https://doi.org/10.2352/ISSN.2470-1173.2017.18.COLOR-057

A Color Image Model with Applications to Denoising

Megan M. Fuller, Jae S. Lim; Massachusetts Institute of Technology, Department of Electrical Engineering and Computer Science, Research Laboratory of Electronics; Cambridge, MA

Abstract

In this paper, we will present a model of color images that provides insight into how the color channels are related. We will show experimental results that illustrate the efficacy of this model. We will then demonstrate how this model can be used to design a simple chrominance based image denoising system.

Introduction

The image denoising problem has been widely studied. The paradigm most commonly addressed is that of black and white images degraded by additive white Gaussian noise, and it is not always straightforward to extend these methods to color images. The simplest way is to apply the method to each color channel separately, but this ignores the relationships among the color channels, leaving room for improvement in the results. In this paper, we will present a way to model the relationship between the color channels and show how this model can be applied to the problem of denoising.

In this section, we briefly review some of the techniques currently used to extend monochromatic denoising methods to color images. We will also review color image modeling. One idea is to treat each pixel as a three element vector. This is the approach taken in [2], [14], [13], and [6].

A related approach is to consider the image as existing in a higher dimensional space. This can be used to extend total variation minimization type denoising algorithms to color images. For example, one idea, proposed in [8] and [17] and expounded on in [10], is to treat a color image as a two-dimensional manifold embedded in a five-dimensional space (x,y,R,G,B) and then try to minimize the surface area in a way that is sensitive to edges. The details of how to compute the surface area are fairly technical and the reader is referred to [8] and [10] for a full explanation. However, it is relevant to this discussion in that the image model proposed in [9], though not directly incorporated into the method, is used to make certain parameter choices. We will discuss this model in greater detail later in the paper.

Other denoising algorithms first transform the RGB image to the YUV (or similar) domain. The less noisy luminance component can then be used to estimate model parameters, or the distance between pixel values can be computed based on some perceptual criteria. This approach is taken in [18], [19], [3], and [5].

Another technique is given in [16]. Here, the authors propose a wavelet shrinkage denoising algorithm. The wavelet coefficients are attenuated proportionally to the probability that they are “insignificant” in a precise sense defined in the paper. In computing that probability, the authors take into account the values of the corresponding coefficients in other color bands, the expectation being that if a particular coefficient is significant in one color band, it will be significant in the others also.

Some recent methods are based more explicitly on modeling. For example, in [15], the authors observe that the histograms of color images are well described by lines in RGB space. This observation is used for denoising in [20]. This model focuses on the relationship between the colors of neighboring pixels, rather than on the relationship between the color channels of a given pixel. However, it does suggest a locally affine relationship between the color channels. As another example, an affine model is also proposed in [21].

In the image denoising literature, we have found very little in the way of theory to explain spectral correlations in images. The relationship between color channels within an image has been explored more thoroughly in the demosaicking literature. A linear relationship between color channels was proposed in [1] and explained more precisely in [9]. We shall address this model in the next section.

The general observation that the color content is correlated at high frequencies has been used in many more recent systems. For example, in [7], the authors design a demosaicking system based on the idea that the difference between the red and green, and the blue and green color channels will consist of primarily low-frequency components, having eliminated the correlated high frequencies. The values of the differences can be extracted by a proper wavelet decomposition. In [22], the authors use this same idea to propose a method of compressing mosaicked images.

The rest of this paper is organized as follows. In the next section, we will take the simple model proposed by Kimmel in [9] and adjust it to denoising applications. Though Kimmel used it to choose the parameters of a denoising system in [10], our discussion will be more general and will use a more direct application of the model. We will then present results showing that this model is good in practice, and we will demonstrate how this model can be used to create a simple denoising system that often achieves nearly theoretically optimal performance, given that it does no spatial domain processing. Finally, we will conclude with some suggestions for future work.

Model

In [9], Kimmel proposes the following image model, assuming the camera is measuring the scene irradiance and the object being imaged reflects light diffusely.

$$R(x) = \rho_R(x)\langle \vec{N}(x), \vec{l}\rangle \tag{1}$$

$$G(x) = \rho_G(x)\langle \vec{N}(x), \vec{l}\rangle \tag{2}$$

$$B(x) = \rho_B(x)\langle \vec{N}(x), \vec{l}\rangle \tag{3}$$

where $R(x)$, $G(x)$, and $B(x)$ are the values of a pixel at location x, $\vec{N}(x)$ is the surface normal of the object being imaged, $\vec{l}$ is the light source, and $\rho_i(x)$ is the albedo of the object being imaged. The albedo is a function that captures the characteristics of how an object reflects light and is different for each wavelength. Note that $\rho_i(x)$ is a function of wavelength, as indicated by the subscript, and $\vec{N}(x)$ and $\vec{l}$ are vectors.

