preserved.

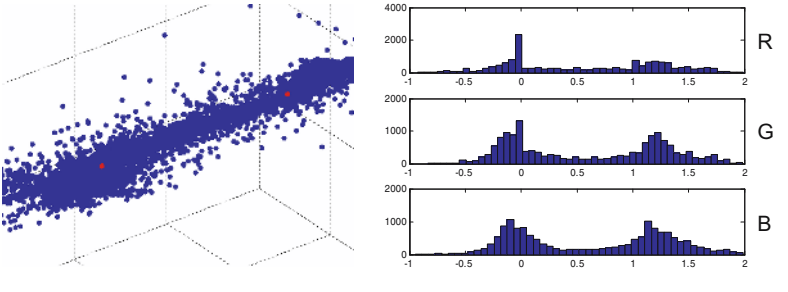


Fig. 2. Visualization of the two color model in a local neighborhood. For 100 randomly chosen pixels near edges in our calibration chart data set (shown in Table 1), the two representative colors were found and the 25 neighboring pixels of each were compared to the representative colors. On the left, a 3D color space plot of agreement with the two color model, as most nearby samples fall at or between the representative colors, shown in red. On the right, a histogram of the relative frequency for each channel, showing that the majority of pixels cluster near the representative colors (at 0 and 1).

Finally, previous approaches assume that when a single color channel changes, a similar change is likely reflected in all channels. This assumption can create sharp edges with color fringing. The yellow or purple edges of fringing result when a demosaicer makes an abrupt change in the value of one channel a single pixel before making a similar change in another channel. This is because edge boundaries cannot be reliably detected in a single color channel due to the sparse sampling of the Bayer grid. Because all three, just two, or only one channel may change, assuming simultaneous changes in all channels may create visible artifacts. However, once both representative colors are determined (i.e. the colors on each side of an edge), we can determine which channels actually vary.

3.1 Two Color Clustering

To discover the underlying two color model at each pixel, the neighborhood surrounding each pixel must be clustered into those two colors. The Bayer image provides only a single channel sample, leaving two unknown RGB triples and an unknown blending coefficient to specify the model.

To reduce the problem complexity, a preliminary demosaicing pass assigns each Bayer sample a fully specified RGB triple using any desired pre-existing demosaicer. Then, the demosaiced colors in the surrounding neighborhood of each pixel can be clustered. Using the K-Means algorithm evaluated in RGB space, two clusters can be computed, with their means being the representative colors. We use a weighted K-Means in which the weight is the inverse Euclidean distance from each sample to the center of the kernel. Note that cluster sizes are not balanced, so a single pixel detail in an otherwise smooth area can be preserved. Also, clustering can be performed in other color spaces, such as L$\alpha\beta$ or XYZ, but clustering in these spaces made little difference in regards to accuracy.

The neighborhood size of samples to cluster is a function of how large color details appear in the source image. We found a two pixel radius around the kernel's center to work well in all of our test cases. This supplies a sufficient number of samples from each of the Bayer color channels.

Because it is possible that more than two colors may exist in a local image area or that significant noise may be present, an outlier rejection stage is included. Using the mean and variance of the clusters, samples that lie outside of a single standard deviation of their closest cluster mean are rejected. K-Means is then repeated to obtain cleaner cluster means. This provides for better reproduction where color values change rapidly away from the kernel center.

The major factor in the quality of clustering is the choice of algorithm used for the "bootstrapping" demosaicing to make clustering tractable. Although any demosaicer could be used, there are qualities that improve performance. The first is preservation or accentuation of high frequency features. Algorithms such as bilinear interpolation and median interpolation have a tendency to low-pass filter, which should be avoided. Alternately, algorithms that preserve high frequencies are prone to generate edge fringing and aliasing. We obtained the best results using the Malvar et al.'s [10] demosaicer, which preserves high frequencies while not generating too many fringing artifacts.

