Derivation, Experimental Verification, and Applications of a New Color Image Model

by

Megan Fuller

B.S. Brigham Young University (2012)
M.S. Massachusetts Institute of Technology (2014)

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy in Electrical Engineering and Computer Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2018

Author .....
Department of Electrical Engineering and Computer Science
May 9, 2018

Certified by .....
Jae S. Lim
Professor of Electrical Engineering
Thesis Supervisor

Accepted by .....
Leslie A. Kolodziejski
Professor of Electrical Engineering
Chairman, Department Committee on Graduate Students

Derivation, Experimental Verification, and Applications of a New Color Image Model

by

Megan Fuller

Submitted to the Department of Electrical Engineering and Computer Science
on May 9, 2018, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy in Electrical Engineering and Computer Science

Abstract

Image modeling is an important area of image processing. Good image models are useful, for example, in image restoration problems because they provide constraints that can be imposed on degraded images to retrieve better approximations of the original image. Many physical models of images are separable functions of position and wavelength, which means that images can be written as a color independent local average multiplied by a color dependent residual. We will present experimental results showing that this is the case in practice, and discuss the limitations of this model. We will also show that several commonly used observations in image processing follow from this model. Finally, we will demonstrate the results of imposing the model constraints in several image denoising problems and show that degraded images can be improved by imposing the model constraints.

Thesis Supervisor: Jae S. Lim
Title: Professor of Electrical Engineering

Acknowledgments

First and foremost, I would like to thank my adviser, Professor Jae S. Lim. I have learned so much from him. The freedom he gave me to explore my own ideas and the confidence it implied have been invaluable to my growth as a researcher. He has shown me by example the value in being meticulous and thorough, not just in research but also in teaching. As his TA, I have seen the thought and preparation that go in to his classes, and I know that my own teaching has improved as a result. I would also like to thank the other members of my thesis committee, Dr. John Apostolopoulos and Professor Fredo Durand, both of whom made many good suggestions that have greatly improved the quality of this thesis.

My time at MIT has been greatly enriched by my friends and labmates, Xun Cai and Lucas Nissenbaum. The many long conversations I've had with them have been a highlight of my graduate school experience. Cindy LeBlanc also was always looking out for me, which was an enormous support.

I have had the opportunity to work with many fantastic people over the course of my education. Bertice Tise and Brian McMurtrey were excellent mentors during my internship at Sandia National Labs, and I will always be grateful for the time I got to spend learning from them. During my time at BYU I was able to work with Dr. Brian Jeffs, Dr. Michael Wirthlin, and Brian Pratt, all of whom were unfailingly kind and patient.

I would also be remiss not to thank Michael Nord, who got me started on this path.

Finally, I would like to thank my family: my parents, Ned and Julie, who have always been there for me and have been an even greater help than they probably realize; my sisters, Rachel, Emily, and Jessica, who have been a huge source of love and encouragement; and my grandpa, Reed Jensen, whom I respect more than I can say. I could not have done this without them.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

What is an image? The most general answer is that an image is a two-dimensional signal. This definition fails to capture the structure inherent in the signals we think of as "images"–a randomly generated two-dimensional signal will look like noise. It is only a tiny fraction of the possible two-dimensional signals that contain the kind of information associated with the word "image."

Image models attempt to define this expected structure mathematically. This is desirable in many applications. For example, a good image model would allow us to reconstruct natural images from degraded ones. It could also aid in compression or computer vision by reducing the image from its full complexity to a smaller number of model parameters.

Many different image models have been proposed, some of which will be discussed in detail later on. In this thesis, a new model will be derived. This new model is based on physical image models. Specifically, it can be derived from any physical model that can be written as the product of a function only of position and a function only of wavelength. It describes the relationship between the color channels of an image. In particular, it says that the color channels can be written as the product (or, in the logarithmic domain, the sum) of a color dependent local average and a color independent residual.

This thesis is organized as follows. In Chapter 2, an overview of the physical models from which the new model can be derived will be given. In Chapter 3, the

derivation will be presented. In Chapter 4, experimental results verifying the accuracy of the model will be shown, and in Chapter 5, some of the limitations of the model will be discussed. In Chapter 6, certain implications of the model will be explained, and in Chapter 7, the effect of imposing the constraints described by the model in different image processing situations will be demonstrated. Finally, conclusions will be presented in Chapter 8.

Chapter 2

Physical Image Models

In this chapter, several physical image models will be discussed. First, the general mechanisms of reflection will be explained. Then, a few specific models will be presented and it will be shown that they can be represented or approximated as separable functions of position and wavelength. (A separable function of position and wavelength is one that can be written as the product of a function only of wavelength and a function only of position.) This will be important for the derivation of the model, which will be given in the next chapter. After presenting these physical models, we will discuss retinex theory, which highlights certain strengths and weaknesses of the physical models. We will conclude with a brief look at the practical applications of image modeling generally.

2.1 Reflectivity

Images are formed by the sensors of a camera measuring the intensity of a small spectrum of wavelengths of light over a small area. This light comes from three sources:

1. Specular reflections from objects in the scene.
2. Diffuse reflections from objects in the scene.
3. Ambient light.

The sum of the light from these three sources is the value measured by the camera. If there are multiple light sources in the scene, the contributions for each are summed. In this chapter and throughout the thesis, a single light source will be assumed.

Specular reflections are due to light bouncing off the surface of an object without penetrating it. The reflection occurs at an angle equal to the incoming light. That is, if the incoming light hits a vertical surface at an angle θ below the horizontal, the reflected light will be at an angle θ above the horizontal. Perfect mirrors are pure spectral reflectors, and objects that primarily reflect specularly tend to have glossy textures.

Diffuse reflections are due to light penetrating a surface slightly before being reflected due to interactions with the material on a microscopic level. These interactions are difficult to define with precise theory because they depend on very specific parameters of the material, but some models that have been proposed will be discussed in the following section. Diffuse reflection produces matte textures–rough wood is an example.

Most objects have both specular and diffuse properties, so that some light is reflected specularly and some is reflected diffusely. The models discussed in the next section focus on diffuse reflections, but some account for spectral reflections also. Few incorporate ambient light, which will also be ignored here.

The quantity used to characterize the reflecting properties of a surface (assuming the scene irradiance is unpolarized and uniform over a large enough area and the surface is uniform and isotropic) is the bidirectional reflectance-distribution function (BRDF), given by

$$f_r(\theta_i, \phi_i; \theta_r, \phi_r) = \frac{dL_r(\theta_i, \phi_i; \theta_r, \phi_r; E_i)}{dE_i(\theta_i, \phi_i)} \tag{2.1}$$

where f_r is the BRDF, θ_i and ϕ_i are respectively the polar and azimuth angles of the incoming light, θ_r and ϕ_r are respectively the polar and azimuth angles of the receiver, dL_r is the differential reflected radiance in direction (θ_r, ϕ_r) due to incident irradiance E_i, and dE_i is the differential incident irradiance in direction (θ_i, ϕ_i). Essentially, the BRDF articulates how much of the reflected radiance in one direction is due to the

incident irradiance from another [48–50].

Notice that the BRDF is extraordinarily general. To learn it for any arbitrary situation would require measuring the incident irradiance and reflected radiance at each point, which of course is not feasible. Furthermore, the BRDF defined above is implicitly a function of wavelength, with each wavelength requiring the specification of a new f_r. Fortunately, there are many situations in which we can make reasonable assumptions about the geometry and reflective properties of the material and derive appropriate models. These models will usually not define the BRDF explicitly, but it could be derived from the relationships given.

2.2 Specific Models

In this section, several specific models that have been proposed to describe reflections of light off objects will be presented in detail. The purpose of this section is not to give the reader a deep understanding of the derivation, or even the meaning, of the models described. The discussion will be limited in scope to how the reflectance models can or cannot be represented as separable functions of position and wavelength.

The notation in the following subsections follows the notation of the original authors as closely as possible, and so it is not consistent across the different models.

2.2.1 The Lambertian Model

One of the oldest image models is the Lambertian model, which has been used in many applications such as demosaicing [33], denoising [32, 34, 63], and shape-from-shading [69]. This model is based on the assumption that the material being imaged is perfectly matte–that is, that there are only diffuse reflections–and that these diffuse reflections are equal in all directions, so the angle of the viewer is irrelevant. It also assumes the lightsource is infinitely distant from the object, so the angle of the incoming light is not a function of position. Under these assumptions, the image values are given by

$$I(x, \lambda) = \rho(\lambda)\langle \vec{N}(x), \vec{l}(\lambda) \rangle \tag{2.2}$$

where

- I is the reflected light and is a function of both position and wavelength.
- ρ is the diffuse albedo of the material. If there is only a single material being imaged, it can reasonably be assumed that this is constant in position, but it is dependent on wavelength.
- $\vec{N}$ is the normal vector of the surface being imaged. It is a function of position but not wavelength.
- $\vec{l}$ is the incoming light. It is assumed constant throughout the scene, so it is a function of wavelength, but not position.

Applying the definition of $\langle \cdot \rangle$ and noting that by definition $|N(x)| = 1$, (2.2) can be rewritten as

$$I(x, \lambda) = \rho(\lambda)|\vec{l}(\lambda)| \cos(\theta(x)) \tag{2.3}$$

where $\theta(x)$ is the angle between $\vec{N}(x)$ and $\vec{l}(\lambda)$ at location x. Notice that (2.3) is a separable function of position and wavelength.

Additionally, if the assumption that the lightsource is infinitely distant from the object is removed then the incoming light is a function of position and (2.2) becomes

$$I(x, \lambda) = \rho(\lambda)\langle \vec{N}(x), \vec{l}(x, \lambda) \rangle \tag{2.4}$$

and (2.3) becomes

$$I(x, \lambda) = \rho(x)|\vec{l}(x, \lambda)| \cos(\theta(x)) \tag{2.5}$$

where $\theta(x)$ is still the angle between the incoming light and the surface normal at location x, which is not a function of wavelength. If the locations of interest are close enough together that the magnitude of the incoming light can be assumed to be independent of position, this is of the same form as (2.3), and is again a separable function of position and wavelength.

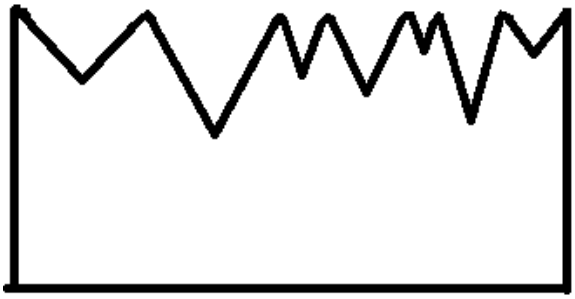


Figure 2-1: Illustration of textured Lambertian material.

2.2.2 The Oren-Nayar Model

Even in perfectly matte materials, the reflection may not be Lambertian. One such case is a textured Lambertian material. In [53–56], Oren and Nayar develop a model for this case, assuming the texture is due to V facets in the surface, as shown in Fig. 2-1. It is assumed the facets are small relative to the pixel size.

The model proposed takes into account shadowing and masking of the facets as well as one interreflective bounce–that is, the light bouncing off one edge and hitting another before being propagated to the camera. However, their model is also only an approximation because of the presence of some difficult integrals in the derivation. Nevertheless, they present results showing that their functional approximation generally matches up with the numeric solution of the integrals quite closely.

The model is defined as

$$I(\theta_r, \theta_i, \phi_r - \phi_i, \sigma, \lambda) = I_1(\theta_r, \theta_i, \phi_r - \phi_i, \sigma, \lambda) + I_2(\theta_r, \theta_i, \phi_r - \phi_i, \sigma, \lambda) \tag{2.6}$$

where

- θ_r is the polar angle of the viewer.
- θ_i is the polar angle of the source.
- ϕ_r is the azimuth angle of the viewer.
- ϕ_i is the azimuth angle of the source.

- σ is the standard deviation of the distribution of the facet slopes in the material and can be thought of as a measure of the roughness of the texture.
- λ is the wavelength.
- I_1 is the contribution from direct illumination.
- I_2 is the contribution from the interreflective bounce.

I_1 is given by

$$I_1(\theta_r, \theta_i, \phi_r - \phi_i, \sigma, \lambda) = \frac{\rho(\lambda)}{\pi} E_0(\lambda) \cos\theta_i [C_1(\sigma) + \cos(\phi_r - \phi_i) C_2(\alpha, \beta, \phi_r - \phi_i, \sigma) \tan(\beta) + (1 - |\cos(\phi_r - \phi_i)|) C_3(\alpha, \beta, \sigma) \tan\left(\frac{\alpha + \beta}{2}\right)]$$

where

- ρ is the diffuse albedo and is a function of wavelength.
- E_0 is the irradiance when the facet is illuminated head-on and is a function of wavelength.
- $C_1 = 1 - .5\left(\frac{\sigma^2}{\sigma^2 + .33}\right)$
- $C_2 = \begin{cases} .45\left(\frac{\sigma^2}{\sigma^2+.09}\right)\sin(\alpha) & \cos(\phi_r - \phi_i) \geq 0 \\ .45\left(\frac{\sigma^2}{\sigma^2+.09}\right)\left(\sin(\alpha) - \left(\frac{2\beta}{\pi}\right)^3\right) & \text{otherwise} \end{cases}$
- $\alpha = \max[\theta_r, \theta_i]$
- $\beta = \min[\theta_r, \theta_i]$
- $C_3 = .125\left(\frac{\sigma^2}{\sigma^2+.09}\right)\left(\frac{4\alpha\beta}{\pi^2}\right)^2$

I_2 is given by

$$I_2(\theta_r, \theta_i, \phi_r - \phi_i, \sigma, \lambda) = .17\frac{\rho(\lambda)^2}{\pi} E_0(\lambda) \cos(\theta_i) \left(\frac{\sigma^2}{\sigma^2 + .13}\right) \left[1 - \cos(\theta_r - \theta_i)\left(\frac{2\beta}{\pi}\right)^2\right] \tag{2.7}$$

where the symbols have the same meaning as previously stated.

This model is quite complicated. However, if we note that only ρ and E_0 are functions of wavelength and we consider all the angles to be encapsulated in a single position variable x, we can write

$$I_1(x, \lambda) = \rho(\lambda) E_0(\lambda) f_1(x) \tag{2.8}$$

$$I_2(x, \lambda) = \rho(\lambda)^2 E_0(\lambda) f_2(x) \tag{2.9}$$

Then (2.6) becomes

$$I(x, \lambda) = I_1(x, \lambda) + I_2(x, \lambda) \tag{2.10}$$

$$= \rho(\lambda) E_0(\lambda) f_1(x) + \rho(\lambda)^2 E_0(\lambda) f_2(x) \tag{2.11}$$

This is not a separable function of position and wavelength because of the presence of the term due to the interreflective bounce.

It is therefore natural to ask how significant the second bounce is. Notice that Oren and Nayar do not account for any higher order bounces. The contributions from these would be proportional to at least the cube of the albedo, which is always between zero and one, and are not large enough to be considered. Whether the second bounce is also negligible probably depends on the material. In a material with low reflectance–small albedo–the second bounce can probably be disregarded. In a material with high reflectance–large albedo–it probably can't. Oren and Nayar are ambivalent about this. In [54], they state, "Our analysis and experimental results suggest that the contribution due to interreflections can be significant and cannot in general be ignored." However, later in the paper they propose a second, simplified model for applications where accuracy is less important than computational efficiency. In what they refer to as the qualitative model, they do ignore the interreflections, as well as the term proportional to C_3 in the direct reflection.

Consider a model in between the accuracy of the full model above and that of the qualitative model. Specifically, the contribution due to C_3 will be retained but the

interreflections will be ignored. In this case, the model becomes

$$I(x, \lambda) = \rho(\lambda) E_0(\lambda) f_1(x) \tag{2.12}$$

which is a separable function of position and wavelength.

2.2.3 The Hapke Lunar Model

A different kind of surface is considered by Hapke in [22]. This surface is comprised of circular tubes with axes parallel to the direction of the incident radiation. Light that is reflected at an angle such that it leaves the tube without hitting an edge propagates to the receiver without attenuation, while other reflections are attenuated exponentially. Interreflective bounces are assumed to be small enough to ignore. This model is meant to imitate the way light is reflected off the surface of the moon, and it is given by

$$I(\phi, \gamma, \beta, \lambda) = E_0(\lambda) a d\omega \frac{2}{3\pi} b(\lambda) \left(\frac{1}{1 + \frac{\cos(\gamma)}{\cos(\gamma + \phi)}} \right) \left(\frac{\sin|\phi| + (\pi - |\phi|)\cos|\phi|}{\pi} \right) B(\phi, g) \tag{2.13}$$

where

- ϕ is the lunar phase angle.
- γ is the lunar longitude of interest (Hapke uses λ, but we have changed this notation to avoid confusion with wavelength, which Hapke does not address specifically).
- β is the lunar latitude of interest.
- E_0 is the incident light.
- λ is the wavelength.
- a is the area of the detector.
- $d\omega$ is the solid angle of the acceptance cone of the detector.

- b is total reflectivity of an object.
- $B(\phi, g)$ is referred to as the retrodirective function, the details of which are not important here, except to note that it is not a function of wavelength.
- $g = \frac{2y}{\tau}$.
- y is the radius of the tubes.
- τ is the mean attenuation length of a beam of light rays in the medium, which can be approximated as being inversely proportional to the product of the cross-section of the objects and the number of objects per unit volume.

Following our prior pattern, we can group all the terms that depend only on spatial variables into one function, $f(x)$, and we find

$$I(x, \lambda) = E_0(\lambda)b(\lambda)f(x) \tag{2.14}$$

which is a separable function of position and wavelength.

2.2.4 The Torrance-Sparrow Model

In [68], Torrance and Sparrow propose a model somewhat similar to that of Oren and Nayer. The crucial difference is that Torrance and Sparrow consider surfaces that are both specularly and diffusely reflective. They also ignore the effects of interreflections. Their model is

$$N(\psi, \theta, \phi, \lambda) = N_s(\psi, \theta, \phi, \lambda) + N_d(\psi, \lambda) \tag{2.15}$$

where

- ψ is the polar angle of the incoming light.
- θ is the polar angle of the reflected light.
- ϕ is the azimuth angle of the reflected light.
- λ is the wavelength.

- N_s is the specularly reflected light.
- N_d is the diffusely reflected light.

Torrance and Sparrow define their coordinate system so that the incoming light has an azimuth angle of 0°.

N_d is given by

$$N_d(\psi, \lambda) = a(\lambda) N_i(\lambda) \cos(\psi) \tag{2.16}$$

where N_i is the incoming light. The diffuse component is not the focus of the paper, and Torrance and Sparrow do not elaborate on what a is, beyond saying it is a constant. However, it seems that they are assuming the diffuse component is Lambertian and a is the product of the diffuse albedo and a proportionality constant that describes how much of the light is reflected diffusely as opposed to specularly. Since Torrance and Sparrow are dealing with a single wavelength, they properly refer to a as being constant, though here it is necessary to note that it is a function of wavelength.

N_s is given by

$$N_s(\psi, \theta, \phi, \lambda) = \frac{f N_i(\lambda) d\omega}{4} F(\psi', \hat{n}(\lambda)) \left[\frac{G(\psi_p, \theta_p)}{\cos(\theta)} \right] P(\alpha) \tag{2.17}$$

where

- f is the area of a facet.
- $d\omega$ is the angle subtended by N_i when viewed from a point on the surface.
- $F(\psi', \hat{n})$ is the Fresnel reflectance of the material.
- ψ' is the angle between the incoming light and the normal of the facet.
- $\hat{n}$ is the complex index of refraction of the material for normal incidence. This is a physical property of the material. It describes how light is reflected, and it is a function of wavelength.

- $G(\psi_p, \theta_p)$ is called the "geometric attenuation factor." It accounts for the effects of masking and shadowing of the facets. The details of how it is computed depend on the precise angles involved and are not important to the current discussion.

- ψ_p and θ_p are projection angles completely determined by ψ, θ, and ϕ.

- $P(\alpha) = be^{-c^2\alpha^2}$ is the probability distribution function of the normals of the facets.

- b and c are constants describing the texture of the surface.

- α is the angle of the facet normal with respect to the global normal of the surface.

Grouping all the terms which are a function only of position into one function, $f(x)$, (2.17) becomes

$$N_s(x, \lambda) = N_i(\lambda)F(\psi', \hat{n}(\lambda))f(x) \tag{2.18}$$

(2.18) and (2.16) can now be used to write (2.15) as

$$N = a(\lambda)N_i(\lambda)\cos(\psi) + N_i(\lambda)F(\psi', \hat{n}(\lambda))f(x) \tag{2.19}$$

Unfortunately, this is not a separable function of wavelength and position. The problem is that there are both diffuse and specular reflections present, and there is no clear relationship between the coefficients of those different reflections. They may, in fact, be very different. For example, a glossy red surface may generally be red due to diffuse reflection of red light, but the spectral highlights will probably appear to reflect all the incoming wavelengths.

Assume, however, that the reflection is entirely specular. (Entirely diffuse reflection was addressed in Section 2.2.1.) Then (2.19) becomes

$$N = N_i(\lambda)F(\psi', \hat{n}(\lambda))f(x) \tag{2.20}$$

The Fresnel coefficient is not a separable function, and so even in this case the function is not separable. However, Torrance and Sparrow illustrate that there are some materials in which the Fresnel coefficient can be reasonably approximated by a constant over a range of ψ'. For such materials, F becomes a function only of wavelength, and (2.20) is separable.

Related physical and empirical image models have been proposed in the early computer graphics literature [3, 7, 58, 73, 75]. These also look at reflected light as the sum of a specular and a diffuse component, with some considerations made for ambient light. Also similar is [24], which, in addition to including a specular and diffuse term, adds a "directional diffuse" term, which depends on the Fresnel coefficient and the properties of the material surface. The details depend on the different assumptions made about object geometries. Analyzing these models proceeds in a way very similar to that done above–if just one of the components is examined on its own, the resulting function may be separable or approximately separable, but when multiple components are summed, the separability is lost.

2.2.5 The Trowbridge-Reitz Model

Another model for specular reflection from a textured surface was proposed by Trowbidge and Reitz in [70]. The surface they considered was one composed of many curved, randomly oriented facets that each reflect specularly. The light propagated to the camera is then given by

$$I_T = \frac{\mathcal{R}(s,n,k)\Phi_I(\lambda)\cos(s)}{\cos(\beta)\cos(\theta)} \left| J\left(\frac{\alpha, z}{\theta, \psi}\right)\right| \frac{1}{A_I}\sum_{i=1}^{N_n} \sigma_{\alpha i}\sigma_{zi} \tag{2.21}$$

where

- I_T is the reflected light.
- $\mathcal{R}(s,n,k)$ is the Fresnel reflectance coefficient.
- s is the angle between the average surface normal of the facets and the observer.