In Eqs. (1)- (3), it is assumed that the light is constant and white throughout the scene. Additionally, let us assume that in a small local area W, we are imaging a single object. Within the object, the albedo is approximately constant, and we can compute the local average of a pixel value for each color channel. For example, $R_{avg}(x)$ is defined as follows

$$\begin{aligned} R_{avg}(x) &= \frac{1}{|W|}\sum_{k\in W}\rho_R(x+k)\langle\vec{N}(x+k),\vec{l}\rangle && (4)\\ &\approx \frac{1}{|W|}\rho_R(x)\sum_k\langle\vec{N}(x+k),\vec{l}\rangle && (5)\end{aligned}$$

If we then compute $q(x) = \frac{R(x)}{R_{avg}(x)}$, we find from Eqs. (1) and (5) that

$$\begin{aligned} q(x) &= \frac{R(x)}{R_{avg}(x)} && (6)\\ &\approx \frac{\rho_R(x)\langle\vec{N}(x),\vec{l}\rangle}{\frac{1}{|W|}\rho_R(x)\sum_k\langle\vec{N}(x+k),\vec{l}\rangle} && (7)\\ &= \frac{|W|\langle\vec{N}(x),\vec{l}\rangle}{\sum_k\langle\vec{N}(x+k),\vec{l}\rangle} && (8)\end{aligned}$$

Notice that $q(x)$ is not a function of wavelength. In fact, it is straightforward to show that this is true even if the light is not white, as long as it is constant. Therefore, at any pixel, we can write the RGB values as

$$\begin{aligned} R(x) &= q(x)R_{avg}(x) && (9)\\ G(x) &= q(x)G_{avg}(x) && (10)\\ B(x) &= q(x)B_{avg}(x) && (11)\end{aligned}$$

That is, we can write the R, G, and B values as the product of the color dependent local mean and a color independent residual. We will refer to this as the multiplicative model. This is actually a special case of the model suggested by Zachevsky and Zeevi in [21], but it is simpler because it says the color channels are strictly proportional to each other, without an additive term. It also provides a clear physical interpretation of the model and a straightforward way to compute the model parameters.

Since the human visual system responds approximately logarithmically to changes in intensity [11], it can be useful to consider logarithmic domain processing [1]. In this case, the model becomes additive. Specifically,

$$\begin{aligned} \log(R(x)) &= \log(q(x)) + \log(R_{avg}(x)) && (12)\\ \log(G(x)) &= \log(q(x)) + \log(G_{avg}(x)) && (13)\\ \log(B(x)) &= \log(q(x)) + \log(B_{avg}(x)) && (14)\end{aligned}$$

For notational simplicity, we will define

$$f'(x) = \log(f(x)) \quad (15)$$

With this notation, Eqs. (12)- (14) can be expressed as

$$\begin{aligned} R'(x) &= q'(x) + R'_{avg}(x) && (16)\\ G'(x) &= q'(x) + G'_{avg}(x) && (17)\\ B'(x) &= q'(x) + B'_{avg}(x) && (18)\end{aligned}$$

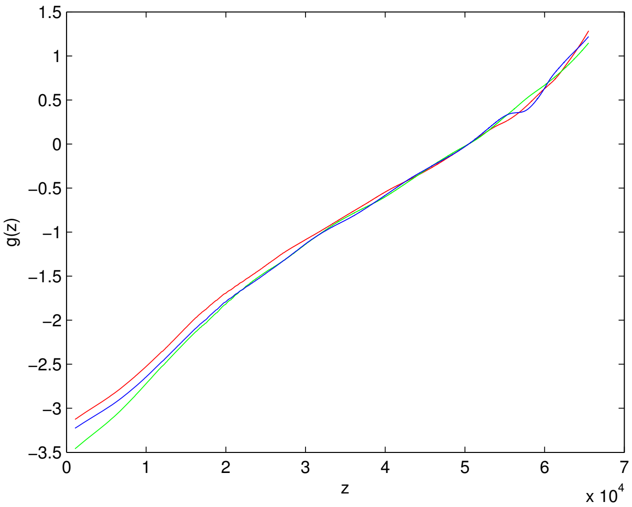


Figure 1: Camera calibration functions for the Sigma DP1.

Experimental Results

To test the efficacy of this model on real images, where a local patch may contain more than one object and there may be specular as well as diffuse reflection, we collected a database of 53 images taken with a Sigma DP1 camera. This camera uses a Foveon X3 sensor, so it actually measures red, green, and blue values at each pixel. There is no demosaicking [12]. Therefore, if the model works with these images, it is because the model is accurate, not because it correctly mimics the demosaicking process. The database was designed to include images with a wide variety of characteristics, with an emphasis on capturing many kinds of textures. A few examples will be shown later in the paper.

Since our model describes image irradiance, we used the camera calibration method proposed in [4] to determine how to map the values reported by the camera to scene irradiance measurements. This method uses several images of the same scene taken at different exposures and a user selected smoothing parameter to solve for the the camera calibration function, which is a mapping from z, the image value, to $g(z)$, a function which is equal to the logarithm of the scene irradiance up to an additive constant term that depends on the exposure value. For a more detailed discussion, see [4].