4 Two Color Demosaicing

The two color model provides two RGB priors, $\bar{J}$ and $\bar{K}$, for each pixel x in the image. The color C of pixel x is assumed to be a linear combination these two colors, i.e.,

$$C = (1 - \alpha)\bar{J} + \alpha\bar{K}. \tag{1}$$

Within a neighborhood of our pixel x, our Bayer sensor gives us a set of samples $s_i \in S$. The index of the RGB color channel specified by sample s_i is denoted by t_i. If $\bar{J}_{t_i}$ specifies the t_ith color channel for color $\bar{J}$, and similarly for $\bar{K}$ and C, then we could compute the unknown value of α directly from the Bayer sample s_x at location x (the central pixel) as:

$$\alpha = \frac{s_x - \bar{J}_{t_x}}{\bar{K}_{t_x} - \bar{J}_{t_x}}. \tag{2}$$

However, if the difference between $\bar{J}_{t_x}$ and $\bar{K}_{t_x}$ is small, our estimate of α will be inaccurate due to discretization and image noise.

We compute a more robust estimate of α using the entire set of samples S. That is, we want to find the most likely value $\hat{\alpha}$ of α given our sample set S and color priors $\bar{J}$ and $\bar{K}$:

$$\hat{\alpha} = \arg\max_{\alpha} P(\alpha|S, \bar{J}, \bar{K}). \tag{3}$$

Using Bayes' theorem, and assuming $\bar{J}$ and $\bar{K}$ are independent of S and α, we can rearrange (3) to yield

$$P(\alpha|S, \bar{J}, \bar{K}) = \frac{P(S|\alpha, \bar{J}, \bar{K})P(\alpha)}{P(S)}. \tag{4}$$

Assuming all s_i are independent and $P(S)$ is a uniform distribution, we find

$$P(\alpha|S, \bar{J}, \bar{K}) \propto P(\alpha) \prod_i P(s_i|\alpha, \bar{J}, \bar{K}). \tag{5}$$

$\bar{J}$, $\bar{K}$, and α specify a predicted color $C^* = (1 - \alpha)\bar{J} + \alpha\bar{K}$ for pixel x. Assuming an independent identical distribution (i.i.d.) for neighboring color noise, the relationship between $C^*_{t_i}$ and s_i can be modeled using a normal distribution:

$$P(s_i|\alpha, \bar{J}, \bar{K}) \propto exp\left(-\frac{(s_i - C^*_{t_i})^2}{2\sigma_i^2}\right). \tag{6}$$

The distribution between neighboring pixels has been shown to be highly kurtotic [27], but for computational efficiency we assume a Gaussian distribution. The variance σ_i^2 is dependent on two factors: the global per-channel image noise σ_N and the pixel distance between x and s_i. We assume pixel colors are locally similar, and less similar farther away. Thus, the variance between s_i and C_{t_i} increases as their distance in image space increases. We compute the variance σ_i as

$$\sigma_i = \sigma_N(1 + \lambda\Delta_d), \tag{7}$$

where Δ_d is the pixel distance between x and s_i, and λ is empirically set to 6.

Since the value of s_i is known and we want to compute the value of α that maximizes equation (6), we find it useful to rearrange it as follows:

$$exp\left(-\frac{(s_i-((1-\alpha)\bar{J}_{t_i}+\alpha\bar{K}_{t_i}))^2}{2\sigma_i^2}\right) = exp\left(-\frac{\left(\alpha-\frac{s_i-\bar{J}_{t_i}}{\bar{K}_{t_i}-\bar{J}_{t_i}}\right)^2}{2\left(\frac{\sigma_i}{\bar{K}_{t_i}-\bar{J}_{t_i}}\right)^2}\right). \quad (8)$$

Equation (8) is a Gaussian over α with mean α_i and variance $\sigma_{\alpha_i}^2$:

$$\alpha_i = \frac{s_i-\bar{J}_{t_i}}{\bar{K}_{t_i}-\bar{J}_{t_i}} \text{ and } \sigma_{\alpha_i}^2 = \left(\frac{\sigma_i}{\bar{K}_{t_i}-\bar{J}_{t_i}}\right)^2. \quad (9)$$

We can then combine equations (5) and (8) to yield

$$P(s_i|\alpha,\bar{J},\bar{K}) \propto \prod_i exp\left(-\frac{(\alpha-\alpha_i)^2}{2\sigma_{\alpha_i}^2}\right). \quad (10)$$

The optimal value of α for $P(s_i|\alpha,\bar{J},\bar{K})$ is

$$\alpha^* = \frac{\sum_i(\sigma_{\alpha_i}^{-2}\alpha_i)}{\sum_i \sigma_{\alpha_i}^{-2}} \quad (11)$$

In practice, we ignore the contribution of color components where the absolute difference between $\bar{J}_{t_i}$ and $\bar{K}_{t_i}$ is less than 2.0. When all color components are this close, only one color is present. Thus, α is set to perform a simple average of the cluster means.