- n and k are respectively the refraction and absorption indices of the material and are functions of wavelength.
- Φ_I is the flux of the entire incident beam and is a function of wavelength.
- β is the polar angle of the incident light.
- θ is the polar angle of the observer.
- $J(\cdot)$ is the Jacobian determinant.
- α is the polar angle of the average surface normal.
- z is the azimuth angle of the average surface normal.
- ψ is the azimuth angle of the observer.
- A_I is the cross-sectional area of the incident beam.
- N_n is the number of facets that contribute to the irradiance at a particular point and is related to A_I.
- $\sigma_{\alpha i}$ and σ_{zi} are the radii of curvature of facet i.

It is easy to see that the only things that depends on wavelength in (2.21) are the incident flux and the Fresnel coefficient $\mathcal{R}$. Unfortunately, as discussed above, the Fresnel coefficient is not, in general, a separable function of position and wavelength. However, if the material is such that it can be approximated as a constant over a range of angles, then it is a function only of wavelength within that interval, and (2.21) can be approximated by a separable function.

2.2.6 The Kajiya Model

A very general model for specular reflections off of dielectric anisotropic textures was developed from Maxwell's equations by Kajiya [31]. It is given by

$$\psi(x,\lambda) = -\frac{i}{4\pi|r_0(x)|}e^{i|k(\lambda)||r_0(x)|}\int_S n(s)[R(k_1(\lambda)-k_2(\lambda),x)-(k_1(\lambda)+k_2(\lambda))]e^{i(k_1(\lambda)-k_2(\lambda))s}dS \tag{2.22}$$

where

- ψ is the electric field at the receiver.
- x is the spatial variable.
- λ is the wavelength.
- r_0 is the vector from x to the center of the region of scattering.
- k_1 and k_2 are the wave vectors of the incident and reflected waves, respectively, and $|k_1| = |k_2| = |k|$.
- S is the surface of the object being imaged.
- n is the surface normal.
- R is the Fresnel reflection coefficient.
- $s = r_0 - r$ where r is the vector between x and the point being integrated.

(Note that this function has been slightly modified from the form in [31] to make the spatial and spectral dependencies explicit.)

This is not a separable function of position and wavelength. Even if, as with the Torrance-Sparrow model, we assume the Fresnel coefficient was constant as a function of position, we would still have trouble with the two exponential terms, so, in general, specular reflections off arbitrary anisotropic textures cannot be modeled as separable functions of position and wavelength.

2.2.7 Conclusion

In this section, we have looked at several different physically based image models, some of which were separable, some of which were not, and some of which could be approximated by separable functions. Based on the models we have examined, it seems that diffuse reflections without interreflective bounces are well-represented by separable functions. Specular reflections may be approximated by separable functions

in certain situations. Combined specular and diffuse reflections cannot be represented by separable functions. The limitations of the separability assumption will be explored further in Chapter 5.

2.3 Retinex Theory

The scene irradiance of an image is often written as the product of the illumination and the reflectance, both of which are allowed to be spatially varying, namely

$$I(x, \lambda) = s(x, \lambda)r(x, \lambda) \tag{2.23}$$

where I is the scene irradiance, s is the illumination, and r is the reflectance [26]. This formulation is sufficiently general that it captures some of the ideas presented above, particularly those of the Lambertian model.

The motivation for writing an image this way comes from the observation that the human visual system is remarkably insensitive to changes in the illumination of a scene. The perception of color depends much more on reflectance than on illumination [36]. It is therefore natural to ask how the brain separates these two components, and to try to mimic this process artificially. The field of work focusing on this problem is called retinex theory.

Obviously, separating reflectance and illumination is an ill-posed problem. Solutions tend to rely on an assumption that the illumination is smooth and then to focus on the image gradients (or gradients of the logarithm of the image). A discussion of the details is outside the current scope, but the interested reader is referred to [47] and the references therein.

Retinex work is interesting here because it illustrates two aspects of the physical models previously discussed. First, the parameters of these models are very difficult to compute from just an image. Like separating illumination from reflectance, the problem of parameter estimation is ill-posed, and any method to do it would have to make additional assumptions about the scene which are not required by the underlying

model itself. Second, if these parameters could be computed, it would give us insight into the physical nature of the underlying scene. This could be useful, for example in shape-from-shading [19], which uses physical models of scene formation to generate a 3D model of the objects in a 2D image.

2.4 Image Priors

Image models are of interest not just because of the physical insights they can give, but also because of their usefulness in what can be broadly referred to as the image reconstruction problem. The goal of image reconstruction is to restore a true or natural-looking image from partial information about the scene. Examples include denoising, where the image data is available but has been corrupted by noise; deblurring, where the image data is available but has been filtered with a function that may or may not be known; superresolution, where we want to use a low-resolution image of a scene to create a high-resolution image of that scene; and demosiacing, where we want to use data from one color channel to help fill in missing data from other color channels.

These problems are all ill-posed, and therefore any solution to them depends on having knowledge of the characteristics of the images. This knowledge comes from image models, or "priors," that can be either physically or empirically based. Far too many such priors have been proposed to give a detailed review of all of them here, and a general discussion will suffice for our purposes.

Image priors can be roughly categorized into two groups: spatial and spectral priors. As the name implies, spatial priors articulate the spatial features expected in images and apply to black and white images or to the channels of a color image separately. One example of a spatial prior is image self-similarity, or the idea that an image will have many patches that resemble each other. This has been used in state of the art denoising systems [4, 9]. There is no strict physical justification for this observation, and one could imagine a highly intricate scene where it did not hold, but it has proven to be a good descriptor of most natural images.

Other empirical spatial priors can be derived from observed image statistics. For example, the gradients of an image tend to have a heavy-tailed distribution [17], which can be used in deblurring. It has also been found that the power spectrum of natural images tends to fall with frequency according to a power law, which may be due to the scale invariance of natural images (that is, the kinds of structures that appear in images may appear at any size as a photographer moves closer to or farther away from them) [61, 62]. These observations are made for either the color channels individually or the luminance channel, implying that they hold regardless of wavelength.

Spectral priors, on the other hand, articulate the relationships between the color channels of an image. For example, one simple chrominance prior is that the ratio of two colors within an object is a constant. This can be derived from the Lambertian model and used for demosaicing [33]. Other spectral priors include the assumption that the chrominance channels of an image tend to be low frequency signals and that the pixels of an image tend to form lines in RGB space [51].

Here, we are more interested in spectral than spatial priors. In the next chapter, we will derive an image model that can be used as a spectral prior. The relationship between our model and the spectral priors mentioned above will be explored in detail in Chapter 6. The use of spectral and spatial priors is not mutually exclusive, and most color image reconstruction algorithms will want to combine the two. An example of this will be shown in Chapter 7.

Chapter 3

Derivation

Consider a generic separable model of an image. That is,

$$I(x, \lambda) = f(x)g(\lambda) \tag{3.1}$$

Where $I(x, \lambda)$ is the value measured by the camera at a particular location and wavelength, $f(x)$ is an arbitrary function of position, and $g(\lambda)$ is an arbitrary function of wavelength.

When a digital camera measures the incoming light from a scene, it multiplies it by a color channel specific spectral response function $s_i(\lambda)$ and integrates over wavelength. So, for the red channel, the measured pixel value $R(x)$ would be

$$R(x) = \int s_R(\lambda) f(x) g(\lambda) d\lambda \tag{3.2}$$

$$= f(x) \int s_R(\lambda) g(\lambda) d\lambda \tag{3.3}$$

For convenience, let $g_i = \int s_i(\lambda) g(\lambda) d\lambda$. Then the values measured by the camera in each color channel can be written as

$$R(x) = f(x) g_R \tag{3.4}$$

$$G(x) = f(x) g_G \tag{3.5}$$

$$B(x) = f(x) g_B \tag{3.6}$$

Consider $R_{avg}(x)$, the local average of the pixels in a window W around a particular location x in the red channel. This is given by

$$R_{avg}(x) = \frac{1}{|W|} \sum_{k \in W} f(x+k) g_R \tag{3.7}$$

$$= \frac{g_R}{|W|} \sum_{k \in W} f(x+k) \tag{3.8}$$

Dividing the pixel under consideration by this local average produces a "residual" $q_R(x)$ given by

$$q_R(x) = \frac{R(x)}{R_{avg}(x)} \tag{3.9}$$

$$= \frac{f(x) g_R}{\frac{g_R}{|W|} \sum_{k \in W} f(x+k)} \tag{3.10}$$

$$= \frac{f(x)}{\frac{1}{|W|} \sum_{k \in W} f(x+k)} \tag{3.11}$$

Notice that $q_R(x)$ does not depend on g_R, and so if this procedure was repeated in the green or blue channels, the result would be

$$q_R(x) = q_G(x) = q_B(x) = q(x) \tag{3.12}$$

That is, the value at a given pixel is the product of the color dependent local average and the color independent residual. Specifically, this means that the color channels can be written as

$$R(x) = R_{avg}(x) q(x) \tag{3.13}$$

$$G(x) = G_{avg}(x) q(x) \tag{3.14}$$

$$B(x) = B_{avg}(x) q(x) \tag{3.15}$$

Equations (3.13)-(3.15) will be referred to as the multiplicative model. Notice that this implies a proportional relationship between the color channels.

As will be discussed in more detail in the next chapter, it can also be useful

to consider logarithmic domain processing because of the way cameras work. The notation $h'(x)$ will be used to indicate the logarithmic domain. In this case, the model becomes additive. Specifically, (3.4)-(3.6) become

$$\log(R(x)) = \log(f(x)) + \log(g_R) \tag{3.16}$$

$$\log(G(x)) = \log(f(x)) + \log(g_G) \tag{3.17}$$

$$\log(B(x)) = \log(f(x)) + \log(g_B) \tag{3.18}$$

For notational simplicity, we define $f'(x) = \log(f(x))$, $g'_R = \log(g_R)$, and so forth, so we have

$$R'(x) = f'(x) + g'_R \tag{3.19}$$

$$G'(x) = f'(x) + g'_G \tag{3.20}$$

$$B'(x) = f'(x) + g'_B \tag{3.21}$$

Then, computing the local average:

$$R'_{avg}(x) = \frac{1}{|W|}\left(\sum_{k\in W} f'(x+k) + g'_R\right) \tag{3.22}$$

$$= g'_R + \frac{1}{|W|}\sum_{k\in W} f'(x+k) \tag{3.23}$$

Subtracting this from the pixel under consideration, the residual is

$$q'_R(x) = f'(x) + g'_R - \left(g'_R + \frac{1}{|W|}\sum_{k\in W} f'(x+k)\right) \tag{3.24}$$

$$= f'(x) - \frac{1}{|W|}\sum_{k\in W} f'(x+k) \tag{3.25}$$

and again $q'_R(x)$ is not a function of wavelength, so it can simply be referred to as

$q'(x)$ and the color channels can be written as

$$R'(x) = R'_{avg}(x) + q'(x) \tag{3.26}$$

$$G'(x) = G'_{avg}(x) + q'(x) \tag{3.27}$$

$$B'(x) = B'_{avg}(x) + q'(x) \tag{3.28}$$

Equations (3.26)-(3.28) will be referred to as the additive model.

It is worth pointing out that, unlike retinex work and the physical models we started with, the parameters of this model do not necessarily have any particular physical meaning in themselves. The goal of this derivation is to use physical principles to derive a model that articulates a relationship between color channels that can be easily computed from images produced by a camera. It is not to provide a way to use an image captured by a camera to compute physical properties of an underlying scene. Therefore, while both retinex theory and our model rest on similar models of image formation, the critical assumptions of the properties of the scene are different–retinex work assumes that the illumination in the scene is slowly varying, while here it is assumed that it is a separable function of position and wavelength. Furthermore, retinex work focuses on computing derivatives to locate edges. At this point, we have assumed there are no edges (an assumption that will be addressed later in more detail).

It is possible that our model could be strengthened by including additional assumptions about the spectral behavior of the lightsource and reflectance of the object. For example, there is empirical evidence that both daylight [15, 30] and natural objects [6, 27, 57] have spectral behavior that can be well-described by the sum of only a few basis functions, and this becomes even more pronounced when the spectral responses of the human visual system (or, presumably, a camera) are included in the analysis [13, 43]. Such results could potentially be used to derive a relationship between the local averages, or to put a bound on the modeling error when the underlying physical model that is appropriate for a given scene is not exactly a separable function. We leave this as an area of future work and will continue without any

assumptions on the structure of $g(\lambda)$.

We also note here the difference between our model and homomorphic filtering [52]. The idea of homomorphic filtering is to transform an image so that components of interest, for example, signal and noise or reflectance and luminance, have an additive relationship to make processing easier. When these components have a multiplicative relationship initially, this means taking the logarithm of the image and doing the processing in the log domain (e.g. [72]). While we also are interested in the multiplicative relationship of the components involved in image formation, we compute the log only to mimic the camera's behavior–the local average and residual are straightforward to compute and process regardless of whether we are using the additive or the multiplicative model, and, as said before, they have no physical meaning. There is no reason to prefer one over the other except that the datasets we will be using happen to already be in the log domain.

Chapter 4

Verification

In this chapter, the experimental verification of the model will be presented. First, the properties of the image datasets used for testing will be described. Then, the selection of the window W and the computation of the residual will be discussed. Finally, experimental results will be shown.

4.1 Image Datasets

To verify that this model does work in practice, it was tested on three sets of images. The first set was comprised 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 demosaicing [40]. Therefore, if the model works with these images, it is not simply an indication that it correctly mimics the demosaicing process. The database was designed to include images with a wide variety of characteristics, containing both somewhat stylized images of textures and more natural images of scenes. These images were 2640x1760 pixels, with 16 bits stored per pixel in each color channel. All pictures were captured with the ISO set to 100, the lowest available sensitivity, to minimize sensor noise, and an aperture of F/9.0. The exposure time varied from 1/800-2/5s, depending on the lighting of the scene.

The second dataset was comprised of 29 images taken with a Pentax K-3 II camera in pixel-shift resolution mode. In this operating mode, the camera takes four pictures

in quick succession, shifting the Bayer array by one pixel for each shot. It then combines the images into one, with the red, green, and blue channels actually measured at each pixel [64]. Again, there is no demosaicing. Since the use of pixel-shift resolution mode requires four images that are exactly aligned, it is necessary to mount the camera on a tripod and image only scenes that are perfectly still. Because of this, the dataset focuses on images of textures rather than natural scenes. These images were 6016x4000 pixels, with 8 bits stored per pixel in each color channel. Again, the ISO was set to 100 and the aperture was F/9.0. The exposure time ranged from 1/250-5s. (The fact that the camera was mounted on a tripod enabled these relatively long exposure times without introducing blurring.)

The third dataset was the classic Kodak dataset [1]. There are 24 images in this dataset, each 512x768 pixels, with 8 bits stored per pixel in each color channel. These images were taken with a conventional camera, and so presumably there was demosaicing. Therefore, demonstrating that the model works with this dataset only demonstrates that it accurately mimics the camera's demosaicing process. However, taken with the other results that will be presented, the model's success with this dataset is further evidence that the model does work generally, and is not limited to specific cameras.

Thumbnail versions of the images in these three datasets are given in Appendix A.

Since the model describes scene irradiance, the camera calibration method proposed in [14] was used to determine how to map the values reported by the cameras 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 pixel value, to $y(z)$, a function which is equal to the natural logarithm of the scene irradiance up to an additive constant term that depends on the exposure value. This camera calibration function accounts for not only potential non-linearities in the sensor response, but also for any low-level processing the camera may do, such as gamma correction, before reporting the "raw" image values. For a more detailed discussion, see [14].

Results for the Sigma DP1 are shown in Fig. 4-1, and results for the Pentax K-3

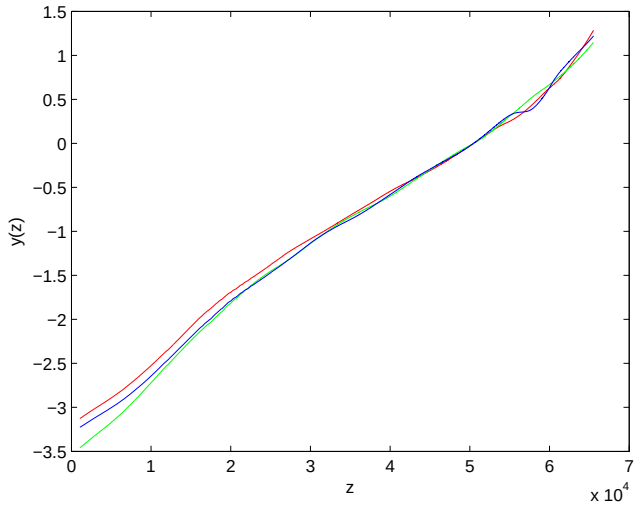


Figure 4-1: Camera calibration functions for the Sigma DP1. The y-axis is the log irradiance (up to an additive constant) and the x-axis is the reported pixel value.

II are shown in Fig. 4-2. The colored lines are the calibration functions for the red, green, and blue channels. The same smoothing parameter was used for each color channel, and it was chosen so that it was just high enough to ensure monotonicity of the resulting calibration functions. The smoothing parameters found in this way were different for the different cameras.

From Fig. 4-1, it can be seen that the values reported by the Sigma DP1 are very nearly proportional to $y(z)$. Note that the arbitrary constant relating $y(z)$ to the log irradiance value is immaterial, as it is uniform throughout the image and will simply be absorbed into the local average. The constant of proportionality will also play only a limited role since it is nearly the same in the three color channels, though differences in slope could cause some modeling error. However, the key point is that the image values are linearly related to the log irradiance, and the use of the additive model of (3.26)-(3.28) in the experiments involving the Sigma dataset is well justified.

From Fig. 4-2, it can be seen that the values repoted by the Pentax K-3 II are nearly piecewise linear functions of the log irradiance. Again, this justifies the use of the additive model in experiments involving the Pentax dataset. The "knee" in the curve at about $z = 40$ and the other slight nonlinearities, as well as the differences in slope, may cause some errors, but since there are relatively large regions of almost

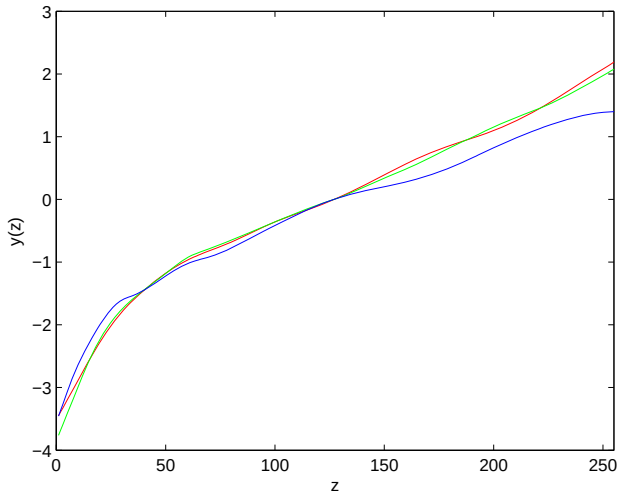


Figure 4-2: Camera calibration functions for the Pentax K-3 II. The y-axis is the log irradiance (up to an additive constant) and the x-axis is the reported pixel value.

linear behavior and almost equal slopes, the additive model can still be reasonably expected to perform better than the multiplicative model.

Finally, it was impossible to perform camera calibration on the Kodak dataset without access to the camera those pictures were taken with. Both the additive and multiplicative model will be tested on this dataset and performance will be compared.

The irradiance mappings described here will be used only to determine whether the additive or the multiplicative model should be applied to the datasets. In the following results, the original images will be modeled, not the corresponding irradiance maps. There are three motivating factors for this choice. First, the camera calibration function of [14] is an imprecise instrument. Results can be altered substantially by changing the user-selected smoothing parameter. Additionally, due to camera glare and other complicating factors, irradiance maps produced in this way cannot really be considered the ground truth of the scene irradiances [45]. Second, for practical applications, the camera calibration function may not be available, and the model will be of limited usefulness if it cannot be applied to images directly. Third, as will be discussed in more detail later, it is hard to define a good image quality metric, and there really is no substitute for human evaluation. However, it is unclear how to display an irradiance map properly and so evaluating the quality of modeled irradiance

maps is even more ambiguous than evaluating the quality of modeled images.

For these reasons, the camera calibration functions are used only to determine if the images have a more linear or more logarithmic relationship to the scene irradiances, not to compute irradiance maps. The original images will be considered the "ground truth" of the scene being modeled.

4.2 Averaging Window

For W in (3.7) and (3.22), we will start with a 7x7 pixel square window centered at the pixel under consideration. Since the assumption that there is only one material in the window will be violated at material boundaries when W is a simple square, a threshold is also included. Specifically, for each 7x7 pixel square S around the pixel under consideration, x_0, the maximum absolute difference between the current pixel and the pixels in the window in each color channel is computed as follows:

$$D_R = \max_{x \in S}(|R'(x_0) - R'(x)|) \tag{4.1}$$

$$D_G = \max_{x \in S}(|G'(x_0) - G'(x)|) \tag{4.2}$$

$$D_B = \max_{x \in S}(|B'(x_0) - B'(x)|) \tag{4.3}$$

The color channel A with the largest difference is then selected ($A = \arg\max_i D_i$), and a threshold is used on the intensity difference between the pixels in the window and the pixel under consideration to determine the pixels within S that will be used. That is, W is defined such that

$$W = \{x | x \in S \text{ and } |A(x) - A(x_0)| < T\} \tag{4.4}$$

The set W is used as the set of pixels over which the local average is computed in all three color channels, the reasoning being that an edge detected in one color channel will also be an edge in the other color channels. The max operator is used in choosing A so that the edges can be found in the color channel where they are the

most prominent.

In practice, the choice of the threshold T will depend on the application for which the model is being used. For example, if noisy images were being modeled, T would depend on the noise level. Since it is only the validity of the model being considered here, without reference to any particular application, the choice of threshold becomes arbitrary. However, the following considerations will be taken into account. 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 will degenerate to $W = \{x_0\}$. 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.

In the Sigma dataset, the dynamic range of the images is 2^{16}, and a threshold of 10,000 was found to give good results empirically, detecting many edges without collapsing to degeneracy. For the sake of consistency, a threshold of 40 will be used for the Pentax and Kodak datasets, which both have a dynamic range of 2^8, so for all the datasets the threshold is about 15% of the dynamic range.