Our results are shown in Fig. 1. The colored lines are the calibration functions for the red, green, and blue channels, respectively. We used the same smoothing parameter in each channel, and chose it so that it was just high enough to ensure monotonicity of the resulting calibration functions.

From Fig. 1, we see that the values reported by the camera are very nearly proportional to $g(z)$. Note that we are not concerned with the constant of proportionality nor the arbitrary constant relating $g(z)$ to the log irradiance value, as both of these are uniform throughout the image and will simply be absorbed into the local average. The key point is that the image values are linearly related to the log irradiance, and so we are well justified in using the additive model of Eqs. (16)-(18) in the following experiments.

For computational reasons, we estimated the average $R'_{avg}(x)$

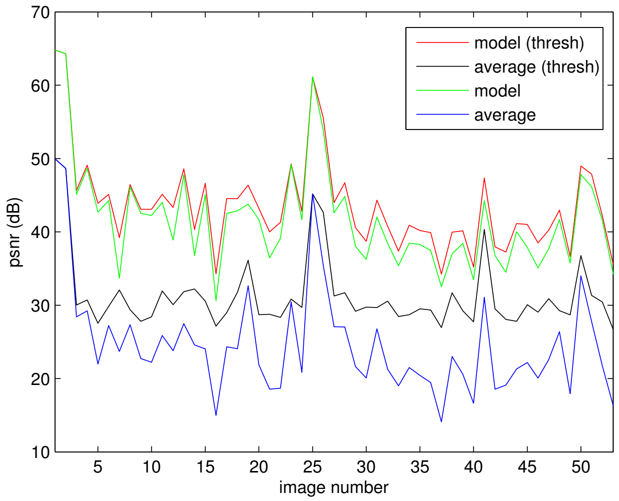


Figure 2: Modeling error for the images in our database.

as the average of the image values. That is, we computed

$$R'_{avg}(x) \approx \frac{1}{|W|} \sum_{k \in W} R'(x+k) \tag{19}$$

Though this is not exactly correct, if the pixel values being averaged are close to each other it will be approximately true, and we found that it did give good results.

To test the model, we computed an estimate of the residual by solving Eqs. (16)- (18) for $q'(x)$. For W, we used a 7x7 pixel square window centered at the pixel under consideration. In theory, the $q'(x)$ estimated from the three channels should be the same. In practice, however, they are not identical. To get a single value, we averaged them, as shown in Eq. (20).

$$\hat{q}'(x) = \frac{1}{3}(R'(x) - R'_{avg}(x) + G'(x) - G'_{avg}(x) + B'(x) - B'_{avg}(x)) \tag{20}$$

We used $\hat{q}'(x)$ to estimate the color channels as

$$\hat{R}'(x) = \hat{q}'(x) + R'_{avg}(x) \tag{21}$$

and so forth. We then computed the error between the modeled image and the original image. Results are plotted as the green line of Fig. 2. The x-axis of the figure is the number of the image in our database. These numbers are arbitrarily assigned and the order does not matter. Interpolation between non-integer values along the x-axis is not meaningful and is done only to make the figure visually cleaner. The y-axis gives the peak signal to noise ratio (psnr) of the modeled image, measured in dB.

For reference, the psnr of the local average is shown as the blue line in Fig. 2. That this is so much lower than the psnr of the modeled image demonstrates that the model does correctly add back high frequency information. The meaning of the red and black lines will be explained later in this section.

As can be seen in the figure, the model performs quite well in the psnr sense, the results dropping below 35dB in only the worst cases. The mean is about 44dB. Artifacts consist primarily of "bleeding" around places in the image where the color abruptly changes, which we will refer to as "chromatic edges" and are due to the model resting on the incorrect assumption that there is only one object in the region over which the local mean is computed.

Since psnr does not necessarily reflect the subjective visual quality of an image, we provide a few examples in Fig. 3- 5. Fig. 3 is a fairly flat image with no object edges and little texture. This is a case of very good performance of the model. Fig. 4, on the other hand, is highly textured with many chromatic edges. This is a case of relatively poor performance of the model. Nevertheless, even in this image, the modeled image is essentially visually indistinguishable from the original, though there are artifacts present that will be looked at more closely in the next example.

Finally, Fig. 5 is a more natural image that contains flat regions, textured regions, and dramatic chromatic edges. This demonstrates a case of slightly below-average performance of the model. A close up is shown in Fig. 6. In the closeup, we can see edge artifacts at the boarder of the red truck. These arise because the model rests on the assumption that there is only one image in the local region being averaged. At edges, this is not true.

Despite the edge artifacts, the modeled images shown do look very similar to the original images. This demonstrates that the high psnrs of the modeled images do in fact indicate that the model works well.