Finally, to find our value of $\hat{\alpha}$ in (3), we need to define a prior over α, $P(\alpha)$. Given that most pixels within an image only get contribution from a single color, we bias α to have a value of 0 or 1:

$$P(\alpha) = \begin{cases} 1 & : \quad \alpha \in \{0,1\} \\ \eta & : \quad \text{otherwise} \end{cases} \quad (12)$$

where $\eta < 1$. The value of η depends upon the amount of smoothing desired. Given a large amount of image noise, $\eta \approx 1$.

Since the α prior function is flat with two impulses, we only need to examine the value of equation (3) at three points: 0, 1, and α^*, as shown in Figure 3.

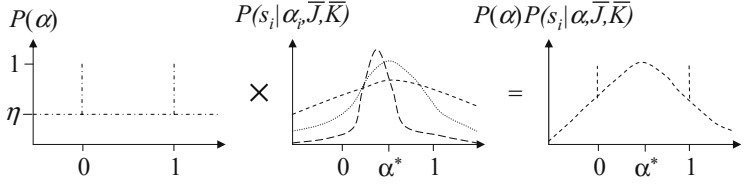


Fig. 3. Estimating the optimal value of α. Note that we only need to compare between α^* (see equation (11)), 0, and 1 because of our definition of the prior $P(\alpha)$.

Whichever is the maximum is assigned as the final value $\hat{\alpha}$ for our pixel, with a corresponding pixel color of $(1 - \hat{\alpha})\bar{J} + \hat{\alpha}\bar{K}$.

The quantitative error of this approach can be further decreased by forcing the red, green, or blue at a output pixel to be the value originally captured by the sensor, while setting the other two channels to be consistent with $\hat{\alpha}$. Because λ weights the central sample heavily, it is unlikely that these two values will be very different. Also, if the output sampling is done on a grid differing from the input grid, a source Bayer sample is not available at each pixel.

4.1 Multi-image Demosaicing

Using our gridless Baysian solution, information from multiple images can be introduced into the model without significantly altering the methodology. These supporting images are assumed to be similar, but not exactly the same, such as from subsequent video frames. Furthermore, these additional Bayer samples can be used without resampling them. To use supporting images, a per-image projective mapping must be computed to register each image to the first image.

As more images are added, we can shrink the neighborhood of samples used and still maintain sufficient samples to cluster into $\bar{J}$ and $\bar{K}$. By doing so, we can reduce the likelihood of it containing more than two representative colors.

If an R, G, and B sample each appear close to the sample we are reconstructing, nearest neighbor interpolation combination of Bayer samples could be used [18]. The notion of using weighted clustering and reconstruction remain the same as with a single input image. Once registered, even if the same color channel appears at the same pixel location, there is still a benefit: noise reduction.

By including supporting images, we risk introducing bad or misleading data. The global projective mapping does not account for all scene changes, such as lighting changes and moving objects. (It would be better to use robust local registration, but this is future work.) To avoid the effect of registration errors, we ensure that only data from the reference and supporting images that are locally similar are combined. Similarity is measured in RGB space using Sum of Absolute Differences (SAD) over a local window of 7×7, denoted as ϵ.

We handle multiple images by adjusting the definition of σ_i to include the correlation error between the reference and supporting images:

$$\sigma_i = \sigma_N(1 + \lambda\Delta_d)(1 + \tau\epsilon). \tag{13}$$

The term $(1 + \tau\epsilon)$ is the mismatch penalty, with $\tau = 0.1$ in our experiments.

To implement multi-image demosaicing, all that is required is adding nearby Bayer samples in the supporting images to the set S and using the above variance equation. Note that the samples in S are the Bayer sensor samples because using the original samples avoids the need for any resampling.

4.2 Super-Resolution

Another advantage of our statistical, grid-less approach is that any sampling grid can be used for the demosaicer's reconstruction, e.g., one with a greater

resolution than the original images. Because floating-point Euclidean distances are used for the statistical measures, a continuous value of α can be generated anywhere in the image. Due to the alignment of samples from multiple images, the value of α may encode edges and sharpness between the pixels in the original grid. We can exploit this to handle super-resolution within our framework.