4.3 Computing the Residual

To test the additive model, an estimate of the residual was computed by solving (3.26)-(3.28) separately for $q'(x)$. If the model was perfect, the $q'(x)$ estimated from the three channels should be the same. In practice, however, they are not identical. To get a single value, they were averaged, as shown in (4.5).

$$\hat{q}'(x) = \frac{1}{3}(R'(x) - R'_{avg}(x) + G'(x) - G'_{avg}(x) + B'(x) - B'_{avg}(x)) \tag{4.5}$$

The function $\hat{q}'(x)$ can then be used to estimate the color channels as

$$\hat{R}'(x) = R'_{avg}(x) + \hat{q}'(x) \tag{4.6}$$

and so forth.

Similarly, to test the multiplicative model, the residual $q(x)$ can be estimated by computing the residuals in the three channels and combining them into a single value by computing

$$\hat{q}(x) = \frac{1}{R_{avg}^2(x) + G_{avg}^2(x) + B_{avg}^2(x)} \left(R_{avg}^2(x) \frac{R(x)}{R_{avg}(x)} + G_{avg}^2(x) \frac{G(x)}{G_{avg}(x)} + B_{avg}^2(x) \frac{B(x)}{B_{avg}(x)} \right) \tag{4.7}$$

This is simply the average of the three residuals computed with weights that will minimize the mean squared error of the modeled image, which is computed as

$$\hat{R}(x) = R_{avg}(x)\hat{q}(x) \tag{4.8}$$

and so forth.

Notice that a naive implementation of the modeling algorithm is quite computationally intensive. Each output pixel requires the computation of a local average, which involves about $7 \times 7 = 49$ additions in all three color channels. The additive model then requires five additions for the computation of the residual, and one more in each color channel to compute the output value. The multiplicative model requires three squarings, three divisions, and four additions to compute the residual, and a multiplication to compute the final output value in each color channel. Note, however, that each pixel is computed independently, so the algorithm is highly parallelizable.

4.4 Experimental Results

We want to check that our modeling does not adversely affect the original image, and we do this by computing the the peak signal-to-noise ratio (PSNR) between the modeled image and the original. PSNR, computed in dB, is given by

$$PSNR = 10 \log_{10} \left(\frac{MAX^2}{\frac{1}{N} \sum (orig - modeled)^2} \right) \tag{4.9}$$

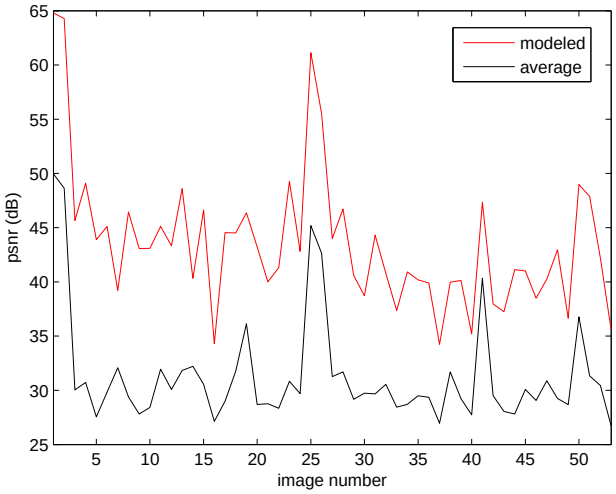


Figure 4-3: Results of modeling the images in the Sigma dataset.

where MAX is the maximum possible value of the original image ($2^{16}-1$ or 2^8-1, depending on the data set), $N = 3\times$ the number of pixels in the image, *orig* is the original image, *modeled* is the modeled image, and the sum is taken over all the pixels and all the color channels. The closer the modeled image is to the original, the larger the PSNR will be. PSNR is, of course, only one possible measure of image quality, but it is a standard metric in the denoising literature (e.g. [9, 10, 18, 71, 76]). Visual examples will also be provided.

The results of using the additive model to reconstruct the images in the Sigma dataset are shown in Fig. 4-3. The x-axis is the number of the image in the dataset. The ordering is arbitrary; interpolation between non-integer values is not meaningful and is done only for visual clarity. The y-axis gives the PSNR of the modeled images in dB. The red line is the error of the modeled images. For comparison, the error of the local averages is shown by the black line. Notice that the local average has a considerably lower PSNR than the modeled image in all cases. This demonstrates both that the local average was non-degenerate and that the residual correctly added back in part of the high frequency information.

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. To give a sense of what these numbers mean for subjective visual quality, a few examples are

(a) Original (b) Modeled

Figure 4-4: Results of reconstructing Sigma image 1 with the proposed model.

(a) Original (b) Modeled

Figure 4-5: Results of reconstructing Sigma image 16 with the proposed model.

provided in Figs. 4-4 through 4-6. Fig. 4-4, which shows image 1 from the database, is a fairly flat image with no material edges and little texture. This is a case of very good performance of the model. Fig. 4-5, which shows image 16, on the other hand, is highly textured with many 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. Finally, Fig. 4-6, which shows image 47, 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 closeup is shown in Fig. 4-7.

In all cases, the modeled images look very similar to the original images. This demonstrates that the high PSNRs of the modeled images do in fact indicate that

(a) Original (b) Modeled

Figure 4-6: Results of reconstructing Sigma image 47 with the proposed model.

(a) Original (b) Modeled

Figure 4-7: Closeup of Sigma image 47 and the result of modeling.

the model works well.

Results of applying the additive model to the Pentax dataset were similar, and are shown in Fig. 4-8. Again, the x-axis is the image number, the y-axis is the PSNR in dB, the red line shows the result of modeling, and the black line is the local average itself. As before, the model performs quite well in the PSNR sense, with the average error being about 45dB.

Fragments of example images are shown in Figs. 4-9 through 4-11. Fig. 4-9, which shows image 3 from the dataset, is a case of particularly good model performance. Fig. 4-10, which shows image 26, is a case of about average performance, and Fig. 4-11, which shows image 13, is a case of particularly poor performance. Again, while edge artifacts are visible in Fig. 4-11, the modeled images overall look quite similar

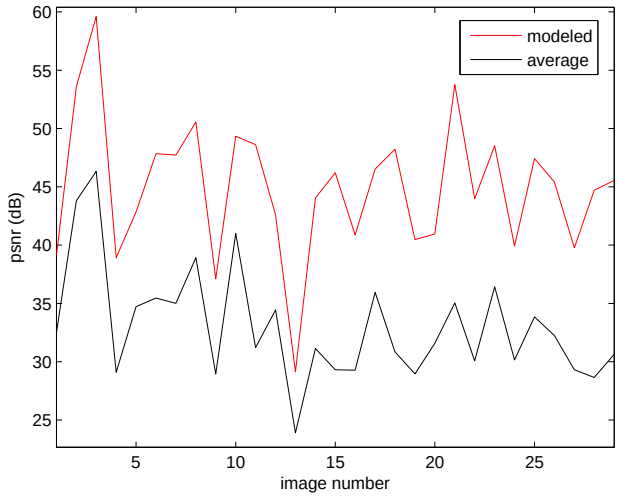
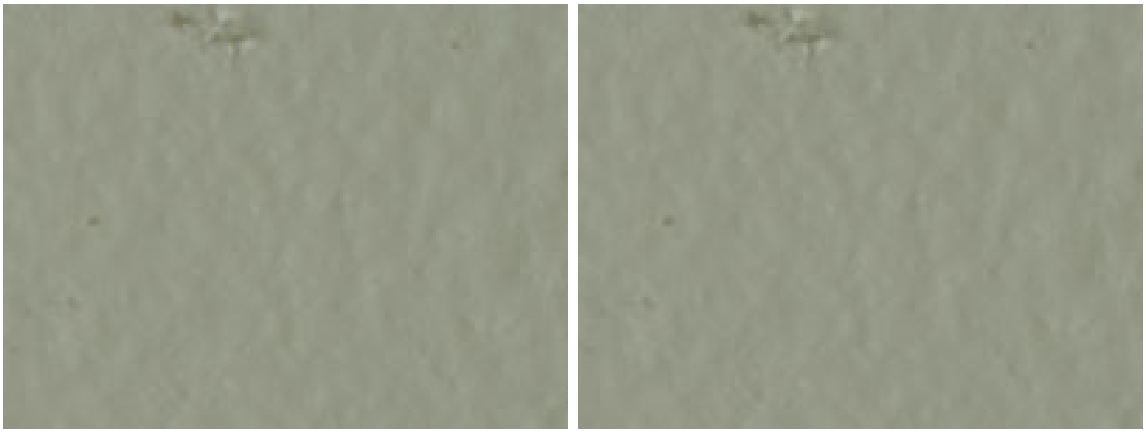


Figure 4-8: Results of modeling the images in the Pentax dataset.

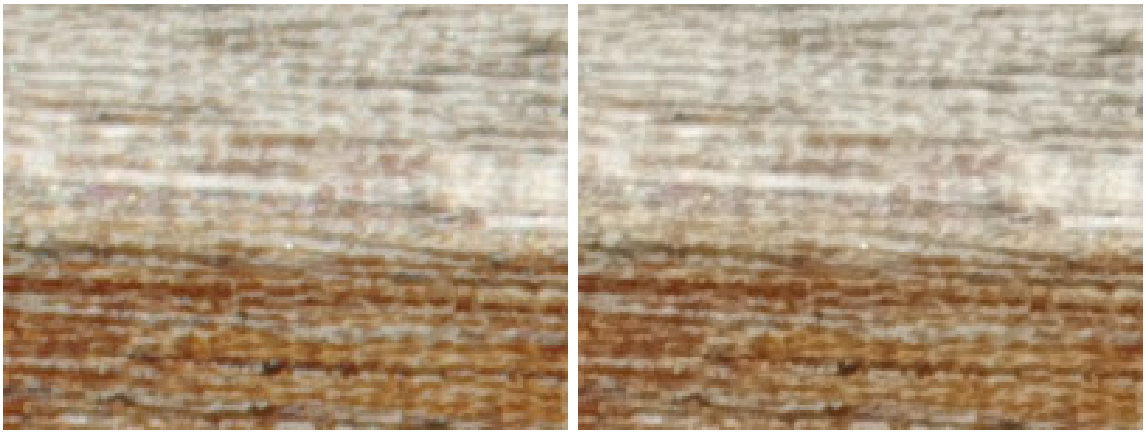


(a) Original (b) Modeled

Figure 4-9: Results of reconstructing Pentax image 3 with the proposed model.

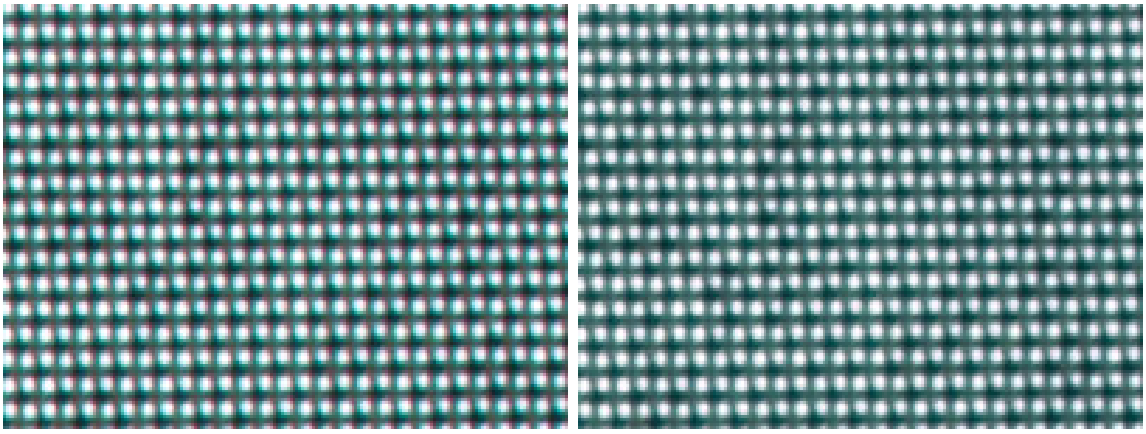
to the originals.

Finally, the results of modeling the Kodak dataset are shown in Fig. 4-12. Once again, the x-axis is the number of the image in the dataset and the y-axis is the PSNR in dB. The black line shows the error in the local average, and the red and blue lines show the error in the additive and multiplicative models, respectively. The average error of the results produced by the additive model was about 43dB, and that of the multiplicative model was about 42dB. Based on these results, it is reasonable to assume the Kodak camera was similar to the other two cameras in that the values it reported were related to the logarithm of the scene irradiances.



(a) Original (b) Modeled

Figure 4-10: Results of reconstructing Pentax image 26 with the proposed model.



(a) Original (b) Modeled

Figure 4-11: Results of reconstructing Pentax image 13 with the proposed model.

An example image is shown in Fig. 4-13, which shows image 11 from the dataset. This is a case of average performance, and again the modeled image is very similar to the original.

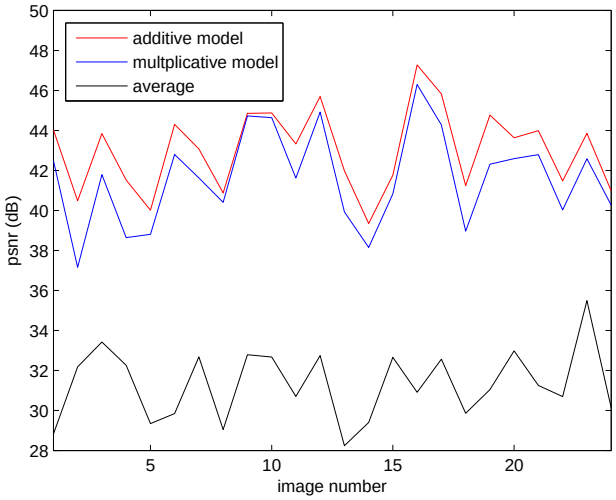


Figure 4-12: Results of modeling the images in the Kodak dataset.

(a) Original (b) Modeled

Figure 4-13: Results of reconstructing Kodak image 11 with the proposed additive model.

Chapter 5

Limitations

In the previous chapter, it was established that the model does generally work. In this chapter, some failure cases will be addressed. Specifically, failures due to undetected edges, saturation, nonlinear camera calibration functions, interreflective bounces, combined specular and diffuse reflection, chromatic aberrations, and bright regions will be discussed. Where possible, examples illustrating these problems will be presented, as will suggestions for how they can be mitigated in practice.

5.1 Undetected Edges

Undetected edges are generally the largest source of error in the modeled images. In the derivation of the model, it was assumed that the scene irradiance was a separable function of position and wavelength. This assumption is only reasonable if there is a single material in the area under consideration. Otherwise, the reflectance properties of the scene (which are a function of wavelength) will change from one position to the next, violating the assumption.

Undetected edges lead to color bleeding artifacts, as illustrated in Fig. 5-1, which shows a close-up of Sigma image 47 and the result of modeling with and without edge detection. Such artifacts can also be seen in Fig. 4-11, where the colors in the modeled image have been visibly washed out due to the many small edges present in the image.

(a) Original (b) Modeled with edge detection. (c) Modeled without edge detection.

Figure 5-1: Closeup of Sigma image 47 and the results of modeling with and without edge detection.

Edge detection in color images is a complicated problem, and the best way to deal with it will depend on the image characteristics. A simple method was used in the previous chapter, and it seems to work reasonably well in most cases. A more sophisticated system could be implemented if the application required it.

5.2 Saturation

When pixel values are saturated, the model could potentially fail. If the saturated pixels are within the window used to compute the local average, then the local average will be lower than it would be without saturation, and so the residual in that color channel will be erroneously large. On the other hand, if the pixel under consideration is saturated, then the residual will be erroneously small.

This problem was not significant–only 1.0% of the pixels in the Sigma dataset, 6.5% of the pixels in the Pentax dataset, and 3.9% of the pixels in the Kodak dataset were saturated or were within a window that contained a saturated pixel. Furthermore, saturation does not always adversely affect the performance of the model–it tends to be the result of a bright light, and so all the colors are saturated together over a region. In such an area, the colors are equal and the model will describe them perfectly.

If it is necessary to deal with saturation, one straightforward way of doing so is to simply detect and ignore saturated pixels and those near them when modeling.

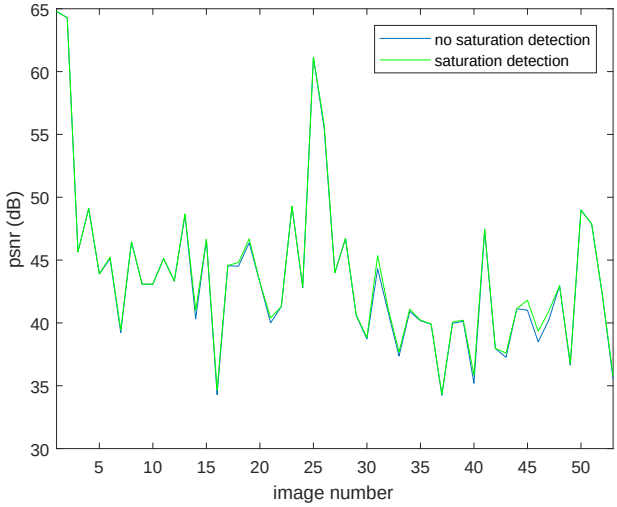


Figure 5-2: Performance improvement due to saturation detection in the Sigma dataset.

Adding this to the verification tests done previously can only improve the results, since it is clean images that are being modeled. A comparison of using and not using saturation detection for the three datasets is shown in Figs. 5-2 through 5-4.

Only in the Kodak dataset is there any clear improvements of the PSNRs of the modeled images when saturation detection is included, and even there, the advantage is modest–the average improvement is about .64dB. The most dramatic improvement occurs in image 20, which is shown in Fig. 5-5a. The result of modeling without saturation detection is shown in Fig. 5-5b, and the result of modeling with saturation detection is shown in Fig. 5-5c. The differences between the two results are mainly in the texture of the sky, and are not visually distinguishable.

These results indicate that saturation is not a large source of error when applying the model, and can probably be ignored. If a particular situation arises in which it must be dealt with, it is easy to detect and can be accounted for in a way appropriate to the situation.

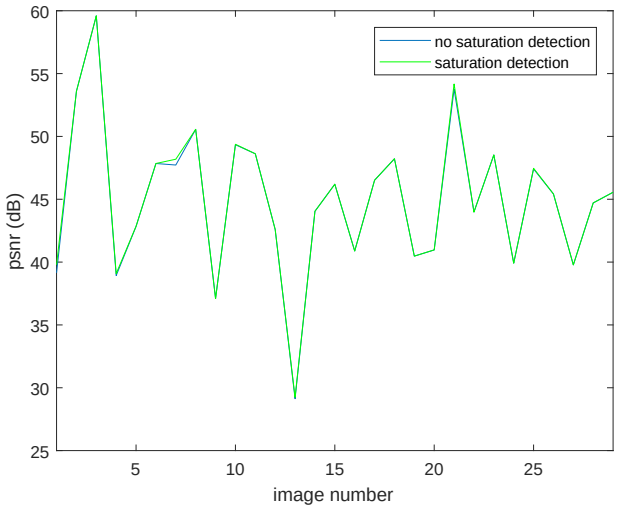


Figure 5-3: Performance improvement due to saturation detection in the Pentax dataset.

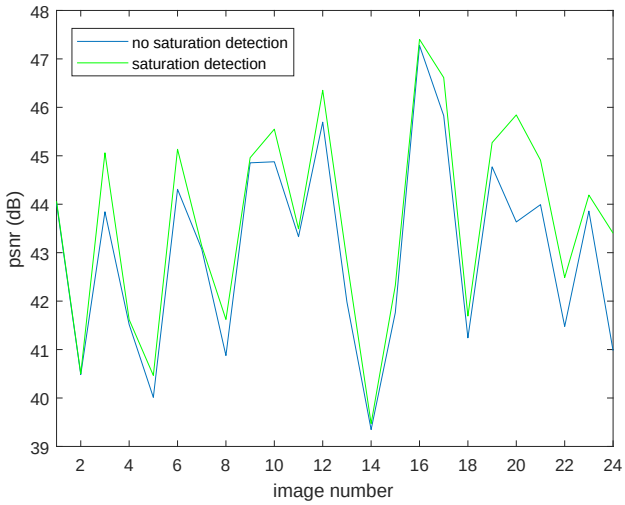


Figure 5-4: Performance improvement due to saturation detection in the Kodak dataset.

5.3 Nonlinear Camera Calibration Functions

The model was derived for scene irradiances, and the pixel values of a camera do not necessarily have a straightforward relationship to these values. The camera calibration functions presented in Fig. 4-1 and Fig. 4-2 are good indicators that the relationship is approximately logarithmic, but only approximately. This approximation is another potential source of error in the modeled images.

(a) Original

(b) Modeled without saturation detection

(c) Modeled with saturation detection

Figure 5-5: Kodak image 20 and the results of modeling with and without saturation detection.

To determine the severity of these kinds of errors, it is useful to focus on the "knee" in the Pentax calibration curve at around $z = 40$, which is the most dramatic nonlinearity present. The performance penalty due to this nonlinearity can be evaluated by breaking the images in the Pentax dataset into blocks of size 16x16 pixels and modeling each block independently. The blocks where there was a color channel whose average value fell below the knee in the calibration curve and a color channel whose average value fell above it were noted. The distributions of the PSNRs are shown in Fig. 5-6.

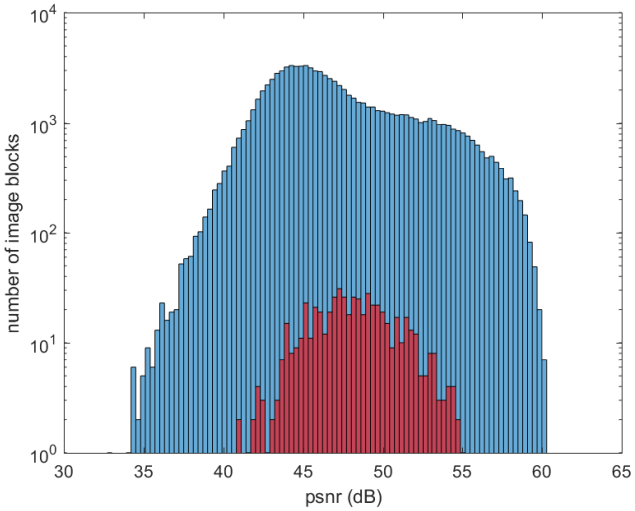


Figure 5-6: PSNRs of the blocks in the Pentax dataset. The blue bars indicate the total number of blocks, and the red bars indicate the number of blocks that contain color channels with averages on both sides of the knee in the calibration curve. Notice that the number of blocks is plotted on a logarithmic scale.