The errors at edges can be mitigated by adaptively changing the averaging window to include only pixels on the same side of the edge as the pixel under consideration. We implemented this by considering only those pixels whose values were within a certain threshold T of the value of the pixel under consideration.

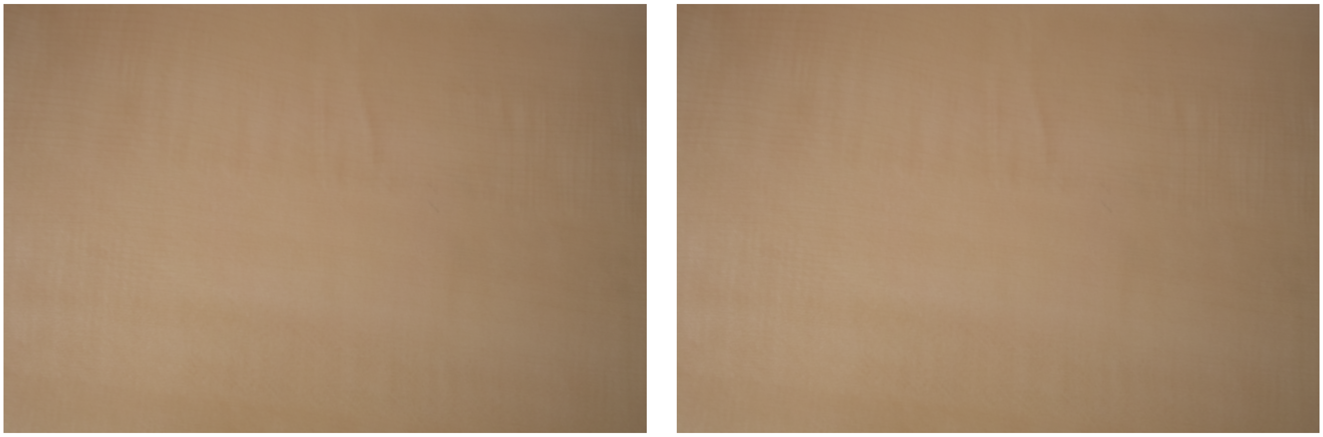
The choice of the threshold T was based on the following considerations. On the one hand, if T is too large, no edges will be detected. On the other hand, if T is too small, then the set of pixels being averaged will degenerate to only the pixel under consideration. In this case, the "local average" will just be the pixel under consideration and the residual will be zero. Since all the residuals are trivially the same, the model will appear to produce perfect results, though obviously this is not the intent. The dynamic range of the images was 2^{16}, and we empirically found that a threshold of 10,000 gave good results, detecting many edges without collapsing to degeneracy.

These results are plotted as the red line in Fig. 2, which shows that thresholding does improve the psnr of the modeled image, as expected. The black line shows the local average computed with thresholding. Although the local average does improve with thresholding, modeling still adds back considerable information, leading to much better performance. This demonstrates that we have avoided degeneracy. A visual example is given in Fig. 7, which shows that the thresholding does remove the edge artifacts. We use this method in the rest of the paper.

Application to Denoising

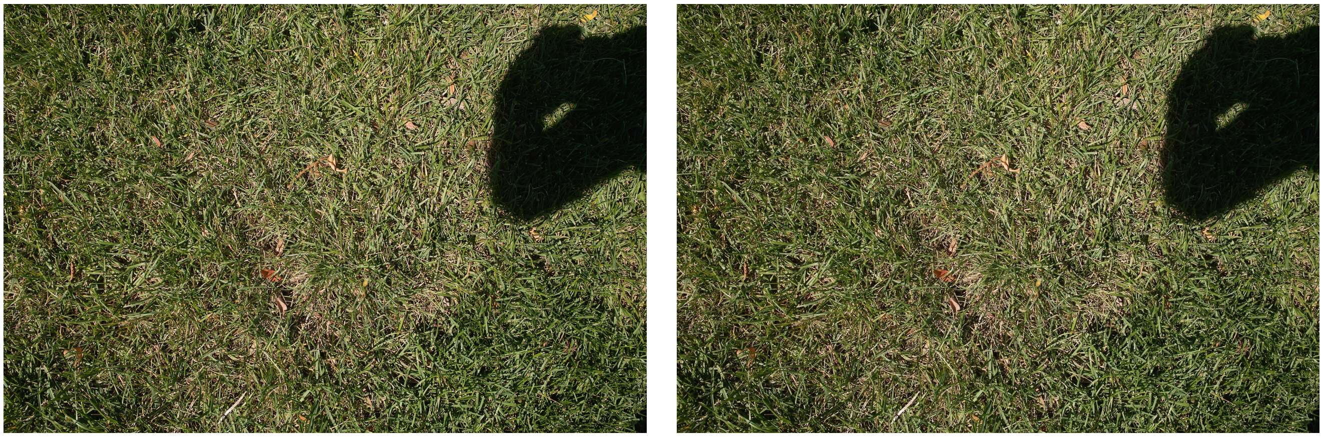
This model can be applied to denoising in a very straightforward way. Specifically, we can compute $\hat{q}'$, $\hat{R}'$, $\hat{G}'$, and $\hat{B}'$ as given in Eqs. (20) and (21) using the noisy image as R', G', and B'. This assumes the noise is independent across color channels.