When super-resolving, the statistical clustering and local neighborhood sizes can be slightly shrunk to capture fine details. Other than that, the system operates similarly as it did in the multi-image demosaicing case.

5 Results

In this section, we discuss perceptual interpretation of demosaicing quality. To verify our system, multiple image sets have been tested.

5.1 Measuring Demosaicing Error

The standard way to benchmark demosaicers requires ground truth captured with a fully specified color at each sample. The Kodak PhotoCD [11] dataset is a popular source, but the image resolutions are low. As a result, we chose to use our own image data sets that are of significantly higher resolution. In our experiments, we sampled Bayer patterns from ground truth images by choosing one channel at each pixel. Although this may not reflect the optical process by which Bayer patterns are captured in cameras, it is common practice for benchmarking. After demosaicing, PSNR is computed against ground truth. We compute PSNR in the red channel, although all channels give similar results.

We believe that there are better metrics for measuring the quality of demosaicing. One metric is to test SNR only in areas near large gradients. This focuses the metric on edge reconstruction and does not carry a penalty for noise reduction in smooth areas. We implemented such a metric by thresholding the log-space gradient magnitude to form a mask, then dilating it by 2 pixels.

Another method is to choose a perceptual metric that accounts for perceptual contrast masking [24] and for the viewing distance. The S-CIELAB metric has these features and a MATLAB implementation is available. This metric returns a numerical score of 1 for any just-noticeable error, with 10 indicating very high error. We compute the percentages of pixels exceeding scores of 3, 5, and 10.

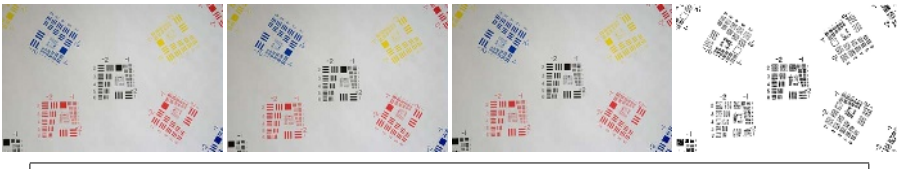
To test super-resolution, we create a reference by demosaicing a single image and bicubically upsampling it to have twice the horizontal and vertical resolution. We then perform three image demosaicings on the doubled resolution sampling grid to create the super-resolution output. By comparing results on calibration chart images, we can visually discern increased resolution.

5.2 Demosaicing Results

To test our algorithms, we chose three multi-image data sets and performed bilinear interpolation, High-Quality Linear Interpolation (HQLI) [10], and our method using a single image and using all three images. Both bilinear and HQLI

Table 1. Numerical analysis of demosaicing methods. We present three data sets and demosaic each with bilinear interpolation, high-quality linear interpolation, and our two color method using both a single input image and three input images. In addition to thumbnails of each input, edge masks (see Section 5.1) are also shown. PSNR is given for areas around edges and for the whole image. S-CIELAB statistics are stated as the percentage of the image with small (≥ 3), medium (≥ 5) and large (≥ 10) errors.

Crayons Data Set					
	Edges	**Whole Image**			
	PSNR	**PSNR**	**≥3CIE**	**≥5CIE**	**≥10CIE**
Bilinear	28.305 dB	31.406 dB	9.036 %	2.537 %	0.272 %
High-Quality LI	30.784 dB	33.716 dB	5.294 %	1.439 %	0.168 %
2 Color - 1 Image	32.119 dB	35.204 dB	4.682 %	1.366 %	0.159 %
2 Color - 3 Images	32.914 dB	35.972 dB	3.988 %	1.128 %	0.122 %



Calibration Charts Data Set					
	Edges	**Whole Image**			
	PSNR	**PSNR**	**≥3CIE**	**≥5CIE**	**≥10CIE**
Bilinear	21.746 dB	32.554 dB	5.526 %	1.806 %	0.268 %
High-Quality LI	24.213 dB	34.857 dB	4.039 %	0.889 %	0.016 %
2 Color - 1 Image	24.583 dB	35.574 dB	2.507 %	0.437 %	0.012 %
2 Color - 3 Images	25.507 dB	36.519 dB	1.826 %	0.339 %	0.012 %