The distribution of the error in the blocks that contain color channels with averages on either side of the knee is fairly similar to the distribution of errors in all the blocks. This indicates that nonlinearities in the camera calibration function are not a dominant source of modeling error. However, it would be possible to correct these errors by modeling the irradiance map instead of the image if a precise camera calibration function were known.

5.4 Interreflective Bounces

Recall the full Oren-Nayer model of (2.11), repeated below for reference:

$$I(x,\lambda) = \rho(\lambda)E_0(\lambda)f_1(x) + \rho(\lambda)^2 E_0(\lambda)f_2(x) \tag{5.1}$$

The second term is due to the interreflective bounce. Consider what happens when such an image is modeled. As usual, assume the data is in the logarithmic domain. For simplicity, ignore the camera response function.

The local average in, for example, the red channel is given by

$$\begin{aligned}
R'_{avg}(x) &= \frac{1}{|W|}\sum_{k\in W}\log(\rho(\lambda_R)E_0(\lambda_R)f_1(x+k) + \rho(\lambda_R)^2E_0(\lambda_R)f_2(x+k)) && (5.2)\\
&= \frac{1}{|W|}\sum_{k\in W}\log(\rho(\lambda_R)E_0(\lambda_R)(f_1(x+k) + \rho(\lambda_R)f_2(x+k))) && (5.3)\\
&= \frac{1}{|W|}\sum_{k\in W}\log(\rho(\lambda_R)E_0(\lambda_R)) + \log(f_1(x+k) + \rho(\lambda_R)f_2(x+k)) && (5.4)\\
&= \log(\rho(\lambda_R)E_0(\lambda_R)) + \frac{1}{|W|}\sum_{k\in W}\log(f_1(x+k) + \rho(\lambda_R)f_2(x+k)) && (5.5)
\end{aligned}$$

and so the residual is given by

$$\begin{aligned}
q'_R(x) &= R'(x) - R'_{avg}(x) && (5.6)\\
&= \log(f_1(x) + \rho(\lambda_R)f_2(x)) - \frac{1}{|W|}\sum_{k\in W}\log(f_1(x+k) + \rho(\lambda_R)f_2(x+k)) && (5.7)
\end{aligned}$$

Since the albedo appears in the residual, it is reasonable to expect larger errors

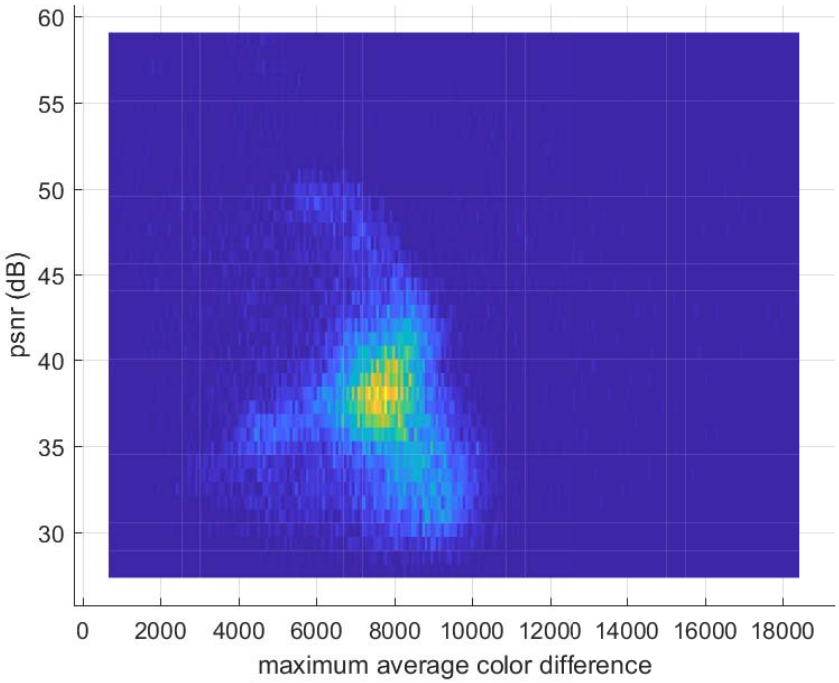


Figure 5-7: PSNR as a function of maximum average color difference in the Sigma dataset.

when the differences between the albedos of the different color channels are larger. As stated previously, the model does not provide a way to measure the albedo. However, we can look at the average value in the color channels of a block as a proxy. Assuming the light is white, a large difference in average value should indicate a large difference in albedo.

The images in the three datasets were segmented into 16x16 pixel blocks, which were modeled independently. Figs. 5-7 through 5-9 are histograms displaying the PSNR of the modeled block as a function of the maximum average color difference within the block, with brighter colors indicating more blocks of a particular maximum average color distance and PSNR. The thin vertical and horizontal lines visible are an artifact of how the plots were rendered and are not meaningful.

These plots do not show any clear relationship between maximum average difference in the color channels and model performance. This indicates that the interreflective bounce is probably not a dominant source of model error.

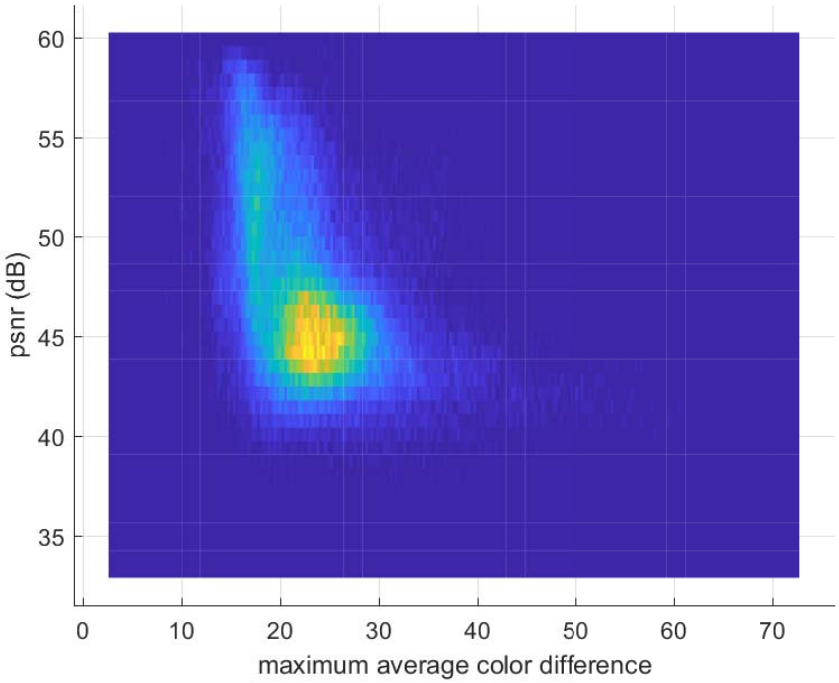


Figure 5-8: PSNR as a function of maximum average color difference in the Pentax dataset.

5.5 Combined Specular and Diffuse Reflection

Recall the Torrance-Sparrow model, given in (2.19) and repeated below

$$N = a(\lambda)N_i(\lambda)\cos(\psi) + N_i(\lambda)F(\hat{n}(\lambda))f(x) \tag{5.8}$$

where it is assumed that the Fresnel coefficient is a constant over the angles of interest.

This is not a separable function of position and wavelength, and so it is problematic for the model. However, consider three specific cases: First, the material is almost completely diffuse ($a(\lambda) >> F(\hat{n}(\lambda))$); second, the material is almost completely specular ($F(\hat{n}(\lambda)) >> a(\lambda)$); third, the material is evenly diffuse and specular ($a(\lambda) \approx F(\hat{n}(\lambda))$).

In the first two cases, the small component can be reasonably ignored, yielding an equation that is a separable function of position and wavelength, as previously

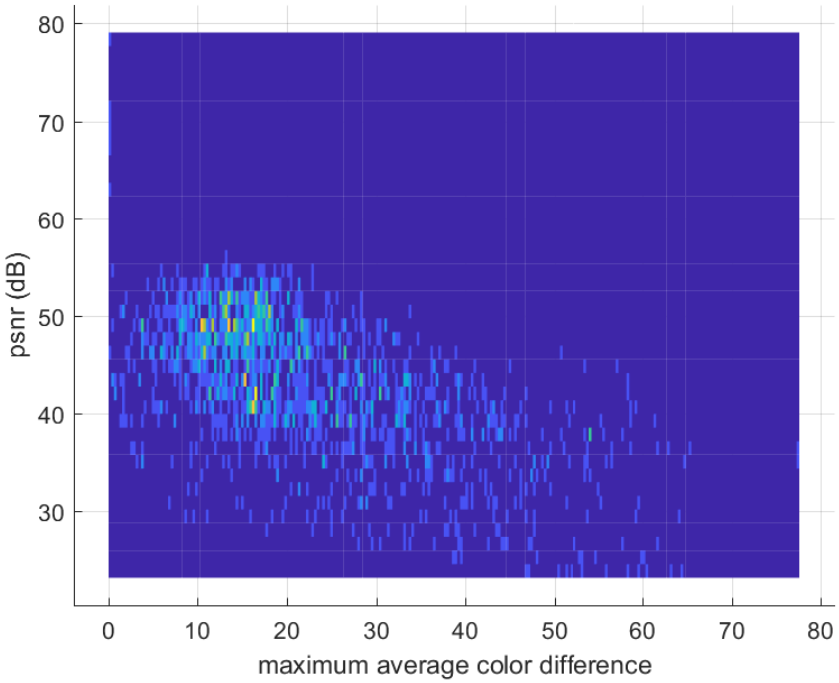


Figure 5-9: PSNR as a function of maximum average color difference in the Kodak dataset.

discussed. In the third case also, we can approximately write

$$F(\hat{n}(\lambda)) = a(\lambda) \tag{5.9}$$

and then (5.8) becomes

$$N = a(\lambda)N_i(\lambda)\cos(\psi) + N_i(\lambda)a(\lambda)f(x) \tag{5.10}$$

$$= a(\lambda)N_i(\lambda)(\cos(\psi) + f(x)) \tag{5.11}$$

which is also separable.

This means that primarily diffuse, primarily specular, and equally balanced diffuse and specular materials are all well approximated as separable functions. Therefore, it is only materials that have significant but unbalanced specular and diffuse reflective components that are going to be a problem. Although such regions are difficult to detect, it does not seem likely that this is a dominant source of error.

5.6 Chromatic Aberrations

Chromatic aberrations occur in images because the camera lens does not refract light of all wavelengths identically. These artifacts can be manifested as either a blurring of the color channels with different blurring functions or as an expansion or contraction of the color channels relative to each other, and generally result in unnaturally colored "halos" around objects in regions of high contrast at the edges of photographs. (For a basic introduction to chromatic aberrations, see [28] and the references therein.) An example of an image with chromatic aberrations is shown in Fig. 5-10a, which is a fragment of Sigma image 33.

In the simplest case, chromatic aberrations can be thought of as registration problems–the value of $I(x_0, \lambda)$ in the red channel has been mapped to location x_0 in the image, but the value of $I(x_0, \lambda)$ in the green channel has been mapped to location $x_0 - 1$ in the image. This will cause modeling errors since the residuals computed in the different color channels at image location x_0 actually correspond the residuals belonging to two different scene locations, and therefore they are not expected to be equal across color channels.

The result of modeling an image that contains chromatic aberrations is shown in Fig. 5-10b. Here, we can see that the washing out of the colors typical of modeling artifacts has affected the aberrant pixels and that the halos have been reduced. While this is erroneous behavior in the sense that the modeled image is not the same as the original, the modeled image arguably looks more natural and could be considered an enhancement.

5.7 Bright Regions

Although in theory the model should work equally well in bright and dark regions, performance is generally better in darker regions. This can be seen most clearly in images that contain a shadow. An example is given in Fig. 5-11, which shows a closeup of Sigma image 16, the result of modeling, and the difference between the

(a) Original

(b) Modeled

Figure 5-10: Sigma image 33 and the result of modeling.

two, normalized so that a difference of zero is mid-gray.

Although the modeled image is visually indistinguishable from the original image, the difference image makes it clear that the model performs better in the shadowed region. This phenomenon happened consistently throughout all the datasets. However, when the images of the entire dataset were segmented into 16x16 pixel blocks and the PSNRs of the modeled blocks were plotted as a function of average luminance of the block, no pattern emerged, as shown in Figs. 5-12 through 5-14, which plot the two-dimensional histograms of PSNR as a function of average luminance. Brighter regions mean more blocks had that PSNR and average luminance. These results indicate that, while the greater errors in brighter regions are apparent in any given image, the overall performance of the model on the image is mostly controlled by other factors.

It is unclear why this discrepancy occurs. It could be due to the greater differences in slope of the camera calibration curves at higher pixel values, but in this case the Pentax dataset would exhibit the effect more strongly due to the more dramatic slope difference in the blue channel, whereas the Sigma calibration curves had more equal slopes. However, this is not the observed behavior. Furthermore, the errors often occur at edges, as can be seen in Fig. 5-11. It may be that in darker regions, edge errors are limited because the range of the values taken by any color channel is limited. However, we would expect this to result in better edge detection in bright regions, so it is difficult to say with confidence that it would produce smaller errors in dark regions. It is possible that, because the light is more diffuse in shadowed regions, bright regions have more significant specular reflections, which could cause model errors in some cases, though this is also difficult to confirm. Determining the cause of this phenomenon is an area for potential future work.

(a) Original

(b) Modeled

(c) Modeling error

Figure 5-11: Sigma image 16 and the result of modeling.

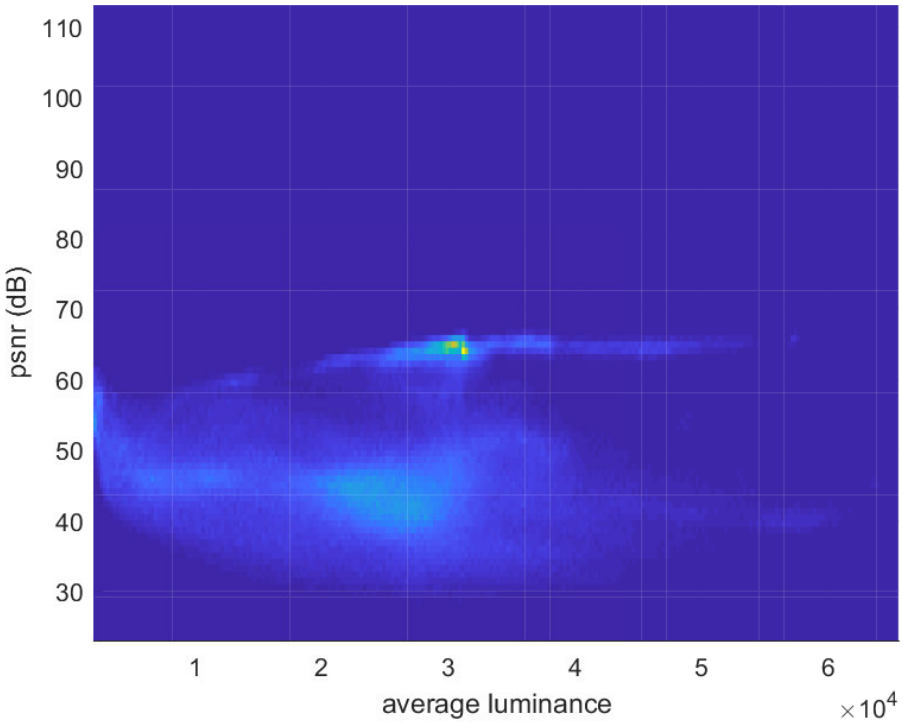


Figure 5-12: PSNR as a function of average luminance in the Sigma dataset.

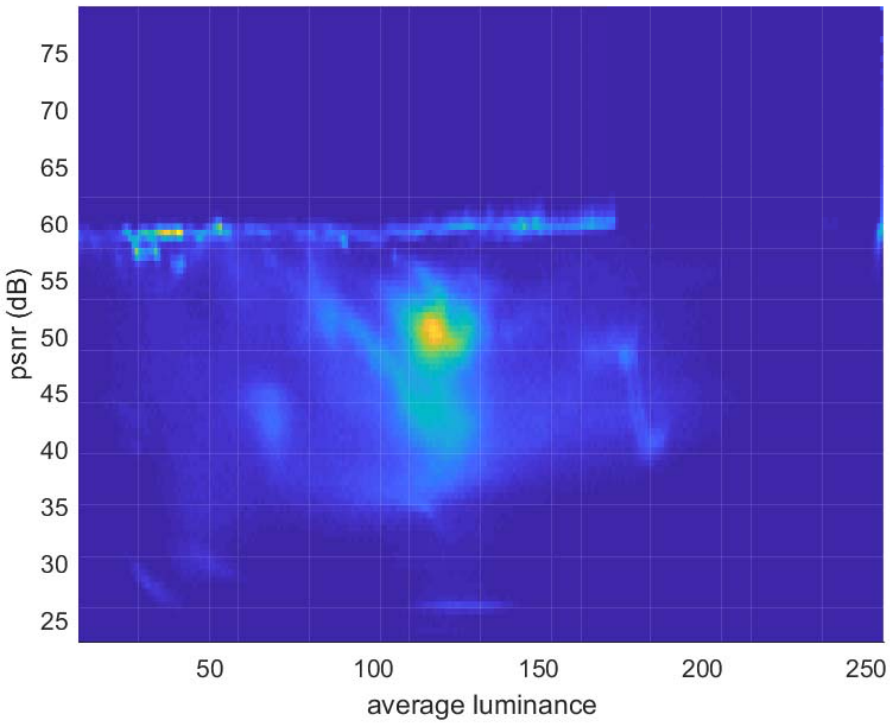


Figure 5-13: PSNR as a function of average luminance in the Pentax dataset.

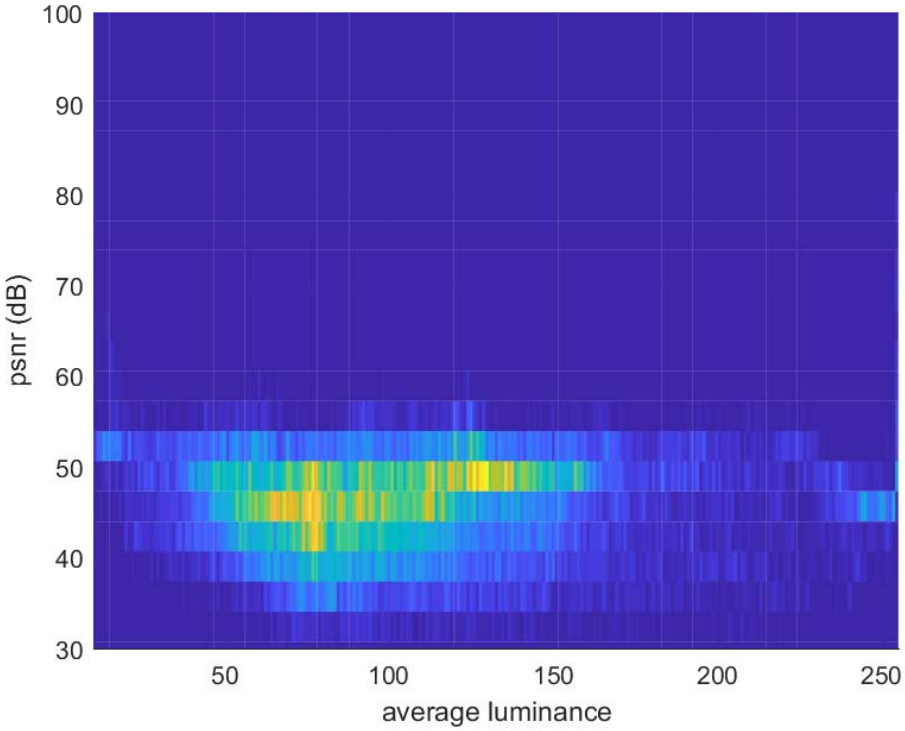


Figure 5-14: PSNR as a function of average luminance in the Kodak dataset.

Chapter 6

Implications of the Proposed Image Model

This model provides physical justification for three observations which have been previously made in image processing applications. First, the claim that the chrominance channels of an image tend to be low-frequency signals will be shown to follow from the model. Then, and relatedly, the claim that the luminance channel of a noisy image has a higher signal-to-noise ratio (SNR) than the original color channels will be shown to follow. Finally, the relationship between this model and the color lines model of Omer and Werman [51] will be explained.

6.1 Low Frequency Chrominance

In this section, the model will be used to validate the observation that the chrominance channels tend to be low-frequency signals. This observation has been used in demosaicing work. In an early paper [33], Kimmel provided some physical justification for this assumption by showing that the Lambertian model implied that the ratio of two color channels was a constant. Here, we will expand on this idea by using the model (which can be derived not only from the Lambertian model but from any physical model that results in a separable function of wavelength and position) to demonstrate the low frequency nature of the chrominance channels in the logarithmic

domain. There are several ways to define chrominance, but for illustration purposes, consider one of those used by the opponent color transform: $C(x) = \frac{1}{\sqrt{6}}R(x) - \frac{1}{\sqrt{6}}B(x)$. Applying the additive model of (3.26)-(3.28) gives

$$C(x) = \frac{1}{\sqrt{6}}(R'_{avg}(x) + q'(x) - (B'_{avg}(x) + q'(x))) \tag{6.1}$$

$$= \frac{1}{\sqrt{6}}(R'_{avg}(x) - B'_{avg}(x)) \tag{6.2}$$

By definition, $R'_{avg}(x)$ and $B'_{avg}(x)$ have most of their energy in the low frequencies. Therefore, the energy of $C(x)$ will also be concentrated in the low frequencies. Finally, it is easy to see that this analysis will hold for other combinations of R, G, and B, as long as the weights sum to zero.

Similarly, the model can also be used to show that the image gradients should be the same in all color channels, except at material boundaries. For example, consider computing the difference between two neighboring pixels in the red channel, $R'(x)$ and $R'(x+1)$. If both pixels are within the same material, then their local averages can be assumed to be the same and the gradient g is given by

$$g = R'(x) - R'(x+1) \tag{6.3}$$

$$= R'_{avg} + q'(x) - (R'_{avg} + q'(x+1)) \tag{6.4}$$

$$= q'(x) - q'(x+1) \tag{6.5}$$

Of course other methods of computing the two-dimensional discrete gradient exist, but this illustrates the general idea–the local averages will be canceled out and only the residuals remain.

If, on the other hand, the two pixels were in different materials, then the local averages would not be the same and the gradients would be different across color channels. This implies that the gradients of the chrominance channels will be zero except at material boundaries, and that chrominance gradients are useful features to examine when doing contour detection. This has indeed been shown to be the case in the literature [8, 74].