If the model were exactly correct and the local means were known perfectly, we would expect the noise variance to be reduced, on average, by a factor of three, corresponding to a psnr increase of $10\log_{10}(3) \approx 4.77$dB. However, since the local means



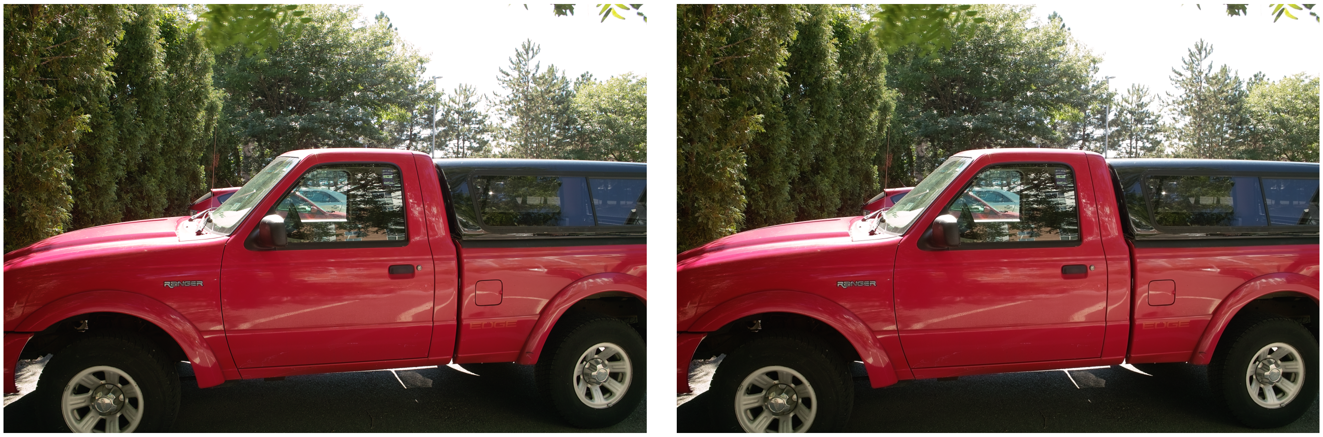
(a) Original (b) Modeled

Figure 3: Results of reconstructing image 1 with the proposed model.



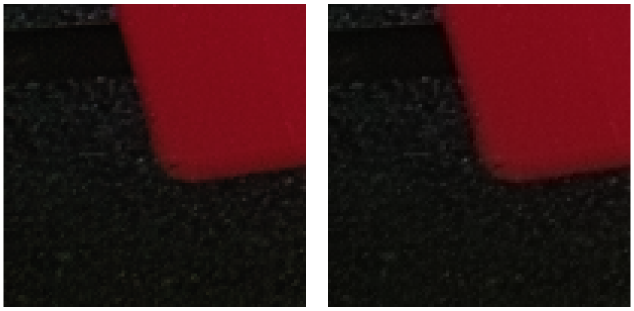
(a) Original (b) Modeled

Figure 4: Results of reconstructing image 16 with the proposed model.



(a) Original (b) Modeled

Figure 5: Results of reconstructing image 47 with the proposed model.



(a) Original (b) Modeled

Figure 6: Close up of image 47 and the result of modeling.

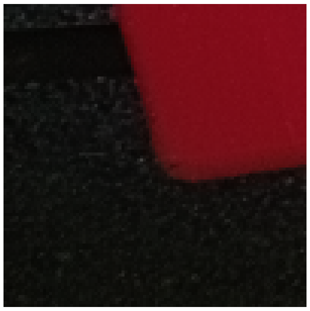


Figure 7: Close up of the result of modeling image 47 using a threshold.

are being estimated from the noisy image using at most a 7x7 pixel window, the actual potential increase is closer to 4.60dB.

To test this, we took the images from our database and added white Gaussian noise ($\sigma = 5000$) to each color channel. We then processed them with the denoising system just described. We used a threshold to select only similar pixels to compute the average. Unlike the noiseless case, there is no danger of degeneracy here. Too small a threshold will simply give back the noisy image again. Empirically, we found a threshold of seven times the standard deviation of the noise to work well. The results are shown in Fig. 8. As in Fig. 2, the x-axis is the image number and the y-axis is the psnr. The red line is the error of the noisy image. The blue line is the result of performing chrominance based denoising, and the green line is simply 4.60dB more than the red line at each point, giving the theoretical limit of the system performance.