Ship Data Set					
	Edges	**Whole Image**			
	PSNR	**PSNR**	**≥3CIE**	**≥5CIE**	**≥10CIE**
Bilinear	27.336 dB	34.263 dB	8.478 %	2.068 %	0.064 %
High-Quality LI	33.301 dB	39.047 dB	3.060 %	0.323 %	0.003 %
2 Color - 1 Image	34.458 dB	40.062 dB	2.428 %	0.262 %	0.005 %
2 Color - 3 Images	35.429 dB	40.770 dB	1.586 %	0.161 %	0.005 %

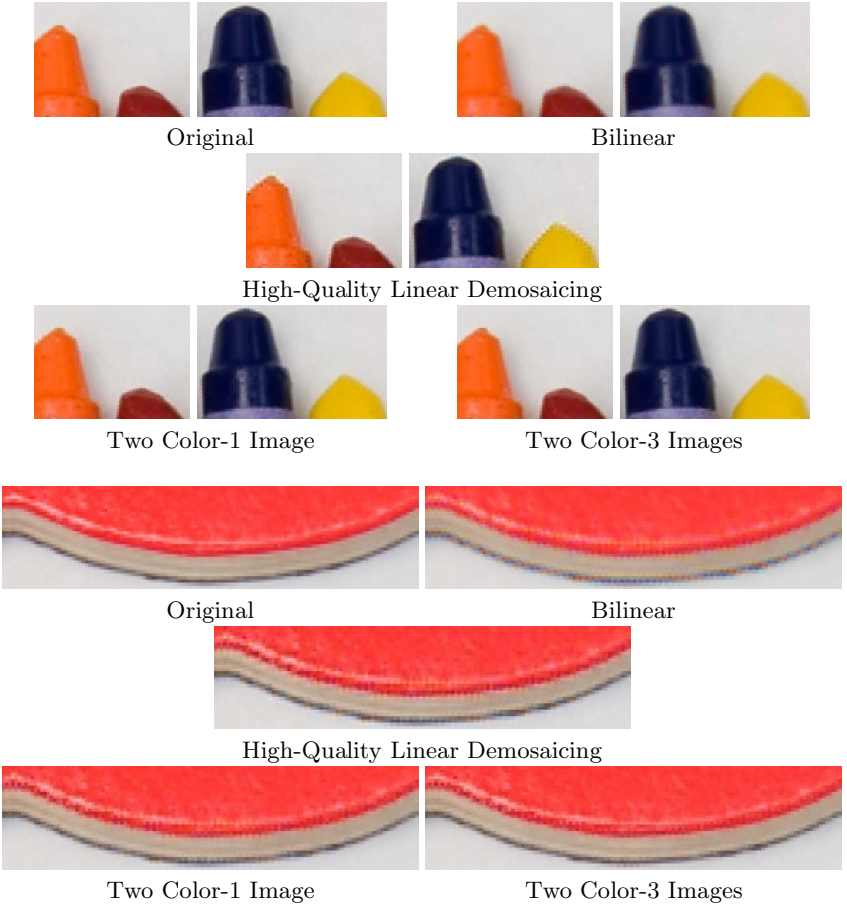


Fig. 4. Comparison of demosaicing two image regions using various demosaicing methods including bilinear, High-Quality Linear Demosaicing [10], and our method with one and three input images

were performed on a single image, as they are single Bayer image algorithms. HQLI was chosen as a representative algorithm both because of its visual performance and because it is our "bootstrap" algorithm. For a detailed comparison of HQLI results against many other previous algorithms, see [10]. The crayon and calibration chart datasets (Table 1) are static scenes taken with a moving camera, whereas the ship data set contains both scene and camera motion.

In Table 1, the PSNR, measured in dB, shows our single image method consistently outperforms the bilinear and HQLI methods for the entire image and for edges. The three image case further improves the results. Judging by the S-CIELAB metric, we also generally lower the number of offending pixels. Figure 5 shows a drop in pixel S-CIELAB error severity from the bilinear approach to our two color/three image technique.