This observation also provides some justification for using the largest gradient in any of the three channels when computing histograms of image gradients for feature detection, as done in [11] and many of the methods built on that work. When the gradients are primarily due to image texture, they will likely be about the same in the color channels, and so it does not matter which is chosen. On the other hand, when the gradients are due to material edges, choosing the largest gradient among the color channels will do the best job of indicating the presence of a boundary.

6.2 Luminance Has Lower Noise

The explanation for why the SNR of the luminance component is higher than it is for any individual color channel is simple. Again using the additive model and the definition of "luminance" from the opponent color transform, the noisy luminance component $\tilde{Y}(x)$ is given by

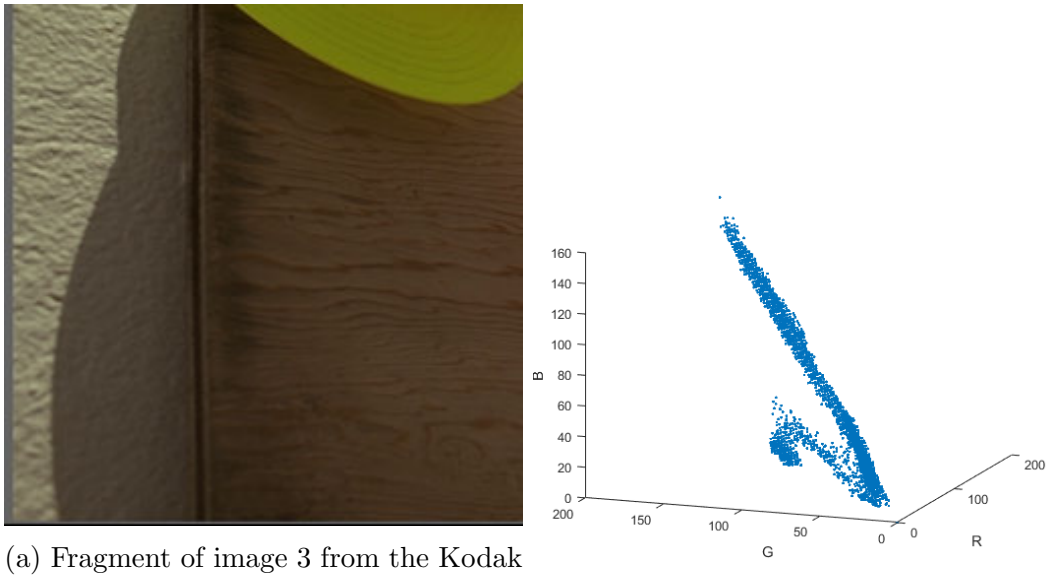
$$\tilde{Y}(x) = \frac{1}{3}(\tilde{R}'(x) + \tilde{G}'(x) + \tilde{B}'(x)) \tag{6.6}$$

$$= \frac{1}{3}(R'_{avg}(x) + q'(x) + n_R(x) + G'_{avg}(x) + q'(x) + n_G(x) + B'_{avg}(x) + q'(x) + n_B(x)) \tag{6.7}$$

$$= \frac{1}{3}(R'_{avg}(x) + G'_{avg}(x) + B'_{avg}(x)) + q'(x) + \frac{1}{3}(n_R(x) + n_G(x) + n_B(x)) \tag{6.8}$$

where tildes indicate a noisy component.

SNR is defined as the ratio of the variance of the signal to the variance of the noise. Since the local averages are slowly varying, the variance of the signal in the color channels is primarily contained in the residual. This residual is preserved in the computation of the luminance. The variance of the noise, however, is reduced by a factor of three since three independent noise samples are averaged. Therefore, we have verified the observation that the luminance component will have a higher SNR than the individual color channels.



(a) Fragment of image 3 from the Kodak dataset

(b) Pixel values plotted in RGB space

Figure 6-1: Example of color lines.

6.3 Color Lines

It has been observed [51] that when the values of the pixels of an image are plotted in RGB space, they tend to form discrete groups that are approximately linear, as shown in Fig. 6-1. The midpoint of each line can be thought of as a basic color used by the image, while the pixels along the line can be thought of as being of that base color but having different RGB values because of textures or illumination changes. Similar ideas were derived from specific image models in [20, 25, 35], though these earlier works present very limited experimental verification.

This observation has been used in segmentation [51], by assuming that pixels within a single color line belong to a single image segment, and in matting [37], where it provided a way to extend a cost function on the foreground opacity of a pixel to color images. It has also been used in denoising [71], by setting constraints in the denoising algorithm to encourage the movement of pixels towards the color lines rather than along them. In this section, it will be shown that the model predicts not only that such color lines exist, but that their slope is one. Some experimental results will also be presented verifying that this is so.

6.3.1 Model Predictions

If we consider only pixels within a single material, then the local averages of (3.26)-(3.28) are no longer functions of position. In this case, the model predicts that the values of the pixels will fall on a line in RGB space. Furthermore, this line will have a slope of one and will intercept the point $(R'_{avg}, G'_{avg}, B'_{avg})$.

Unfortunately, this prediction is difficult to verify because doing so requires segmenting the images into different materials, which is itself a hard problem. Therefore, several related predictions that are easier to verify will be considered.

First, if the images are segmented into square blocks of reasonable size, it can be expected that some of these blocks contain only one material. The pixels within these blocks should fall on a line, and the slope of that line should be one. Second, if only pixels that have similar local averages are selected, regardless of their spatial locations, these pixels should also fall on a line of slope one. Finally, if the local averages are subtracted from the images, effectively normalizing all pixels to have a local average of zero, then all the pixels of all the images in the dataset should fall on a line of slope one.

6.3.2 Experimental Verification

In this subsection, results verifying the predictions listed in the previous subsection on the images of the Sigma, Pentax, and Kodak datasets will be presented.

First, the images were segmented into 16x16 pixel blocks. The intent was that these blocks be large enough that the data within them was meaningful, but small enough that they would generally contain only a single material. The line of best fit to the pixel values was then computed and those blocks that were not well-described by a single line were discarded. "Well-described" blocks were selected by computing the error of the data to the best-fit line and discarding those blocks where the error was too large. This was necessary in order to discard those blocks that contained two or more objects. 40% of the blocks in the Sigma dataset, 65% of the blocks in the Pentax dataset, and 57% of the blocks in the Kodak dataset were retained. Since

only blocks that were well-described by lines were intentionally selected, these results cannot be used to say that lines will form, although the fact that they did in about half the blocks is encouraging. However, we can look at the slopes of the lines that did form and verify that they are close to one, as expected. The average slopes are shown in Table 6.1.

Table 6.1: Average slopes, images segmented into blocks

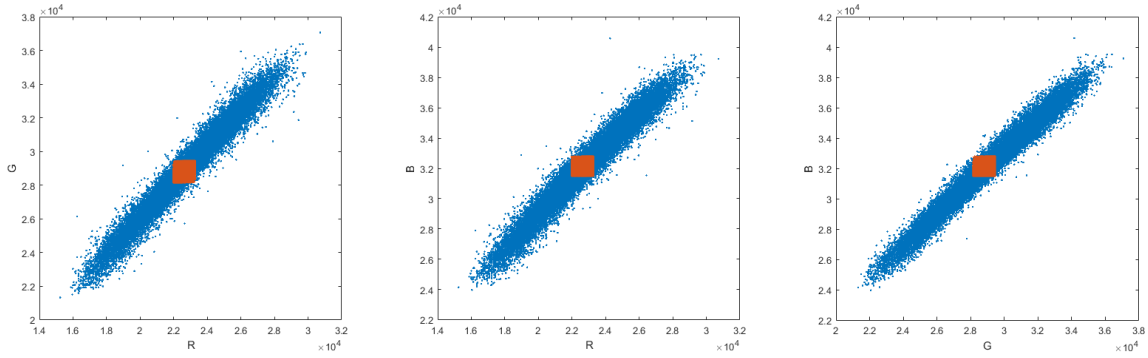
	Sigma	Pentax	Kodak
Average Slope R vs G	.9837	1.0252	.9252
R vs B	.8834	1.1017	.8213
G vs B	.8742	1.0442	.8516

Second, the images were segmented based on pixel value. In particular, the edge-sensitive local averages of the pixels in the three color channels were computed and those pixels whose local averages were similar were taken as a group. The global average of the image was used as a reference point and pixels whose local averages were within $\pm 1\%$ of the dynamic range of the image of the global average in each color channel were taken to form the group. Most of the images in the datasets had sufficiently many pixels in this range for these groups to be meaningful, though a few images that did not have enough pixels and one that had many small material edges that went undetected when computing the local average (Pentax image 13) had to be discarded.

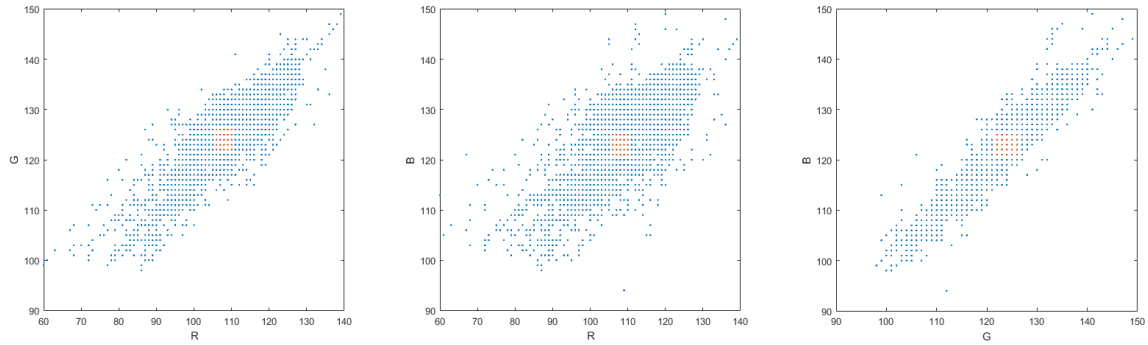
Some examples are shown in Fig. 6-2. These figures take all the pixels of the image whose local averages fall within the orange square and plot the value of one color channel as a function of another. The images used to generate these plots are shown in Fig. 6-3. Although the length and tightness of the lines varied from image to image within the datasets, these examples are representative.

The average slopes of the lines formed by the images in each dataset are shown in Table 6.2. These results also support the idea that color lines with a slope of one occur in images.

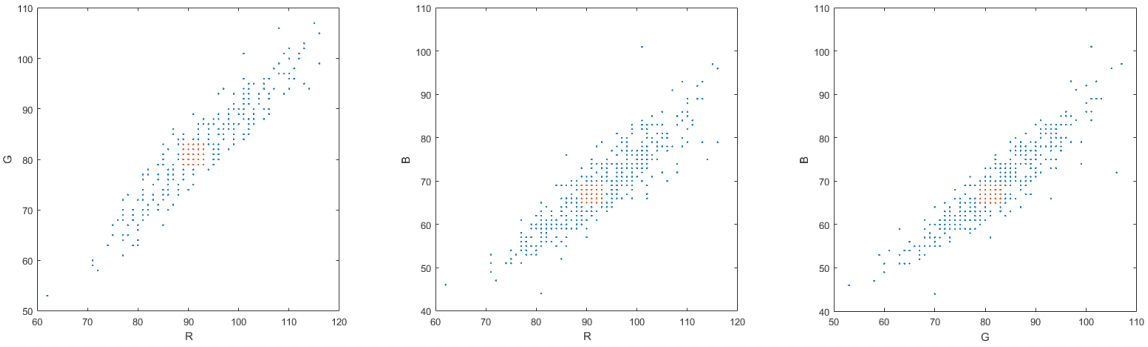
As a final test, the local average was subtracted from the image. This effectively



(a) Pixel values of image 28 from the Sigma dataset



(b) Pixel values of image 12 from the Pentax dataset



(c) Pixel values of image 5 from the Kodak dataset

Figure 6-2: Examples showing the values of pixels with similar local averages.

(a) Sigma image 28 (b) Pentax image 12 (c) Kodak image 5

Figure 6-3: Images used to generate the examples in Fig. 6-2.

Table 6.2: Average slopes, images segmented by local average

	Sigma	Pentax	Kodak
Average Slope R vs G	.8532	.9141	.9112
R vs B	.7716	.9340	.8459
G vs B	.8756	1.0244	.9329

normalizes the local average of all pixels to zero, and so we can look at the values of all the pixels in an image, and, indeed, in a dataset, on a single plot. Fig. 6-4 shows the 2D histograms of the values of all the normalized pixels in the three datasets. The narrow horizontal and vertical lines visible in the figures are not meaningful–they are only an effect of how the plots were rendered. Again, these results show that, once the local averages have been removed, the values of the pixels tend to fall on a line of slope one in RGB space.

These results verify the predictions of the previous section. Taken together, they provide good evidence that the values of the pixels of an image will fall on distinct lines of slope one, as expected.

6.3.3 Discussion

It is worth noting that when Omer and Werman proposed their color-line model, they rejected the idea that it was explainable by the Lambertian model [51]. This rejection was based on two observations: First, the color "lines" were not actually straight lines, and, second, that they did not intercept the origin, as the Lambertian model would predict. Therefore, they required, on average, six points to define the clusters of data they referred to as "lines."

The results presented here indicate that color lines are in fact straight lines in many instances. The difference is likely due to the nature of the datasets. Omer and Werman present results for a dataset which they indicate had been compressed aggressively, and which had probably been subjected to gamma correction. The datasets used here were uncompressed. In the case of the Sigma and Pentax datasets, the most raw images available from the camera were used for testing. Therefore, it

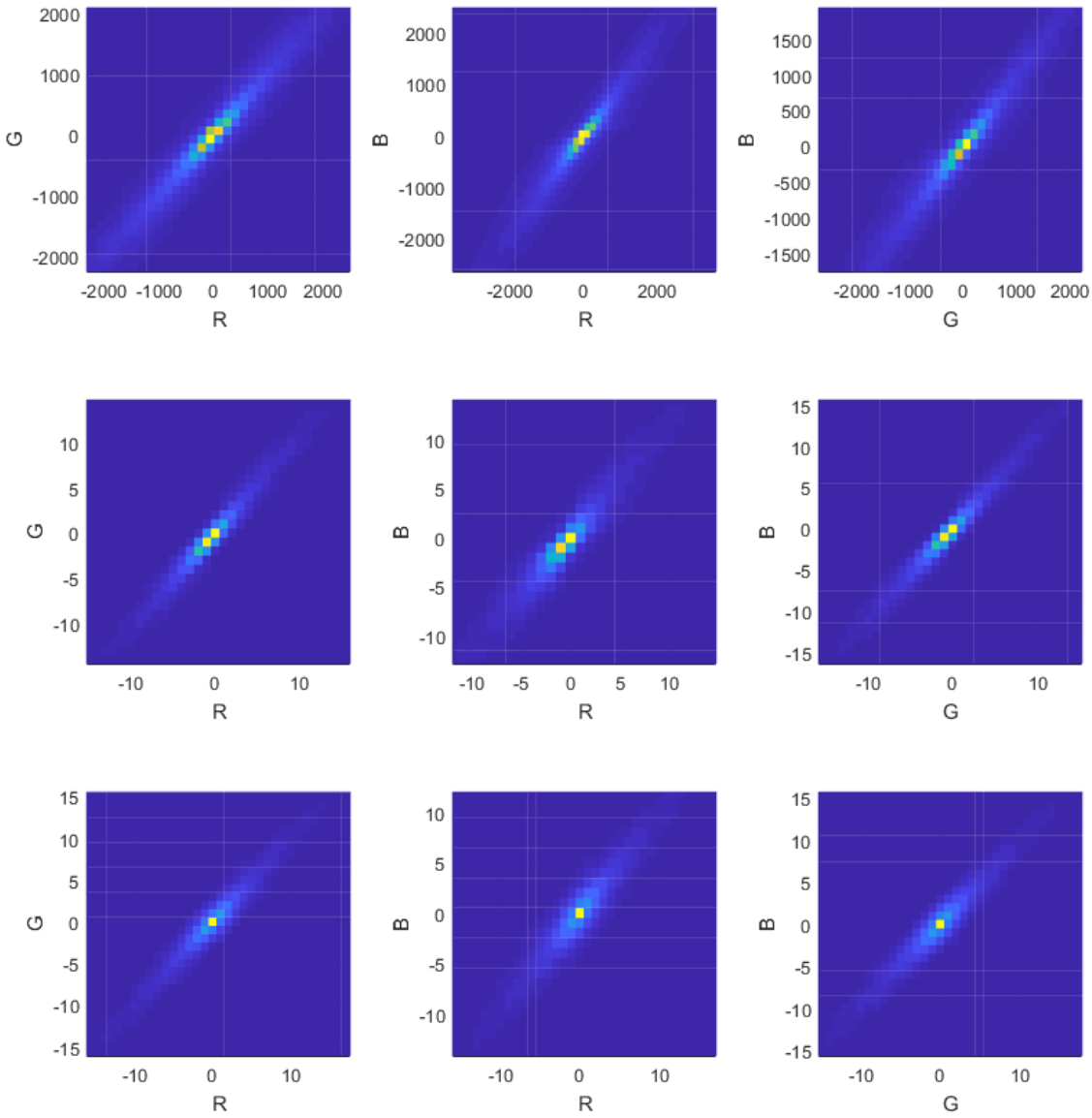


Figure 6-4: Plots of normalized pixel values from the Sigma (top), Pentax (middle) and Kodak (bottom) datasets.

is reasonable to conclude that the non-linear nature of the groups observed by Omer and Werman is due primarily to camera post-processing and is not inherent in the original images.

The notion that the Lambertian model requires the color lines to pass through the origin is only correct if the camera is reporting values that are linearly related to the scene irradiances. However, as discussed in Chapter 4, it is likely that the cameras are actually reporting something similar to the logarithm of the scene irradiance. Once this transformation is accounted for, there is no longer any reason to expect the color lines to intercept the origin; rather, as was shown, we should expect a slope of one.

Chapter 7

Imposing Model Constraints

Image modeling can be thought of as imposing constraints on an image. This is useful in the context of image restoration, where image priors are used to obtain an estimate of an original image from a degraded one. In this chapter, the effect of imposing the constraints of the model in several situations will be examined. As we will show, the model constraints can be used to extend arbitrary monochromatic image denoising systems to color images. Therefore, we will start with a brief overview of how this has been done in the past. We will then look at imposing the model constraints on clean images, on images degraded with additive white Gaussian noise, on images degraded with Poisson noise, and on images corrupted by both additive noise and blurring.

7.1 Background

The image denoising problem has been widely studied. The paradigm most commonly addressed is that of black and white images degraded by additive white noise, and the question of how to extend these methods to color images is not always addressed. For example, [46] provides an excellent overview of and framework for modern spatial denoising methods, but never mentions the issue of color. The simplest way to denoise a color image with a monochromatic method 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.

A more sophisticated idea is to treat each pixel as a three element vector. This is the approach taken in [4]. In this method, the non-local means algorithm, which does denoising by taking a weighted average of pixels in similar neighborhoods, is applied to color images by simply considering the Euclidean distance between two pixel values in three-dimensional space when computing the filter weights. This is a straightforward extension of the black and white image method, which computes weights based on the absolute difference between two pixel values. This method does not exploit spectral correlations as such, but rather leverages the fact that two pixels being "similar" means that they will be similar in all three color channels and therefore that there are three independently noisy datasets (the three color channels) available to use in measuring similarity.

Treating the pixel values as vectors is also used in [44], where the authors extend a Markov random field model of monochromatic images to color images by using three-dimensional cliques rather than two-dimensional ones and develop an appropriate image prior based on color training data. Pixel values are also considered as vectors in [41], in which the authors extend the K-SVD denoising algorithm to color images. In this method, a dictionary is obtained based on color images. In addition, the inner product is redefined to favor reconstructed patches that have the same average color as the original, and this new inner product is used when norms are computed throughout the algorithm. The authors claim that this avoids the "washed out" color artifacts seen in the reconstructions produced by the method in [44]. As another example, in [23], the authors propose a spatially varying filter based on a guidance image. The filter parameters are computed based on the local statistics of the guidance and input images. The filter is then extended to color images by taking the value of a pixel to be a vector and modifying the computation of the statistics accordingly. In any method that relies on training data, it is likely that spectral correlations will be accounted for since they will be present in the training data.

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 [32] and [63] and

expounded on in [34], 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 [32] and [34] for a full explanation. However, it is worth noting here that the fact that the gradients are the same in the three color channels is used to make certain parameter choices. The authors derive this fact from the Lambertian model, but, as we have shown in Chapter 6, this derivation can be generalized to come from any model that is a separable function of position and wavelength.

Other denoising algorithms first transform the RGB image to the YUV (or similar) domain. For example, in [65], the authors propose a curvelet transform-based monochromatic denoising system which they extend to color images by transforming the image to the YUV space and then processing each channel independently before converting back to the RGB space. Similarly, the kernal regression method of [66] is extended to color images by transforming to the YCbCr space and processing each channel independently. In neither case do the authors give any particular motivation for taking this transform. The authors of [66] do mention [16] as precedent, though [16] is dealing primarily with demosaicing. It is, however, conceivable that the higher SNR of the luminance channel and the low frequency nature of the chrominance channels give some advantages.

In [67], bilateral filtering, which involves computing a weighted average of pixels that are similar in both range and domain, is applied to color images by first transforming them to the CIE-Lab color space. Since Euclidean distances in this color space are meant to correspond to perceptual color differences, the weights are computed based on the perceptual difference between the pixel colors and only pixels that are visually similar to each other are averaged. As with viewing color pixels as vectors, this does not take advantage of the spectral correlations per se, but does use the independent measurements available for each pixel.

The well-known BM3D algorithm [10] has been extended to color images in [9] by transforming the noisy image to the YCbCr or opponent color space and then using

the less noisy luminance component to do the block matching. The blocks found to be similar in the luminance component are also assumed to be similar in the chrominance components. Except for this sharing of method parameters, the three channels are processed independently and then transformed back to the RGB domain. The idea of sharing parameters computed in the less noisy luminance component is also used in [18] and [5]. As shown in Chapter 6, the model explains the less-noisy nature of the luminance channel, so, while not using the model directly, these methods do take advantage of the correlations the model implies.

Another technique is given in [59]. 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. Broadly speaking, this indicates that images are expected to have the same structure in all three color channels, which our model indicates as well.