These results are quite good considering that this denoising system is not exploiting spatial correlation. Given the fact that only one pixel is used to do the denoising, the results are in many cases almost as good as theoretically possible. Example images are shown in Figs. 9- 11. Figs. 9a, 10a, and 11a are fragments of Figs. 3a, 4a, and 5a, respectively, with noise added. Figs. 9b, 10b, and 11b are the results of doing chrominance based denoising on Figs. 9a, 10a, and 11a, respectively. These results were typical of the images in our database and demonstrates that the increase in psnr does in fact mean an increase in subjective visual quality.

We performed similar experiments on pictures taken with other cameras, for example, the Kodak database, and found that the type of camera used did not have a significant effect on the results.

We note that this system could be paired with a black and white image denoising system to create a color image denoising method that exploits both the spatial and the spectral correlations

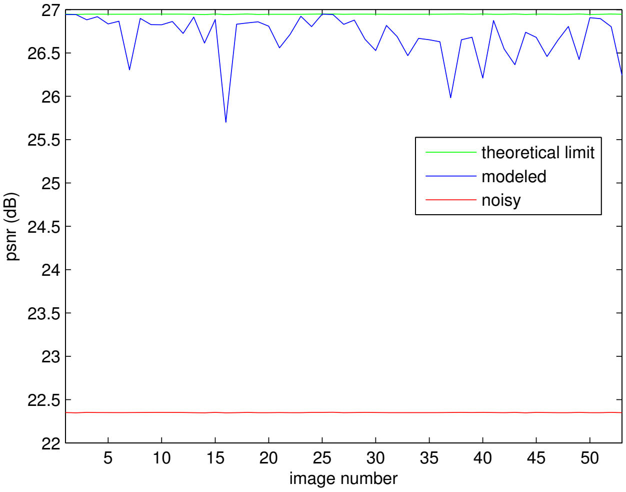


Figure 8: Chrominance based denoising results.

of images. We will not explore this possibility in this paper.

Conclusions and Future Work

In this paper, we have presented a model of color images from the early work on image demosaicking and have reformatted the model to be more applicable to image denoising. We have shown experimentally that this model is effective in describing real images, and we have demonstrated that the model can be applied to a basic chrominance based denoising system to produce good results.

Additionally, we note that this model could provide other ways to extend black and white denoising methods to color images. As one example, a block-based denoising algorithm such as BM3D or non-local means could compute the residuals $q(x)$ of each color channel. Since these residuals are expected to be the same for all channels, it would be possible to search through all three of them for similar blocks, thus increasing the amount of available data and possibly leading to better results than are currently achieved.

Finally, there is nothing about this model which limits it to RGB images. We expect it could be successfully extended to hyperspectral images. Indeed, the more spectral channels that are available, the better the simple chrominance based denoising system presented here is likely to perform.

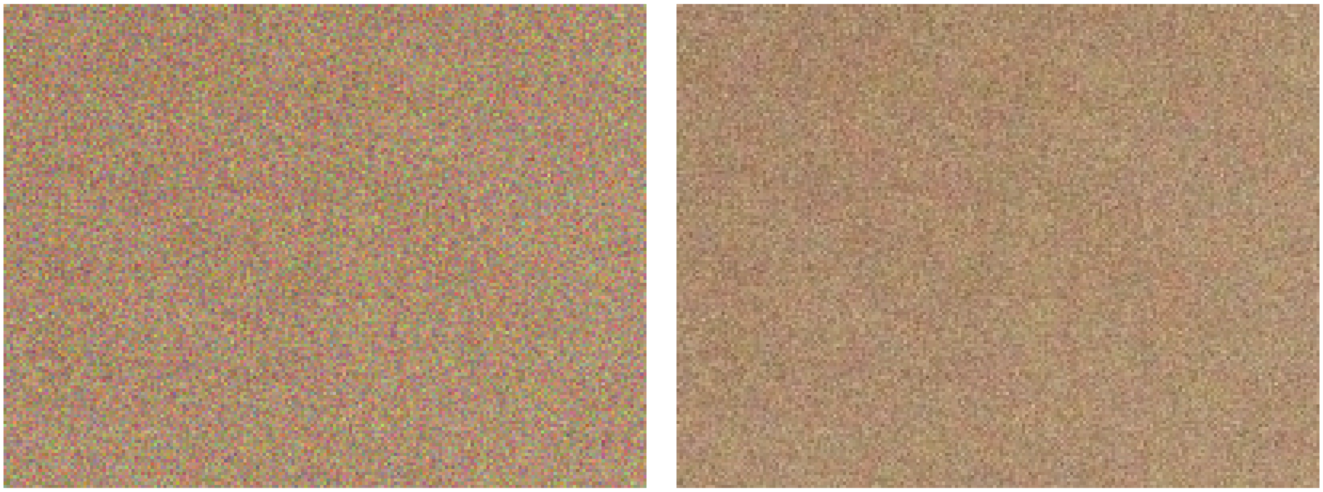
References

[1] J. E. Adams. Interactions between color plane interpolation and other image processing functions in electronic photography. Proc. SPIE, 2416:144–151, 1995.