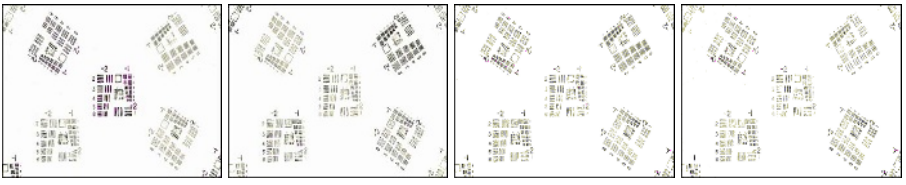


Fig. 5. Comparison of error among demosaicing methods as measured by the S-CIELAB metric. The darker the pixel, the more perceivable the error is. Pixels highlighted in purple have an error greater than 10 CIE, or very noticeable error.

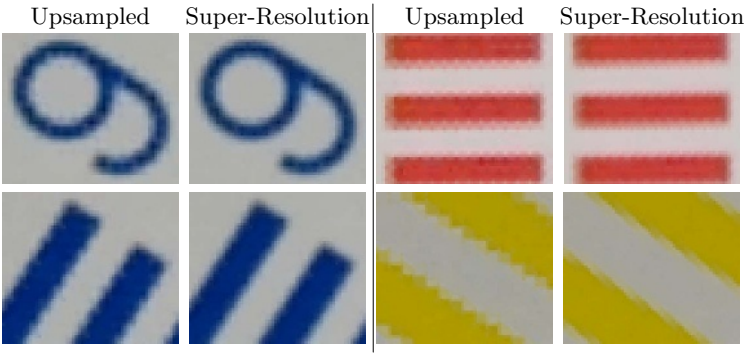


Fig. 6. Close-ups of our super-resolution results. For each image pair, the left is the result of our single image demosacing followed by bicubic up-sampling by a factor of two in each dimension. The right image is reconstructed using our three image demosaicing technique to directly get the up-sampled image. Notice our results are less blocky.

Figure 4 shows close-up views of the demosaiced crayon data. The bilinear method exhibits color fringing and aliasing for both examples, especially around the crayon tips and in the shadow of the red letter "Q". HQLI provides a cleaner result, but still has aliasing issues as well as difficulty handling the orange and yellow crayon edges. Our method provides sharper results with reduced artifacts in the single image and especially in the three image case.

The super-resolution results, shown in Figure 6, are good, but not dramatic. This is not surprising, because of the inherent limitations due to noise and discretization during image formation. Lin and Shum [28] showed that the practical limit is about 1.6 times the input resolution. Still, performing resolution enhancement with our method allows cleaner reconstruction using our image model.

6 Future Work

In Section 4, we assumed that all colors were equally likely to occur; this renders the denominator of equation (4) irrelevant. However, the system could be

refined by substituting actual image statistics, derived either directly from the Bayer samples or from an initial demosaicing pass. Similarly, the neighborhood sizes, both for the clustering and reconstruction stages, are set based on experiments to maximize sharpness and accuracy. These variables could be tweaked or automatically configured to target image conditions and desired results.

In this paper, we focussed on demosaicing Bayer CCDs because the Bayer pattern is the most common CFA and we begin our processing by using the output of a Bayer demosaicer as a "bootstrap". Because our algorithm is inherently free of the necessity of samples falling on a grid, we can support any color filter array pattern given an initial demosaicing guess. Furthermore, it would also be beneficial to cluster without having to "bootstrap" the algorithm. One possible approach would be to first cluster the color channels independently, resulting in three sets of clusters (one each in R, G, and B). These sets of clusters would then have to be reconciled into only two clusters in RGB space.

7 Conclusions

We have presented a demosaicing method and an image model that solves the ill-conditioned demosaicing problem within a well-conditioned Bayesian framework based on local sensor samples clustered into two RGB color values at each pixel. By modeling colors across an edge as linear combinations of the colors on each side, the possibility of inducing color fringing is decreased. Furthermore, the proposed statistical model is not grid-based, thus easily allowing for extensions to both multi-image demosaicing for video processing and non-iterative super-resolution output sampling. By constraining the output image to a linear model, we also reduce visible noise in smooth areas while preserving sharp edges.