Some recent methods are based more explicitly on modeling. For example, the color lines model discussed in Chapter 6 is used for denoising in [71], where the guided filter of [23] is modified so that it encourages the values of noisy pixels to move towards the color lines rather than along them, and in [39], which uses it to denoise color images that have had spatial dependencies introduced in the noise as a result of demosiacing. A somewhat similar idea is used in [29] for simultaneous deblurring and denoising, though the concept of color lines is generalized to handle edges as well. As another example, an affine model is also proposed in [76]. This is a generalization of the color lines derived from our model because it does not require the slope to be one. The affine model is then used for denoising textured images by computing a MAP estimate of the original image, given the noisy image and an assumption that the texture can be represented with a fractional Brownian motion model.

In the remainder of this chapter, we will examine the effects of imposing the model

constraints directly in the contexts of various image restoration problems. While this may not be the best way to actually use the color correlations in practice, we believe this study is interesting because it provides a way to systematically analyze the possible gains available from the color correlations.

7.2 Clean Images

The effect of imposing the model constraints on clean images was dealt with in detail in Chapter 4. Modeling a clean image does not generally change it very much, but the changes that do happen are degradations, not improvements. Although the model is good, it is not perfect, as discussed in Chapter 5. Therefore, the differences between the modeled images and the clean images are due to modeling error and are not an improvement on the clean images.

7.3 Images Degraded by Additive White Gaussian Noise

In this section, the effects of imposing the modeling constraints on images degraded by additive white Gaussian noise will be described. First, the effect of imposing the constraints on the noisy images themselves will be considered. Then, the effects of imposing the constraints on images which have also been processed by a spatial denoising system will be discussed.

7.3.1 Noisy Images

Consider an image corrupted by additive white Gaussian noise. The values in the three color channels are given by

$$\tilde{R}'(x) = R'(x) + n_R(x) \tag{7.1}$$

$$\tilde{G}'(x) = G'(x) + n_G(x) \tag{7.2}$$

$$\tilde{B}'(x) = B'(x) + n_B(x) \tag{7.3}$$

where the noise in the color channels is independent.

Assume for a moment that the true local averages of the color channels are given by an oracle and that the model accurately describes the image. Then the residuals are computed from the noisy image as

$$\tilde{q}'_R(x) = \tilde{R}'(x) - R'_{avg}(x) \tag{7.4}$$

$$= R'_{avg}(x) + q'(x) + n_R(x) - R'_{avg}(x) \tag{7.5}$$

$$= q'(x) + n_R(x) \tag{7.6}$$

and so forth. The average residual is then estimated as

$$\hat{q}'(x) = \frac{1}{3}(\tilde{q}'_R(x) + \tilde{q}'_G(x) + \tilde{q}'_B(x)) \tag{7.7}$$

$$= q'(x) + \frac{1}{3}(n_R(x) + n_G(x) + n_B(x)) \tag{7.8}$$

Because the noise is independent in the three color channels, the noise variance will be reduced by a factor of three, and the modeled image is expected to have a PSNR improvement of about $10\log_{10}(3) \approx 4.77$dB. Since in practice the local averages were estimated from a 7x7 pixel window in the noisy image, the actual theoretical improvement is about 4.60dB. For a detailed derivation of this number, see Appendix B.

Notice that this noise reduction is due entirely to the *spectral* averaging that occurs when computing the average residual. It does not rely on the spatial averaging at all.

Although the model does require the computation of a local spatial average, this is not the source of the noise reduction. To illustrate this, consider a simple example where the image is flat and the noise, while spatially uncorrelated, is the same in all three color channels. Again, suppose the true local averages are given by an oracle. Since the image is flat, the local average will also be the value of the underlying pixel. However, we will compute

$$\hat{q}'(x) = \frac{1}{3}(\tilde{R}'(x) - R'_{avg}(x) + \tilde{G}'(x) - G'_{avg}(x) + \tilde{B}'(x) - B'_{avg}(x)) \tag{7.9}$$

$$= \frac{1}{3}(R'(x) + n(x) - R'(x) + G'(x) + n(x) - G'(x) + B'(x) + n(x) - B'(x)) \tag{7.10}$$

$$= \frac{1}{3}(n(x) + n(x) + n(x)) \tag{7.11}$$

$$= n(x) \tag{7.12}$$

The second step follows by replacing the noisy samples with the underlying clean samples plus the noise and noting that the local average is the same as the clean sample in this simple example.

Now, when $\hat{q}'(x)$ is used to compute $\hat{R}'(x)$, we find

$$\hat{R}'(x) = R'_{avg}(x) + \hat{q}'(x) \tag{7.13}$$

$$= R'(x) + n(x) \tag{7.14}$$

$$= \tilde{R}'(x) \tag{7.15}$$

that is, the modeled image is just the noisy image. Since the residual is added back to the local average, if there has been no noise reduction in the computation of the residual, the noise will also be added back in.

To test this, we took the images from each of the databases and added white Gaussian noise independently to each color channel. The noisy images were then modeled. A threshold was used to select only similar pixels to compute the average. However, unlike the noiseless case, there is no danger of degeneracy here. Too small

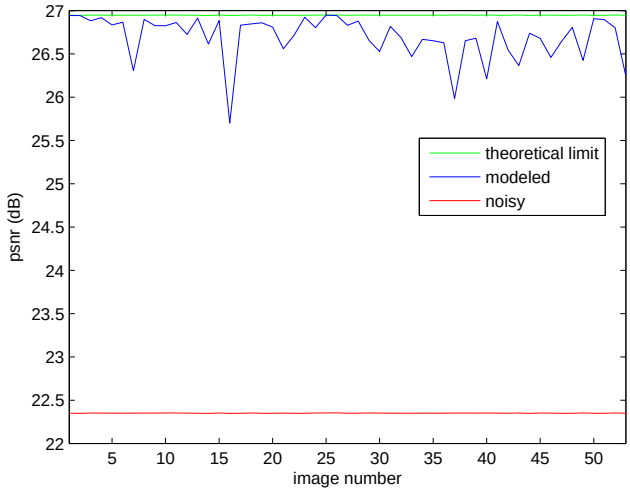


Figure 7-1: Results of modeling the noisy images of the Sigma dataset.

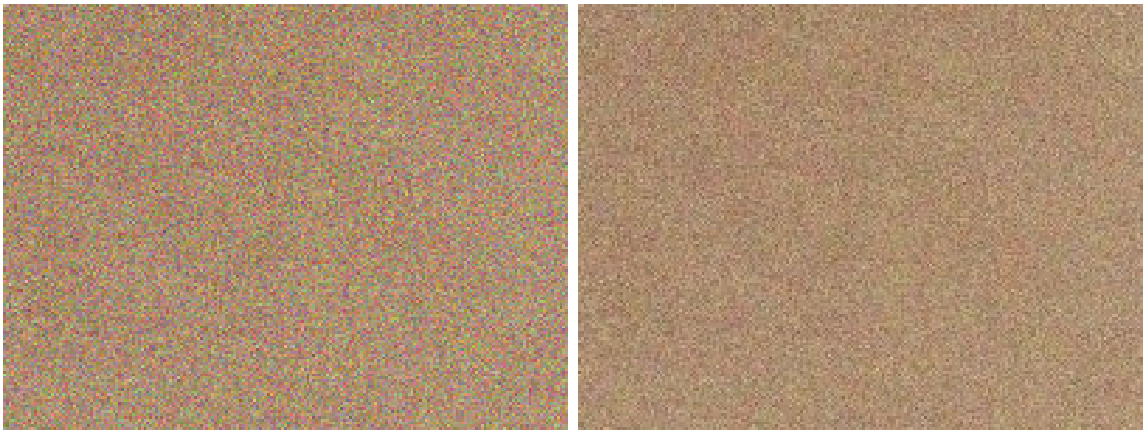
a threshold will simply give back the noisy image again. A threshold of seven times the standard deviation of the noise was found to work well empirically.

Results for the Sigma dataset are shown in Fig. 7-1. The red line shows the PSNR of the noisy image, the blue line is the PSNR of the denoised image, and the green line is the theoretical improvement limit of 4.60dB. The standard deviation of the noise was 5,000.

These results are quite good considering that this "denoising system" is not exploiting spatial correlation. In fact, the results achieve, on average, about 95% of the theoretically possible improvement.

Example images are shown in Figs. 7-2 through 7-4. Figs. 7-2a, 7-3a, and 7-4a are fragments of Figs. 4-4a, 4-5a, and 4-6a, respectively, with noise added. Figs. 7-2b, 7-3b, and 7-4b are the results of modeling Figs. 7-2a, 7-3a, and 7-4a, respectively. These results were typical of the images in the database and demonstrate that the increase in PSNR does in fact mean an increase in subjective visual quality.

Results for the Pentax and Kodak datasets were similar, and are shown in Figs. 7-5 and 7-6, respectively. For these tests, the noise variance was 20. The results for both datasets achieve about 95% of the theoretically possible improvement on average.



(a) Noisy (b) Modeled

Figure 7-2: Results of modeling noisy Sigma image 1.

(a) Noisy (b) Modeled

Figure 7-3: Results of modeling noisy Sigma image 16.

7.3.2 Images Processed by Spatial Denoising Systems

Since, as previously discussed, the model imposes only spectral constraints, it is also interesting to consider the effect of imposing those constraints on images which have been degraded by additive white noise and then processed by a spatially-based image denoising system. Here, the block-matching 3D (BM3D) algorithm proposed in [10] will be used. This method is a standard benchmark used in the denoising literature and it generally produces good results.

At a high level, BM3D works by grouping similar image patches into a 3D block and applying a 3D filter to that block. After filtering, the patches are returned to

(a) Noisy (b) Modeled

Figure 7-4: Results of modeling noisy Sigma image 47.

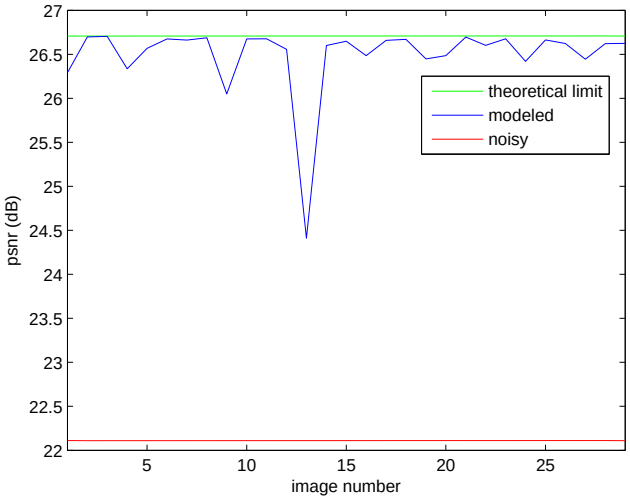


Figure 7-5: Results of modeling the noisy images of the Pentax dataset.

their original positions. The interested reader is referred to [10] and [9] for details. The performance of BM3D relies on a good initial block matching step, and so it is better to impose the modeling constraints before performing BM3D than afterwards– reducing the noise a little initially will allow for better block matching and increase the performance of BM3D. At low to moderate noise levels, BM3D can remove most of the additive noise and the remaining errors are due to processing artifacts, and may or may not be the same in all color channels, so the utility of imposing the modeling constraints afterwards will be limited.

It is difficult to place a theoretical limit on noise reduction when using a spatial

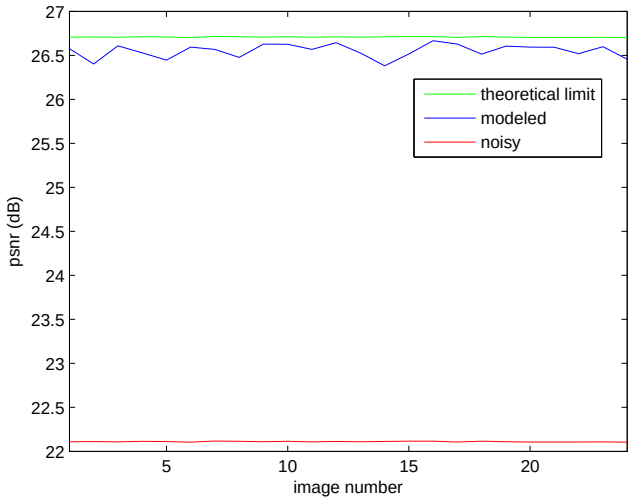


Figure 7-6: Results of modeling the noisy images of the Kodak dataset.

denoising system because the spatial statistics of "natural images" have yet to be well-defined. However, it has been shown [38] that BM3D achieves nearly (to within a few tenths of a dB) optimal performance among possible non-parametric methods based on the use of small image patches. Therefore, to bound the possible performance of modeling and using such a spatial denoising system, we use BM3D to denoise images initially corrupted with noise equivalent to the expected remaining noise level after modeling–that is, the variance of the noise in the images used to compute the limit is about one third the variance of the noise used to corrupt the images for actual testing. This gives us a bound on the performance of a denoising system of this type, though other types of denoising systems could be designed that surpass this bound.

To test the combination of modeling and BM3D, we added white Gaussian noise (with $\sigma = .07$ when the pixel values were normalized to lie in $0 - 1$) to the images in the three datasets and ran BM3D on both the noisy images and the modeled noisy image. Results are shown in Figs. 7-7 through 7-9. The red line shows the PSNR of the noisy image, the blue line shows the PSNR of the modeled image, the black line shows the result of doing BM3D denoising on the noisy image, the green line shows the result of doing BM3D denoising on the modeled image, and the magenta line is the PSNR of the theoretical limit, as described above.

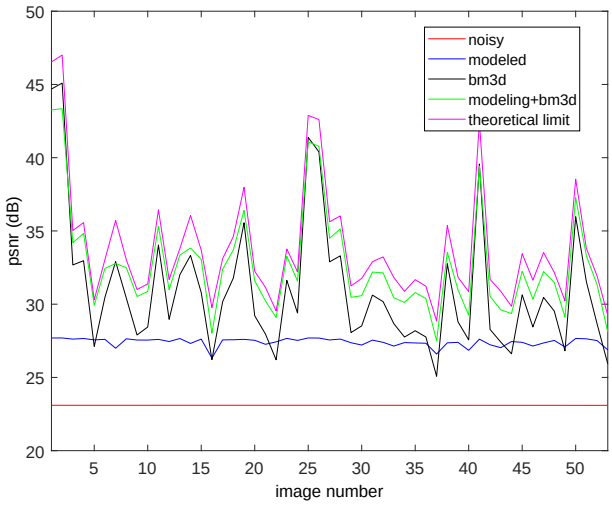


Figure 7-7: Results of imposing the modeling constraints on the noisy Sigma images and processing with BM3D.

These results indicate that first imposing the modeling constraints does improve the performance of BM3D, as expected. Performance is within about 1dB of the theoretical limit, indicating that we achieve most but not all of the possible gains. The gap is probably due to two things. First, even when the model performs well, it might introduce some artifacts. These artifacts may then adversely affect the performance of BM3D. Second, modeling introduces a slight spatial dependency in the noise due to the local average being estimated from the noisy image. This contradicts the assumptions made by BM3D, which relies on the spatial independence of the noise, and therefore could also cause some errors.

An example is shown in Fig. 7-10. Although BM3D has removed the additive noise without the help of the modeling constraints in Fig. 7-10c, it has also introduced some artifacts in the foliage, which has become blurred. The advantage of first modeling can be seen in Fig. 7-10d, where the original texture of the foliage has been better restored, albeit at the cost of some slight modeling artifacts around the edges of the truck.

These results demonstrate that imposing the modeling constraints before doing spatially-based denoising does usually result in an improvement, though there are a few cases where either the model performs exceptionally poorly (Pentax image 13) or

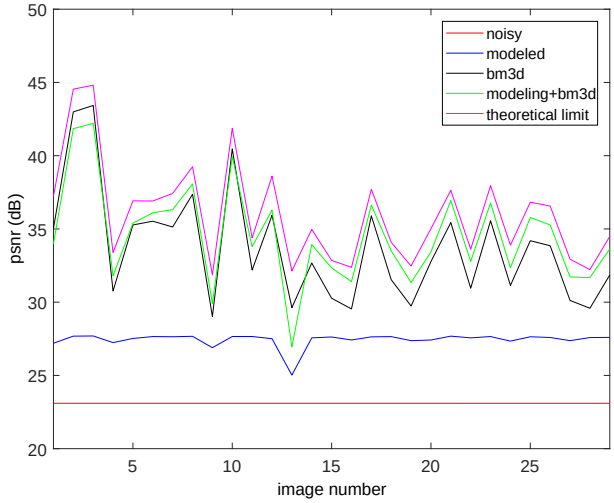


Figure 7-8: Results of imposing the modeling constraints on the noisy Pentax images and processing with BM3D.

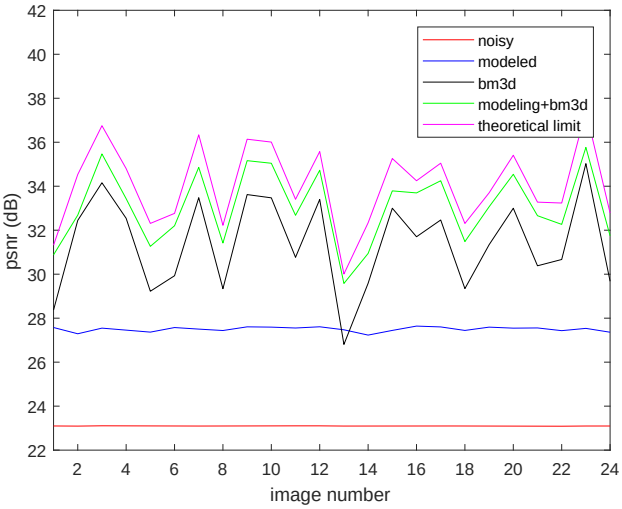
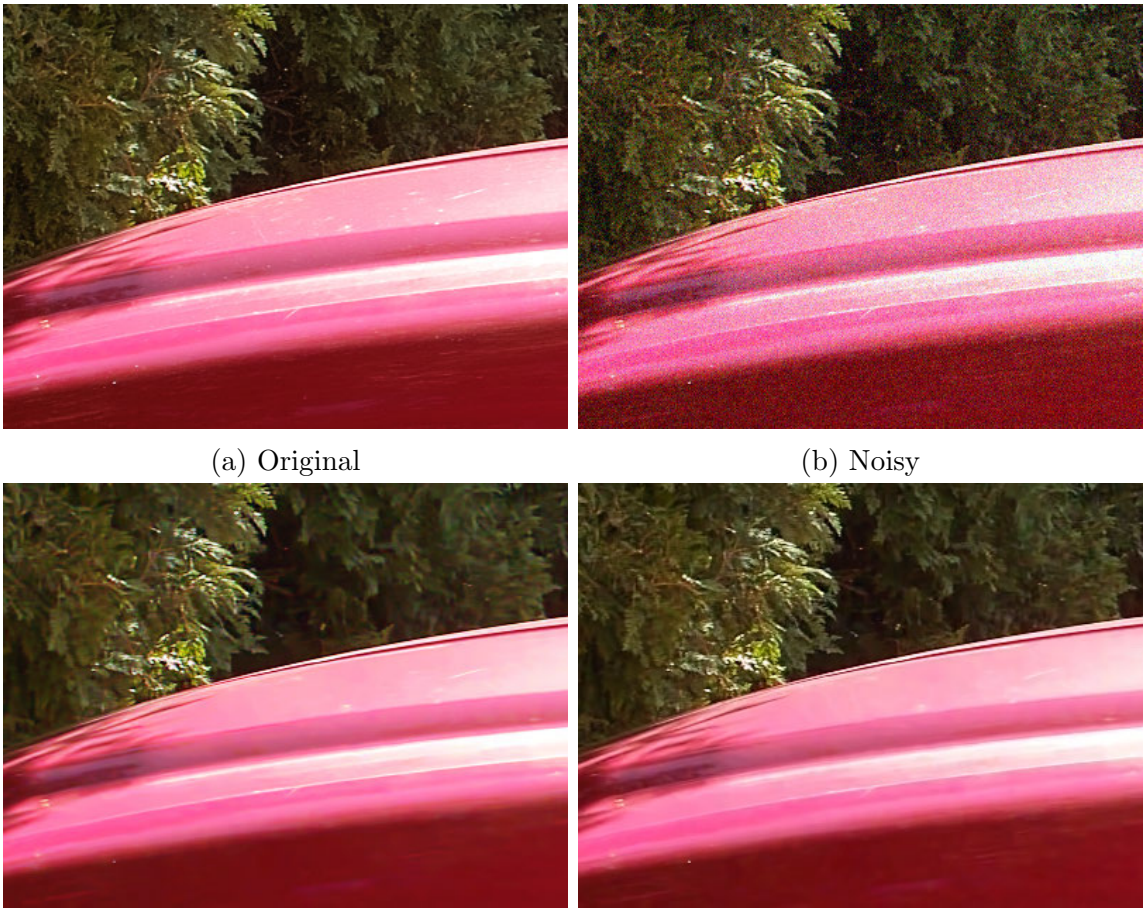


Figure 7-9: Results of imposing the modeling constraints on the noisy Kodak images and processing with BM3D.

BM3D by itself performs exceptionally well (Sigma images 1 and 2, Pentax images 2 and 3), where the modeling errors outweigh the gains.



(a) Original (b) Noisy

(c) BM3D denoised without modeling (d) BM3D denoised after modeling

Figure 7-10: Results of denoising Sigma image 47 with and without imposing the modeling constraints.

7.4 Images Degraded by Poisson Noise

Gaussian noise is not the only type of noise images may be corrupted by. Poisson, or shot, noise is another good model. This kind of noise is due to counting errors, and it is signal dependent in that, in a Poisson distribution, the variance is equal to the mean, which, in this case, is the correct underlying signal value. However, the noisy image can still be written as the sum of the underlying clean image and the noise:

$$\tilde{R}'(x) = R'(x) + n_R(x) \tag{7.16}$$

$$\tilde{G}'(x) = G'(x) + n_G(x) \tag{7.17}$$

$$\tilde{B}'(x) = B'(x) + n_B(x) \tag{7.18}$$

We can then solve for the additive noise, for example in the red channel:

$$n_R(x) = \tilde{R}'(x) - R'(x) \tag{7.19}$$

so the additive noise is a Poisson random variable minus the mean of that random variable. We will refer to this as an "offset Poisson" distribution. In this sense, although the *distribution* of the noise is signal dependent, the actual *realization* of the noise is not. This means we can still say that the noise samples are spectrally and spatially independent.

Notice that it is the image in the logarithmic domain being considered here. This is because these are the images that are actually available for testing. Derivations similar to the ones presented in this section could be worked out for the case when the Poisson noise was added before the logarithm was computed, though they might require the inversion of the logarithm and use of the multiplicative model instead.