[2] A. Buades, B. Coll, and J. M. Morel. A review of image denoising algorithms, with a new one. Multiscale Model. Simul., 4(2):490–530, 2005.

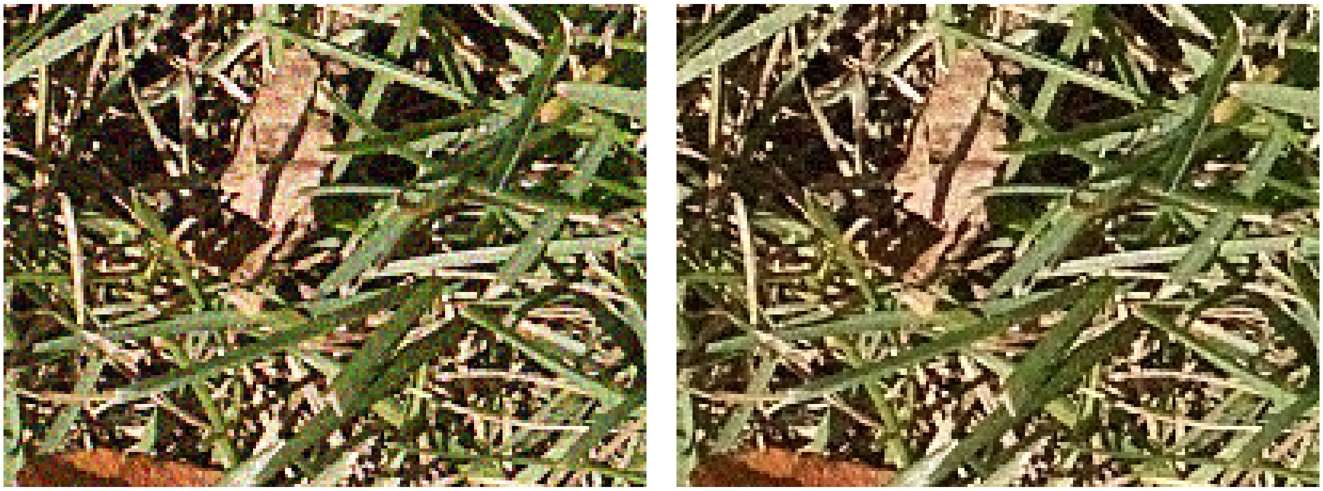
[3] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian. Image denoising by sparse 3-d transform-domain collaborative filtering. IEEE Trans. on Image Proc., 16(8):2080–2095, August 2007.

[4] P. E. Debevec and J. Malik. Recovering high dynamic range radiance maps from photographs. In ACM SIGGRAPH 2008 Classes, SIGGRAPH '08, pages 31:1–31:10, 2008.



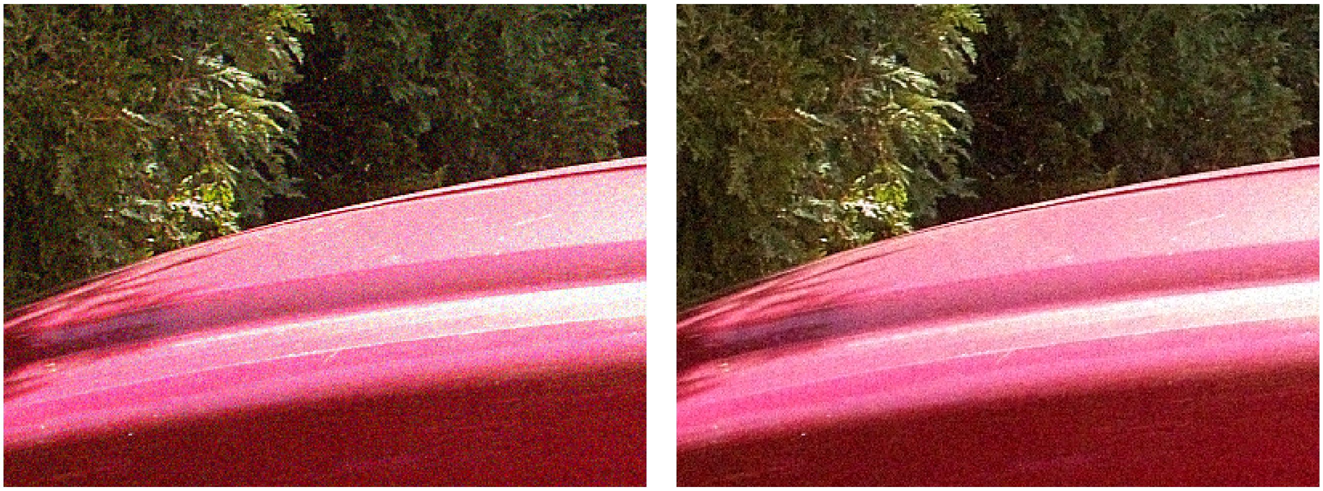
(a) Noisy (b) Denoised

Figure 9: Results of chrominance based denoising on image 1.