References

1. Bayer, B.: Color imaging array. In: U.S. Patent No. 3,971,065. (1976)
2. Cok, D.: Signal processing method and apparatus for producing interpolated chrominance values in a sampled color image signal. In: U.S. Patent No. 5,373,322. (1987)
3. Freeman, W.: Median filter for reconstructing missing color samples. In: U.S. Patent No 4,724,395. (1988)
4. Ramanath, R., Snyder, W., Bilbro, G., Sander, W.: Demosaicking methods for bayer color arrays. J. of Electronic Imaging **11**(3) (2002) 306–315
5. Gupta, M., Chen, T.: Vector color filter array demosaicing. In: Procs. of SPIE, Sensors and Camera Systems for Scientific, Industrial, and Digital Photography Apps. II. Volume 4306. (2001) 374–382
6. Laroche, C., Prescott, M.: Apparatus and method for adaptively interpolating a full color image utilizing chrominance gradients. In: U.S. Patent No. 5,629,734. (1994)
7. Hamilton, J., Adams, J.: Adaptive color plane interpolation in single sensor color electronic camera. In: U.S. Patent No. 5,629,734. (1997)
8. Chang, E., Cheung, S., Pan, D.: Color filter array recovery using a threshold-based variable number of gradients. In: Procs. of SPIE/IS and T EI. Volume 3650. (1999)

9. Kimmel, R.: Demosaicing, image reconstruction from color CCD samples. IEEE Trans. on Image Processing **8** (1999) 1221–1228
10. Malvar, H., He, L., Cutler, R.: High-quality linear interpolation for demosaicing of bayer-patterned color images. In: Procs. of ICASSP. (2004)
11. Eastman Kodak Company: PhotoCD PCD0992 (http://r0k.us/graphics/kodak/)
12. Nayar, S., Narasimhan, S.: Assorted pixels: Multi-sampled imaging with structural models. In: ECCV. Volume 4. (2002) 189–205
13. Hel-Or, Y.: The canonical correlations of color images and their use for demosaicing. Technical Report HPL-2003-164R1, Hewlett Packard Labs. Israel (2004)
14. Brainard, D.: Bayesian method for reconstructing color images from trichromatic samples. In: Soc. for Imaging Science Technology Conf. (1995) 375–380
15. Chuang, Y., Curless, B., Salesin, D., Szeliski, R.: A bayesian approach to digital matting. In: CVPR. Volume 2. (2001) 264–271
16. Zomet, A., Peleg, S.: Multi-sensor super-resolution. In: Procs. IEEE Workshop on Apps. of Computer Vision. (2002) 27–31
17. Freeman, W., Jones, T., Pasztor, E.: Example-based super-resolution. IEEE Computer Graphics and Applications **22**(2) (2002) 56–65
18. Fung, J., Mann, S.: Projective demosaicing using multiple overlapping images. In: Procs. of the Int'l Symp. on Intelligent Multimedia, Video, and Speech Processing. (2004) 190–193
19. Gotoh, T., Okutomi, M.: Direct super-resolution and registration using raw CFA images. In: CVPR. Volume 2. (2004) 600–607
20. Zhang, X., Wandell, B.: A spatial extension of CIELAB for digital color image reproduction. In: Procs. of Soc. For Information Display. (1996) 731–734
21. Zhang, X., Wandell, B.: Color image fidelity metrics evaluated using image distortion maps. Signal Processing **70**(3) (1998) 201–214
22. Fairchild, M., Johnson, G.: The iCAM framework for image appearance, image differences, and image quality. J. of Electronic Imaging **13** (2004) 126–138
23. Pattanaik, S., Ferwerda, J., Fairchild, M., Greenberg, D.: A multiscale model of adaptation and spatial vision for realistic image display. In: ACM SIGGRAPH. (1998) 287–298
24. Daly, S.: The Visible Differences Predictor: An Algorithm for the Assessment of Image Fidelity. In: Digital Images and Computer Vision. MIT Press (1993) 179–206
25. Ramasubramanian, M., Pattanaik, S., Greenberg, D.: A perceptually based physical error metric for realistic image synthesis. In: ACM SIGGRAPH. (1999) 73–82
26. Walter, B., Pattanaik, S., Greenberg, D.: Using perceptual texture masking for efficient image synthesis. In: Procs. of Eurographics. (2002) 393–400
27. Lee, A., Mumford, D., Huang, J.: Occlusion models for natural images: A statistical study of a scale-invariant dead leaves model. IJCV **41** (2001) 35–59
28. Lin, Z., Shum, H.: Fundamental limits of reconstruction-based superresolution algorithms under local translation. IEEE Trans. PAMI **26**(1) (2004) 83–97