7.4.1 Noisy Images

When an image that has been corrupted by Poisson noise is modeled, the fact that the noise is not identically distributed can be taken advantage of by giving a greater weight to those channels where the noise is expected to be smaller. Specifically, assume the following:

1. The local averages are known perfectly and therefore do not contribute to the error of the modeled image.

2. The model is perfect, and the non-noisy residual in each channel is given by $q'(x)$.

3. The weights used to compute the average residual are constrained to add to one.

Then the minimum mean squared error average residual can be computed as follows:

$$\min_{w_R(x),w_G(x),w_B(x)} \mathbb{E}[(q'(x) - (w_R(x)(q'(x) + n_R(x)) + w_G(x)(q'(x) + n_G(x)) + w_B(x)(q'(x) + n_B(x))))^2] =$$
$$\min_{w_R(x),w_G(x),w_B(x)} \mathbb{E}\left[(-w_R(x)n_R(x) - w_G(x)n_G(x) - w_B(x)n_B(x))^2\right]$$

Expanding, exploiting the linearity of expectation, and noting that the noise is zero mean and independent across color channels, we find that this becomes

$$\min_{w_R(x),w_G(x),w_B(x)} w_R^2(x)\mathbb{E}[n_R^2(x)] + w_G^2(x)\mathbb{E}[n_G^2(x)] + w_B^2(x)\mathbb{E}[n_B^2(x)] \tag{7.20}$$

To keep the following derivation as general as possible, we will not be specific about the variances of the noise values at this step. We will simply refer to them as $\sigma_R^2(x)$, $\sigma_G^2(x)$, and $\sigma_B^2(x)$, respectively.

The constraint that the weights sum to one can be enforced by writing $w_B(x)$ as a function of $w_R(x)$ and $w_G(x)$:

$$w_B(x) = 1 - w_R(x) - w_G(x) \tag{7.21}$$

And now the optimization problem becomes

$$\min_{w_R(x),w_G(x)} w_R^2(x)\sigma_R^2(x) + w_G^2(x)\sigma_G^2(x) + (1 - w_R(x) - w_G(x))^2\sigma_B^2(x) \tag{7.22}$$

This is just a two-dimensional quadratic, so to find the minimum, the derivatives can be set to zero, resulting in the following system of equations:

$$2w_R(x)\sigma_R^2(x) - 2(1 - w_R(x) - w_G(x))\sigma_B^2(x) = 0 \tag{7.23}$$
$$2w_G(x)\sigma_G^2(x) - 2(1 - w_R(x) - w_G(x))\sigma_B^2(x) = 0 \tag{7.24}$$

Solving simultaneously and substituting back into (7.21) gives

$$w_R(x) = \frac{\sigma_G^2(x)\sigma_B^2(x)}{\sigma_R^2(x)\sigma_G^2(x) + \sigma_G^2(x)\sigma_B^2(x) + \sigma_R^2(x)\sigma_B^2(x)} \tag{7.25}$$

$$w_G(x) = \frac{\sigma_R^2(x)\sigma_B^2(x)}{\sigma_R^2(x)\sigma_G^2(x) + \sigma_G^2(x)\sigma_B^2(x) + \sigma_R^2(x)\sigma_B^2(x)} \tag{7.26}$$

$$w_B(x) = \frac{\sigma_R^2(x)\sigma_G^2(x)}{\sigma_R^2(x)\sigma_G^2(x) + \sigma_G^2(x)\sigma_B^2(x) + \sigma_R^2(x)\sigma_B^2(x)} \tag{7.27}$$

Notice that this derivation works for any noise distribution, so long as the noise can be written as additive (which it always can) and the noise samples are spatially and spectrally independent.

To use these weights for the specific case of Poisson noise, notice that the variance will be the true pixel value, and so the weights become

$$w_R(x) = \frac{G'(x)B'(x)}{R'(x)G'(x) + G'(x)B'(x) + R'(x)B'(x)} \tag{7.28}$$

$$w_G(x) = \frac{R'(x)B'(x)}{R'(x)G'(x) + G'(x)B'(x) + R'(x)B'(x)} \tag{7.29}$$

$$w_B(x) = \frac{R'(x)G'(x)}{R'(x)G'(x) + G'(x)B'(x) + R'(x)B'(x)} \tag{7.30}$$

Obviously, it is not possible to compute the optimal weights exactly in practice because the clean image values are unknown. Here, three different estimation methods will be evaluated. First, the weights will be computed based on the noisy image values. Second, the modeled image will be computed iteratively, first computing the weights based on the noisy image, then computing the weights based on the modeled image (still modeling the noisy image). And finally, the clean image will be used to compute the weights. This will be done only for comparison purposes to determine how much of a performance penalty an imperfect knowledge of the weights imposes.

The results of these tests are shown in Fig. 7-11. As usual, the x-axis is the image number and the y-axis is the PSNR in dB. The red line shows the PSNR of the noisy image, the green line is the PSNR of the image modeled with equal weight given to the residual of each color channel (basic model), the black line is the PSNR of the image modeled with the optimal Poisson weights computed from the noisy image (p model

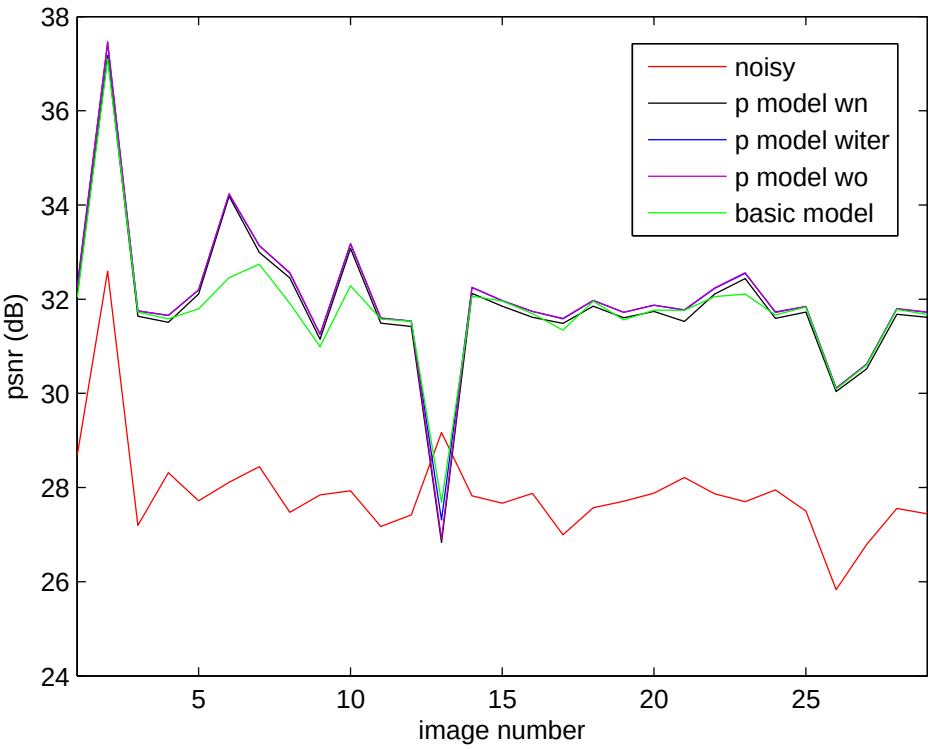


Figure 7-11: Results of modeling of the Pentax dataset in the presence of Poisson noise. The red line is the PSNR of the noisy images, the black line is the PSNR of the images modeled with weights computed from the noisy image, the blue line is the PSNR of the images modeled with weights computed from a prior iteration of modeling, the purple line is the PSNR of the images modeled with weights computed from the clean image, and the green line is the PSNR of the images modeled with equal weights.

wn, or "Poisson model weights from noisy"), the blue line is the PSNR of the image modeled with the optimal Poisson weights computed from the output of p model wn (p model witer, or "Poisson model weights from one iteration"), and the purple line is the PSNR of the image modeled with the optimal Poisson weights computed from the clean image (p model wo, or "Poisson model weights from oracle").

From this, it can be seen that modeling of any kind generally produces a substantial improvement in PSNR. The only exception is image 13, which contains many edges and is unusually difficult to model. Additionally, modeling with weights optimized for a Poisson noise distribution does generally produce a modest improvement over modeling with equal weights. Finally, computing the weights based on one it-

eration of the model is usually slightly better than computing them based on the noisy image and is essentially indistinguishable from using the weights that would be produced by an oracle. However, the gains are small enough that they may not justify the additional computational cost.

7.4.2 Images Processed by Spatial Denoising Systems

We will continue to focus on BM3D as the benchmark spatial denoising system. It has been extended to Poisson noise [42] by use of the Anscombe transform (AT) [2], which is the following mapping:

$$f : x \to 2\sqrt{x + \frac{3}{8}} \tag{7.31}$$

This is what is referred to as a variance stabilizing transform, and it is designed to take a Poisson random variable and transform it to something similar to a Gaussian random variable with the same mean and unit variance. The denoising method is straightforward: perform the Anscombe transform on the image, use BM3D to do denoising as usual, and then perform the inverse transform.

Combining modeling and BM3D is also fairly straightforward. It can be done similarly to the way it was done in the case of Gaussian noise, though the way the modeled image is computed is slightly different. The only substantial question is whether or not to use the Anscombe transform. When we compute the residual, we are computing the weighted average of three offset Poisson random variables. The distribution of the resulting noise is complicated. It will still be discrete, but it won't be limited to integer values, and therefore it won't be Poisson. The purpose is to decrease the variance while keeping the mean the same, and this is very contrary to the properties of a Poisson random variable. Since it is a sum of independent random variables, the central limit theorem indicates that the result could be similar to a Gaussian, though it is a small enough sum that there may still be substantial differences. Therefore, both using and omitting the Anscombe transform in conjunction with modeling and BM3D denoising will be tested here.

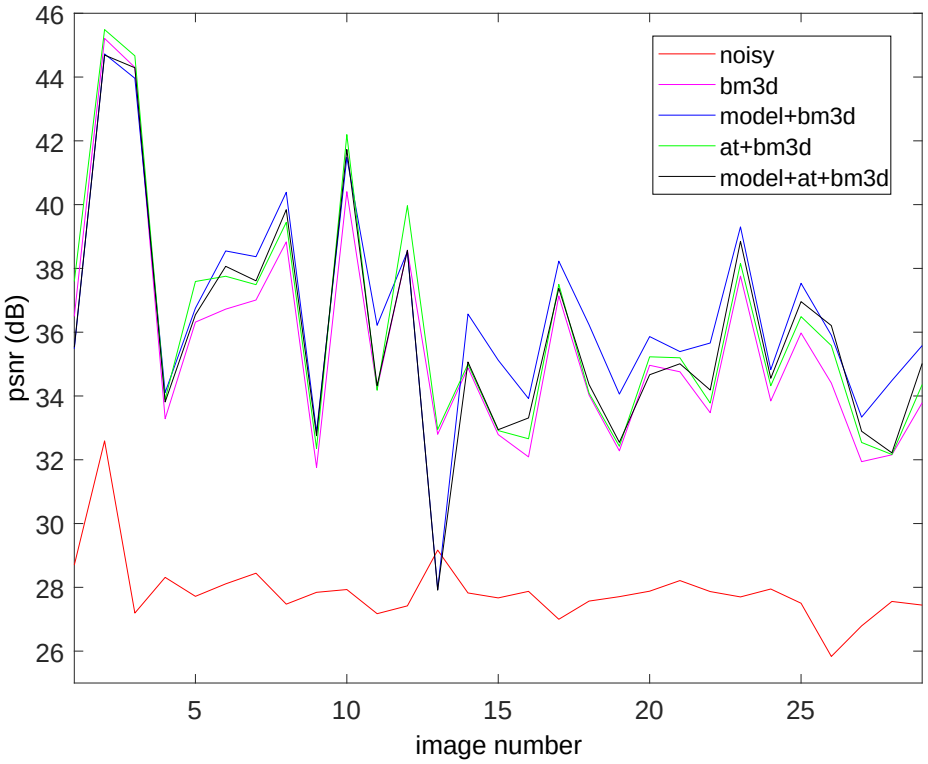


Figure 7-12: Results of imposing the modeling constraints on the images of the Pentax dataset when corrupted with Poisson noise and restored with BM3D.

Results are shown in Fig. 7-12. The x-axis is the image number and the y-axis is the PSNR in dB. The red line shows the PSNR of the noisy images, the magenta line shows the result of denoising with BM3D, the blue line is the result of denoising the modeled image with BM3D (the modeling weights were computed iteratively, as described in the previous subsection), the green line is the result of using the Anscombe transform and BM3D, and the black line is the result of modeling, then using the Anscombe transform and BM3D.

Results here vary from image to image, but generally modeling then doing BM3D performs better than modeling and then doing AT+BM3D. This indicates that the remaining noise has taken on characteristics closer to a Gaussian distribution after modeling and that the Anscombe transform, designed to handle Poisson noise, is now using the wrong model. Furthermore, modeling and then doing BM3D was generally an improvement over just doing AT+BM3D, which is reasonable since modeling is

able to take advantage of the spectral correlations.

7.5 Images Degraded by Noise and Blurring

Images can be degraded by blurring as well as noise. Here, only the case of spatially invariant blur will be considered, so the degraded image can be written as the convolution of the original image with a blurring kernel. Specifically,

$$\tilde{R}(x) = R(x) * w(x) + n_R(x) \tag{7.32}$$

where $\tilde{R}(x)$ is the red component of the blurred image, $R(x)$ is the red component of the original image, $w(x)$ is the blurring function, and $n_R(x)$ is white Gaussian noise. Similar equations hold for the other color channels. It will be assumed that $w(x)$ is the same in all three channels but $n_i(x)$ is a different realization of the noise distribution.

7.5.1 Blurred Images

For the moment, ignore the additive noise and consider modeling a blurred image. Recall that it is assumed that $I(x, \lambda)$ is a separable function of wavelength and position. That is,

$$I(x, \lambda) = f(x)g(\lambda) \tag{7.33}$$

After blurring, we have

$$\tilde{I}(x, \lambda) = \sum_k I(k, \lambda)w(x - k) \tag{7.34}$$

$$= \sum_k f(k)g(\lambda)w(x - k) \tag{7.35}$$

$$= g(\lambda) \sum_k f(k)w(x - k) \tag{7.36}$$

This is still a separable function of position and wavelength, so the earlier derivation of the multiplicative model holds and, if a blurred image is modeled, we expect to get

back the blurred image.

Alternatively, if the blurring is done after the logarithm is taken (as it is when clean images are blurred to test), we get

$$\tilde{I}(x,\lambda) = \sum_k I'(k,\lambda)w(x-k) \tag{7.37}$$

$$= \sum_k (f'(k) + g'(\lambda))w(x-k) \tag{7.38}$$

$$= \sum_k f'(k)w(x-k) + \sum_k g'(\lambda)w(x-k) \tag{7.39}$$

$$= \sum_k f'(k)w(x-k) + g'(\lambda)\sum_k w(x-k) \tag{7.40}$$

$$= \sum_k f'(k)w(x-k) + g'(\lambda)c \tag{7.41}$$

The first term is only dependent on location. In the second term, since the sum is taken over all k, the result is just a constant, independent of x, and so the second term is only dependent on wavelength. Therefore, the blurred image is still the sum of a color dependent function and a color independent function, and the additive model can be derived. Again, this means we expect to get back the blurred image if we model a blurred image.

To test this, we blurred the images in the datasets with a 25 point Gaussian blurring filter ($\sigma = .4$) and then modeled the burred images. The results are shown in Figs. 7-13 through 7-15. The red line shows the PSNR of the blurred image compared to the original. We define the PSNR of an image $f(n_1, n_2)$ compared to another image $g(n_1, n_2)$ as

$$PSNR = 10\log_{10}\left(\frac{MAX^2}{\frac{1}{N}\sum(f(n_1,n_2) - g(n_1,n_2))^2}\right) \tag{7.42}$$

where MAX is the maximum value possible in the image, N is three times the number of pixels, and the sum is taken over all the pixels and all the color channels. This is consistent with the definition we have been using up to this point. The blue line shows the PSNR of the blurred and modeled image compared to the blurred image,

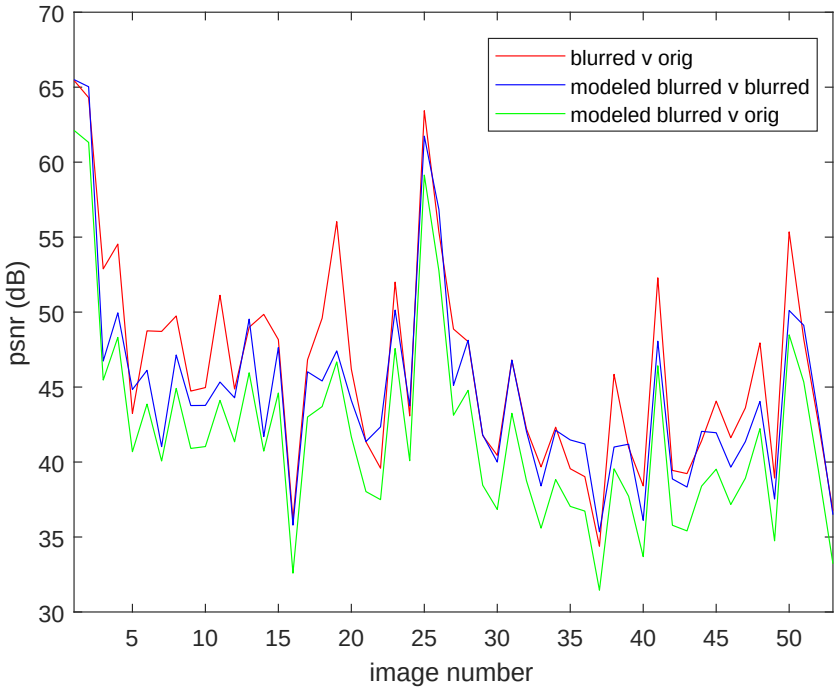


Figure 7-13: Results of imposing the modeling constraints on the blurred images of the Sigma dataset.

and the green line shows the PSNR of the blurred and modeled image compared to the original image.

These results are as expected–the high PSNRs of the modeled images compared with the blurred images indicate that modeling does return something similar to the blurry image. Furthermore, the fact that the PSNR of the blurred compared with the original image is higher than the PSNR of the modeled compared to the original indicates that no deblurring has occurred, which is also as expected. These results show that if the image were both blurred and noisy, applying the modeling constraints would give similar results to those in Section 7.3.1, except that the modeled image would be similar to the blurred image, rather than the original image.

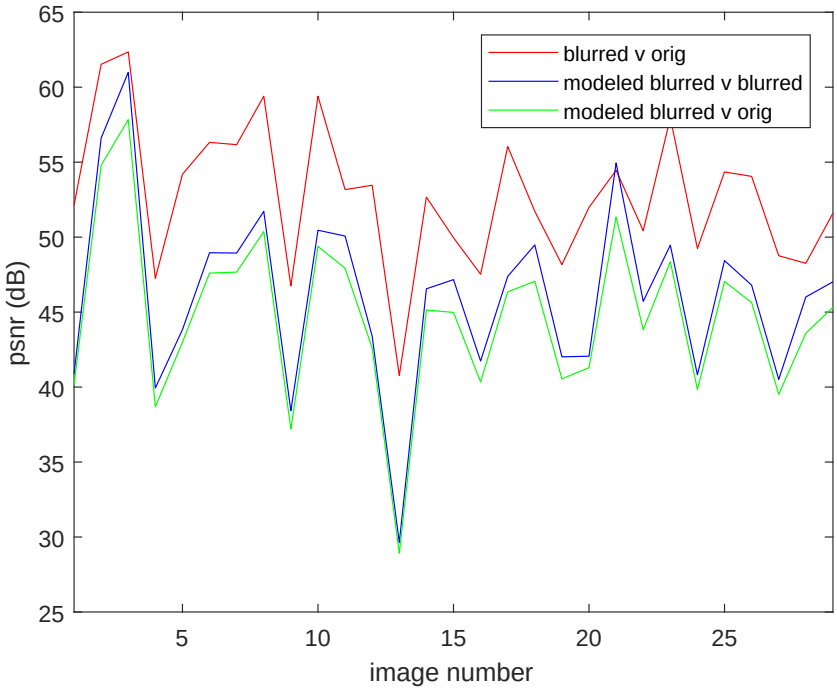


Figure 7-14: Results of imposing the modeling constraints on the blurred images of the Pentax dataset.

7.5.2 Images Processed with a Spatial Deblurring/Denoising System

The BM3D denoising algorithm has been extended to a deblurring algorithm for black and white images [12]. The basic idea is to take a linear algebraic view of BM3D, with the block matching and 3D transform considered as an analysis frame. The problem can then be decoupled into two steps: A regularized inversion of the blurring and threshold-based denoising. These two steps are repeated iteratively until convergence is reached. This process is referred to as iterative decoupled deblurring BM3D (IDD-BM3D).

If the images are modeled before IDD-BM3D is performed, it is reasonable to expect an improvement in performance because we will effectively be reducing the amount of noise in the blurred image. To test this, we degraded the images in the Sigma dataset with both a moderate (25 point Gaussian with $\sigma = .4$) and a large (25 point Gaussian with $\sigma = 1.6$) amount of blur and added white Guassian noise

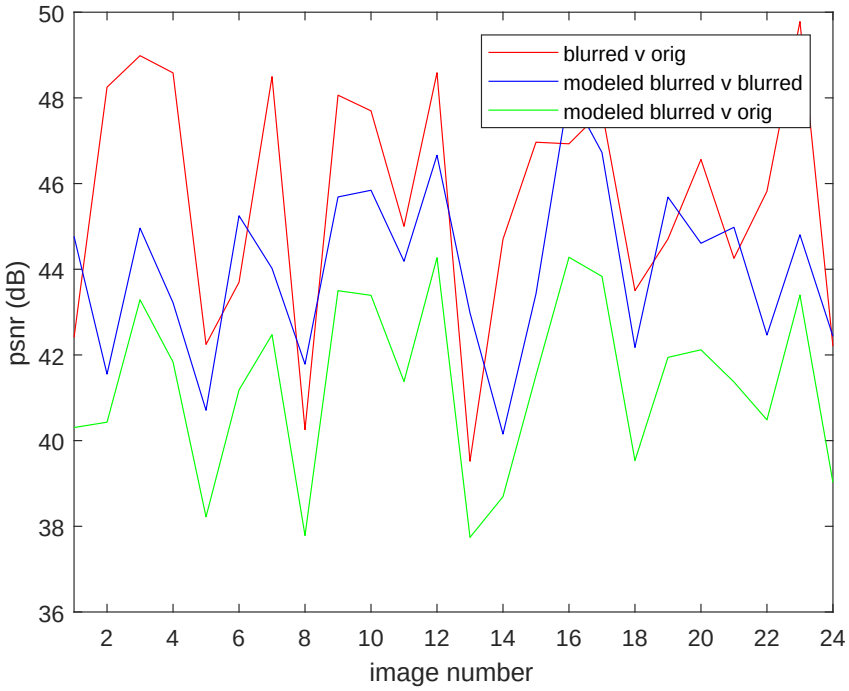


Figure 7-15: Results of imposing the modeling constraints on the blurred images of the Kodak dataset.