(a) Noisy (b) Denoised

Figure 10: Results of chrominance based denoising on image 16.



(a) Noisy (b) Denoised

Figure 11: Results of chrominance based denoising on image 47.

[5] A. Foi, V. Katkovnik, and K. Egiazarian. Pointwise shape-adaptive dct for high-quality denoising and deblocking of grayscale and color images. IEEE Trans. on Image Proc., 16(5):1395–1411, May 2007.

[6] K. He, J. Sun, and X. Tang. Computer Vision – ECCV 2010: 11th European Conference on Computer Vision, Heraklion, Crete, Greece, September 5-11, 2010, Proceedings, Part I, chapter Guided Image Filtering, pages 1–14. 2010.

[7] K. Hirakawa, X. L. Meng, and P. J. Wolfe. A framework for wavelet-based analysis and processing of color filter array images with applications to denoising and demosaicing. In ICASSP, volume 1, pages 597–600, Honolulu, HI, April 2007.

[8] R. Kimmel. Computer Vision — ACCV'98: Third Asian Conference on Computer Vision Hong Kong, China, January 8–10, 1998 Proceedings, Volume I, chapter A natural norm for color processing, pages 88–95. Berlin, Heidelberg, 1997.

[9] R. Kimmel. Demosaicing: Image reconstruction from color ccd samples. IEEE Trans. on Image Proc., 8(9):1221–1228, September 1999.

[10] R. Kimmel, R. Malladi, and N. Sochen. Images as embedded maps and minimal surfaces: movies, color, texture, and volumetric medical images. International Journal of Computer Vision, 39(2):111–129, 2000.

[11] J. S. Lim. Two-Dimensional Signal and Image Processing. Prentice Hall, 1990.

[12] R. F. Lyon and P. M. Hubel. Eyeing the camera: Into the next century. In Proc. IS&T/SID 10th Color Imaging Conf., 2002, pages 349–355, 2002.

[13] J. Mairal, M. Elad, and G. Sapiro. Sparse representation for color image restoration. IEEE Trans. on Image Proc., 17(1):53–69, January 2008.

[14] J. J. McAuley, T. S. Caetano, A. J. Smola, and M. O. Franz. Learning high-order mrf priors of color images. In Proc. of the 23rd International Conference on Machine Learning, Pittsburgh, PA, 2006.

[15] I. Omer and M. Werman. Color lines: image specific color representation. In CVPR, volume 2, pages 946–953, 2004.

[16] A. Pizurica and W. Philips. Estimating the probability of the presence of a signal of interest in multiresolution single- and multiband image denoising. IEEE Trans. on Image Proc., 15(3):654–665, March 2006.

[17] N. Sochen, R. Kimmel, and R. Malladi. A general framework for low level vision. IEEE Trans. on Image Proc., 7(1):310–318, March 1988.

[18] J. L. Stark, E. J. Candes, and D. L. Donoho. The curvelet transform for image denoising. IEEE Trans. on Image Proc., 11(6):670–684, June 2002.

[19] C. Tomasi and R. Manduchi. Bilateral filtering for gray and color images. In ICCV, pages 839–846, January 1998.

[20] C. Tsai, W. Tu, and S. Chien. Efficient natural color image denoising based on guided filter. In IEEE ICIP, pages 43–47, Quebec City, QC, September 2015.

[21] I. Zachevsky and Y. Y. Zeevi. Denoising of natural stochastic colored-textures based on fractional brownian motion model. In IEEE ICIP, pages 1065–1069, Quebec City, QC, September 2015.

[22] N. Zhang and X. Wu. Lossless compression of color mosaic images. IEEE Trans. on Image Proc., 15(6):1379–1388, June 2006.

Author Biography

Megan M. Fuller received her B.S. in electrical engineering from Brigham Young University in 2012 and her M.S. in electrical engineering from the Massachusetts Institute of Technology in 2014. She is currently a graduate student at MIT working towards a Ph.D. in electrical engineering. Her work has focused on image restoration.

Jae S. Lim received the S.B., S.M., E.E., and Sc.D. degrees in EECS from the Massachusetts Institute of Technology in 1974, 1975, 1978, and 1978, respectively. He joined the M.I.T. faculty in 1978 and is currently a Professor in the EECS Department.

His research interests include digital signal processing and its applications to image, video and speech processing. He has contributed a number of articles to journals and conference proceedings. He is a holder of more than 40 patents in the areas of advanced television systems and signal compression. He is the author of a textbook, Two-Dimensional Signal and Image Processing. He is the recipient of many awards including the Senior Award from the IEEE ASSP Society and the Harold E. Edgerton Faculty Achievement Award from MIT. He is a member of the Academy of Digital Television Pioneers and a fellow of the IEEE.