($\sigma = 8/255$, where the images were normalized to lie in the range of $0-1$). The images were then modeled and both the degraded and the modeled images were processed with IDD-BM3D. For computational reasons, the images were downsampled by a factor of two. Results are shown in Fig. 7-16 and Fig. 7-17.

From these figures, it can be seen that imposing the modeling constraints when there is a moderate amount of blurring provides a substantial advantage over running the spatial based deblurring system alone. However, that advantage falls dramatically when the amount of blurring is increased. This is possibly because modeling itself causes errors in the blurred image. For example, blurred edges are ambiguous and difficult to detect, so edge errors are expected to be significant. These errors are not really of the same form as Gaussian noise, which the denoising step in the deblurring system is designed to deal with. They are also in the high frequencies, which must somehow be amplified by the denoising system. Therefore, the error introduced by modeling becomes itself a significant source of error in the restored image.

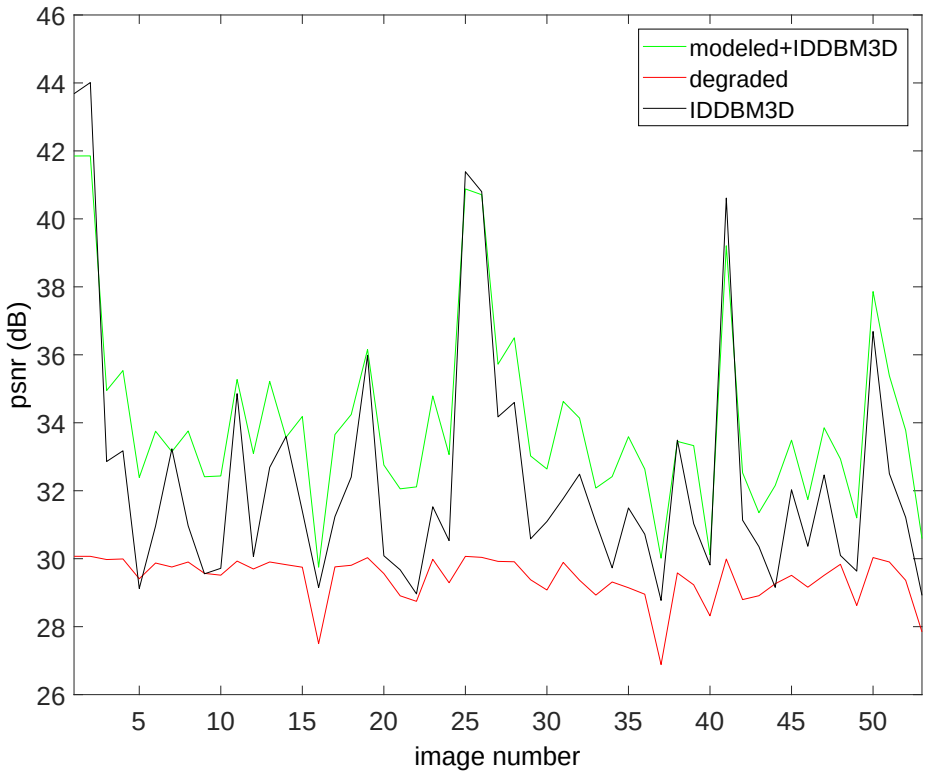


Figure 7-16: Results of imposing the modeling constraints on the blurred and noisy images of the Sigma dataset, then performing IDD-BM3D when there was a moderate amount of blurring.

If this is the case, we would expect two things. First, that as blurring (and therefore necessary high-frequency amplification) increases, the effect of modeling errors would also increase, and the advantage of modeling would decrease, as we have just seen.

Second, as noise increases, we would expect the advantage due to modeling to also increase, even as the quality of the restored image decreases. The more that the remaining white noise dominates the modeling error, the less effect those errors will have on the deblurring algorithm.

To test this, we took Sigma image 47 and blurred it with the wider of the two Gaussian functions and added noise of increasing variance. The image was then modeled and IDD-BM3D was performed on both the degraded image and the modeled image, as before. Results are shown in Fig. 7-18. When there was no noise, the error in

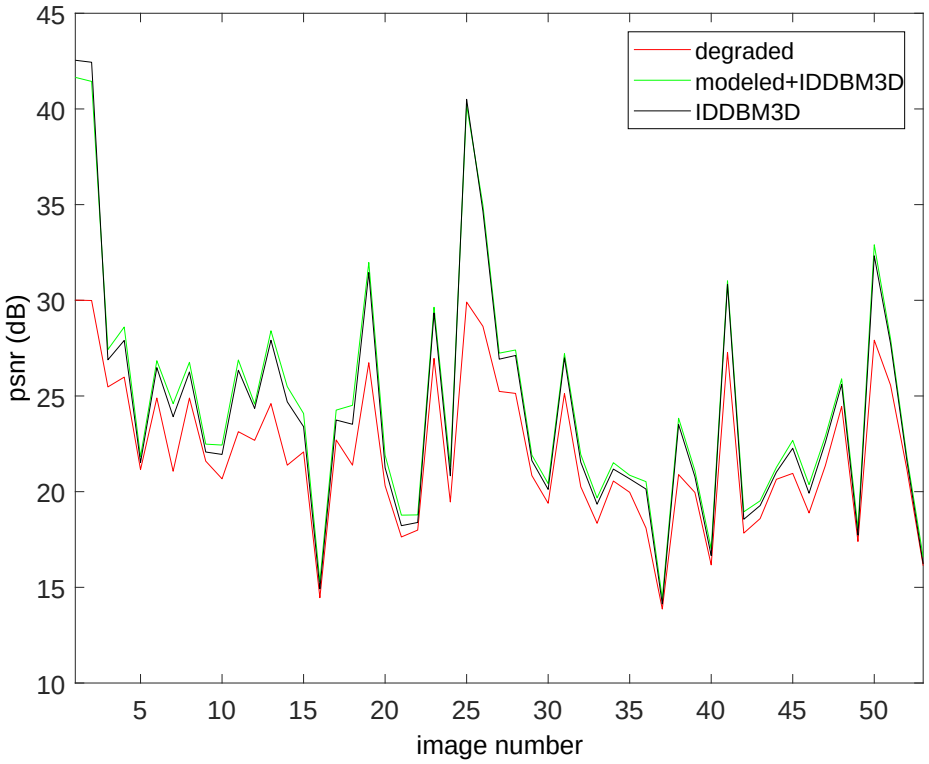


Figure 7-17: Results of imposing the modeling constraints on the blurred and noisy images of the Sigma dataset, then performing IDD-BM3D when there was a large amount of blurring.

the image restored by modeling and then IDD-BM3D had a PSNR of about 23.4dB, so this can be thought of as a rough cap on the possible quality of the image restored by modeling and IDD-BM3D. As noise increases, this cap allows for greater and greater improvement over the result obtained by IDD-BM3D, while at low noise levels, the cap actually causes a performance decrease. This is as expected.

7.6 Discussion

In this chapter, it has been shown that imposing the modeling constraints tends to improve images in a variety of restoration settings. This is as expected. The modeling constraints articulate the spectral relationships within an image, and taking advantage of these relationships and the redundancy they imply should lead to more

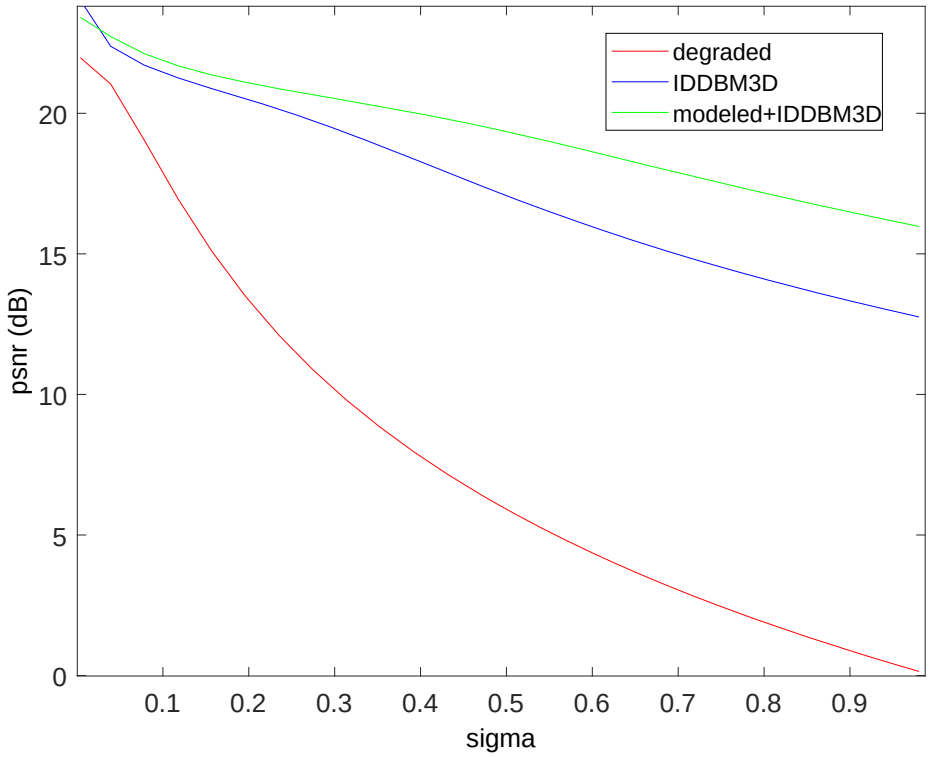


Figure 7-18: Results of imposing the modeling constraints on Sigma image 47 after degradation with heavy blurring and varying amounts of noise and before performing IDD-BM3D.

accurate image restoration. Note, however, that this model is not the only way to take advantage of those relationships, nor is enforcing the modeling constraints directly the only way the model could be used. For example, as we discussed at the beginning of the chapter, many image denoising methods start by computing the luminance channel as a less noisy estimate of the image. Specifically, this has been used in [9] to extend BM3D to a color denoisng method, CBM3D, which has been shown to achieve very good results. Nevertheless, direct application of the modeling constraints provides a straightforward and modular way to take advantage of spectral relationships, and may be desirable in some situations.

We also note that, while we have used BM3D as the spatial denoising system of choice in the examples in this chapter, the chrominance denoising system described here could be paired with WNNM [21], wavelet GSM [60], or any other spatial denois-

ing system. Similarly, in the case of blurred images, any spatial deblurring system could be used after chrominance-based denoising.

Chapter 8

Conclusion

In this thesis, it was shown that the color channels of an image can be represented as the product (or, in the logarithmic domain, the sum) of a color dependent local average and a color independent residual. This model was derived from physical image models and was also verified experimentally using three image datasets, and the limitations were discussed. Although there are situations where the model fails, it generally represents a wide array of images accurately. Furthermore, this model can be used to provide physical justifications for the observations that the chrominance channels tend to be low-frequency signals, the luminance channel tends to be less noisy, and the pixels of an image tend to form lines in RGB space. Finally, the effects of imposing the modeling constraints in various image processing situations were explored. It was found that imposing the modeling constraints often led to improved images, and that modeling could be paired with spatial processing for additional gains.

8.1 Future Directions

The derivation of the model proposed in this thesis did not require any assumptions on the spectral characteristics of the lightsource or the reflectance functions of the objects in the scene except that they were separable functions of position and wavelength. As mentioned in Chapter 3, there is empirical evidence that these functions are highly

structured, and it might be possible to take advantage of this structure to either strengthen the model by saying something about the relationship between the local averages or by bounding the modeling error when separability doesn't hold.

In addition, there may be applications of this image model other than those explored here. For example, it might be possible to use the model to remove salt and pepper noise by first identifying pixels that are in error and then excluding those pixels from the computation of the local average and the residual and modeling the missing data.

It is also possible that the model could be applied to image compression, though standard compression algorithms already take advantage of the low-frequency nature of the chrominance channels. Similarly, it's possible that the model could be applied to demosaicing, although similar models have already been used for this application in the past.

Finally, It is worth noting that, although this work has focused on RGB images, there is nothing in the theory of the model that limits it to such images. It may also be interesting to see how it could be applied to hyperspectral images, as well.

Bibliography

[1] Kodak lossless true color image suite. http://r0k.us/graphics/kodak/. Accessed: 11-29-16.

[2] F. J. Anscombe. The transformation of poisson, binomial and negative-binomial data. *Biometrika*, 35(3/4):246–254, 1948.

[3] J. F. Blinn. Models of light reflection for computer synthesized pictures. In *Siggraph*, pages 192–198, 1977.

[4] 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.

[5] P. Chatterjee and P. Milanfar. Patch-based near-optimal image denoising. *IEEE Trans. on Image Proc.*, 21(4):1635–1649, April 2012.

[6] J. Cohen. Dependency of the spectral reflectance curves of the munsell color chips. *Psychon Sci*, 1:369–370, 1964.

[7] R. L. Cook and K. E. Torrance. A reflectance model for computer graphics. *ACM Trans. on Graphics*, 1(1):7–24, January 1982.

[8] D. C. Costa, C. A. B. Mello, and T.J. dos Santos. Boundary detection based on chromatic color difference and morphological texture suppression. In *IEEE International Conference on Systems, Man, and Cybernetics*, pages 4305–4310, 2013.

[9] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian. Color image denoising via sparse 3d collaborative filtering with grouping constraint in luminance-chrominance space. In *IEEE International Conference on Image Proc.*, volume 1, pages 313–316, San Antonio, TX, 2007.

[10] 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.

[11] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *CVPR*, July 2005.

[12] A. Danielyan, V. Katkovnik, and K. Egiazarian. Bm3d frames and variational image deblurring. *IEEE Trans. on Image Proc.*, 21(4):1715–1728, 2012.

[13] J. L. Dannemiller. Spectral reflectance of natural objects: how many basis functions are necissary? *Journal of the Optical Society of America*, 9(4):507–515, April 1992.

[14] 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.

[15] E. R. Dixon. Spectral distribution of australian daylight. *Journal of the Optical Society of America*, 68(4):437–450, April 1978.

[16] S. Farsiu, M. Elad, and P. Milanfar. Multiframe demosaicing and super-resolution of color images. *IEEE Trans. on Image Proc.*, 15(1):141–159, January 2006.

[17] R. Fergus, B. Singh, A. Hertzmann, S. T. Roweis, and W. T. Freeman. Removing camera shake from a single photograph. *ACM Trans. Graph.*, 25(3):787–794, July 2006.

[18] 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.

[19] B. V. Funt, M. S. Drew, and M. Brockington. Recovering shading from color images. In *ECCV*, pages 124–132, 1992.

[20] T. Gevers and A. W. M. Smeulders. Color-based object recognition. *Pattern Recognition*, 32:453–464, 1999.

[21] S. Gu, L. Zhang, W. Zuo, and X. Feng. Weighted nuclear norm minimization with application to image denoising. In *CVPR*, June 2014.

[22] B. W. Hapke. A theoretical photometric function for the lunar surface. *Journal of Geophysical Research*, 68(15):4571–4586, August 1963.

[23] 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.

[24] X. D. He, K. E. Torrance, F. X. Sillion, and D. P. Greenberg. A comprehensive physical model for light reflection. *Computer Graphics*, 25(4):175–186, July 1991.

[25] G. Healey. Segmenting images using normalized color. *IEEE Trans. on Systems, Man, and Cybernetics*, 22(1):64–73, January/February 1992.

[26] B. K. P. Horn. Determining lightness from an image. *Computer Graphics and Image Processing*, 3(4):277–299, December 1974.

[27] T. Jaaskelainen, J. Parkkinen, and S. Toyooka. Vector-subspeace model for color representation. *Journal of the Optical Society of America*, 7(4):725–730, April 1990.

[28] M. K. Johnson and H. Farid. Exposing digital forgeries through chromatic aberration. In *Proceedings of the 8th Workshop on Multimedia and Security*, pages 48–55, 2006.

[29] N. Joshi, C. L. Zitnick, R. Szeliski, and D. J. Kriegman. Image deblurring and denoising using color priors. In *CVPR*, June 2009.

[30] D. B. Judd, D. L. MacAdam, and G. Wyszecki. Spectral distribution of typical daylight as a function of correlated color temperature. *Journal of the Optical Society of America*, 54(8):1031–1040, August 1964.

[31] J. T. Kajiya. Anisotropic reflection models. In *Siggraph*, pages 15–21, 1985.

[32] 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.

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

[34] 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.

[35] G. J. Klinker, S. A. Shafer, and T. Kanade. A physical approach to color image understanding. *International Journal of Computer Vision*, 4:7–38, 1990.

[36] E. H. Land and J. J. McCann. Lightness and retinex theory. *Journal of the Optical Society of America*, 61(1):1–11, 1971.

[37] A. Levin, D. Lischinski, and Y. Weiss. A closed-form solution to natural image matting. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 30(2):228–242, February 2008.

[38] A. Levin and B. Nadler. Natural image denoising: Optimality and inherent bounds. In *CVPR*, June 2011.

[39] C. Liu, R. Szeliski, S. B. Kang, C. L. Zitnick, and W. T. Freeman. Automatic estimation and removal of noise from a single image. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 30(2):299–314, February 2008.

[40] 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.

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

[42] M. Makitalo and A. Foi. Optimal inversion of the anscombe transformation in low-count poisson image denoising. *IEEE Trans. on Image Proc.*, 20(1):99–109, January 2011.

[43] L. T. Maloney. Evaluation of linear models of surface spectral reflectance with small numbers of parameters. *Journal of the Optical Society of America*, 3(10):1673–1683, October 1986.

[44] 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*, 2006.

[45] J. J. McCann and A. Rizzi. Camera and visual veiling glare in hdr images. *Journal of the Society for Information Display*, 15(9):721–730, 2007.

[46] P. Milanfar. A tour of modern image filtering. *IEEE Signal Processing Magazine*, 30(1):106–128, 2013.

[47] J. M. Morel, A. B. Petro, and C. Sbert. A pde formalization of retinex theory. *IEEE Trans. on Image Proc.*, 19(11):2825–2837, 2010.

[48] F. E. Nicodemus. Directional reflectance and emissivity of an opaque surface. *Applied Optics*, 4(7):767–775, July 1965.

[49] F. E. Nicodemus. Reflectance nomenclature and directional reflectance and emissivity. *Applied Optics*, 9(6):1474–17475, June 1970.

[50] F. E. Nicodemus, J. C. Richmond, J. J. Hsia, I. W. Ginsberg, and T. Limperis. *Geometrical considerations and nomenclature for reflectance*. US Department of Commerce, National Bureau of Standards, 1977.

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

[52] A. V. Oppenheim. Generalized superposition. *Information and Control*, 11:528–539, November/December 1967.

[53] M. Oren and S. K. Nayar. Diffuse reflectance from rough surfaces. In *IEEE computer society conference on computer vision and pattern recognition*, pages 763–764, 1993.

[54] M. Oren and S. K. Nayar. Generalization of lambert's reflectance model. In *Proc. of the 21st annual conference on computer graphics and interactive techniques*, pages 239–246, 1994.

[55] M. Oren and S. K. Nayar. Seeing beyond lambert's law. In *ECCV '94*, volume 2, pages 269–280, Stockholm, Sweden, May 1994.

[56] M. Oren and S. K. Nayar. Generalization of the lambertian model and implications for machine vision. *International journal of computer vision*, 14:227–251, 1995.

[57] J. P. S. Parkkinen, J. Hallikainen, and T. Jaaskelainen. Characteristic spectra of munsell colors. *Journal of the Optical Society of America*, 6(2):318–322, February 1989.

[58] B. T. Phong. Illumination for computer generated pictures. *Communications of the ACM*, 18(6):311–317, June 1975.

[59] 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.

[60] J. Portilla, V. Strela, M. J. Wainwright, and E. P. Simoncelli. Image denoising using scale mixtures of gaussians in the wavelet domain. *IEEE Trans. on Image Proc.*, 12(11):1338–1351, November 2003.

[61] D. L. Ruderman. The statistics of natural images. *Network: Computation in Neural Systems*, 5(4):517–548, July 1994.

[62] E. P. Simoncelli and B. A. Olshausen. Natural image statistics and neural representation. *Annual Review of Neuroscience*, 24:1193–1216, 2001.

[63] 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.

[64] PF Staff. How pentax pixel shifting works. http://www.pentaxforums.com/articles/photo-articles/how-pentax-k-3-ii-pixel-shifting-works.html, 2015. Accessed: 11-29-16.

[65] 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.

[66] H. Takeda, S. Farsiu, and P. Milanfar. Kernel regression for image processing and reconstruction. *IEEE Trans. on Image Proc.*, 16(2):349–366, January 2007.

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

[68] K. E. Torrance and E. M. Sparrow. Theory for off-specular reflection from roughened surfaces. *Journal of the optical society of America*, 57(9):1105–1114, 1967.

[69] S. Tozza and M. Falcone. Analysis and approximation of some shape-from-shading models for non-lambertian surfaces. *Journal of mathematical imaging and vision*, 55:153–178, 2016.

[70] T. S. Trowbridge and K. P. Reitz. Average irregularity representation of a rough surface for ray reflection. *Journal of the Optical Society of America*, 65(5):531–536, May 1975.

[71] 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.

[72] L. I. Voicu, H. R. Myler, and A. R. Weeks. Practical considerations on color image enhancement using homomorphic filtering. *Journal of Electronic Imaging*, 6(1):108–113, January 1997.

[73] G. J. Ward. Measuring and modeling anisotropic reflection. *Computer Graphics*, 26(2):265–272, July 1992.

[74] S. Wesolkowski, M. E. Jernigan, and R. D. Dony. Comparison of color image edge detectors in multiple color spaces. In *IEEE International Conference on Image Proc.*, pages 796–799, 2000.

[75] T. Whitted. An improved illumination model for shaded display. *Communications of the ACM*, 23(6):343–349, June 1980.

[76] 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.

Appendix A

Image Datasets

Figure A-1: Thumbnails of figures 1-20 from the Sigma dataset. Image 1 is in the top left corner and numbering continues left to right, then top to bottom.

Figure A-2: Thumbnails of figures 21-48 from the Sigma dataset.

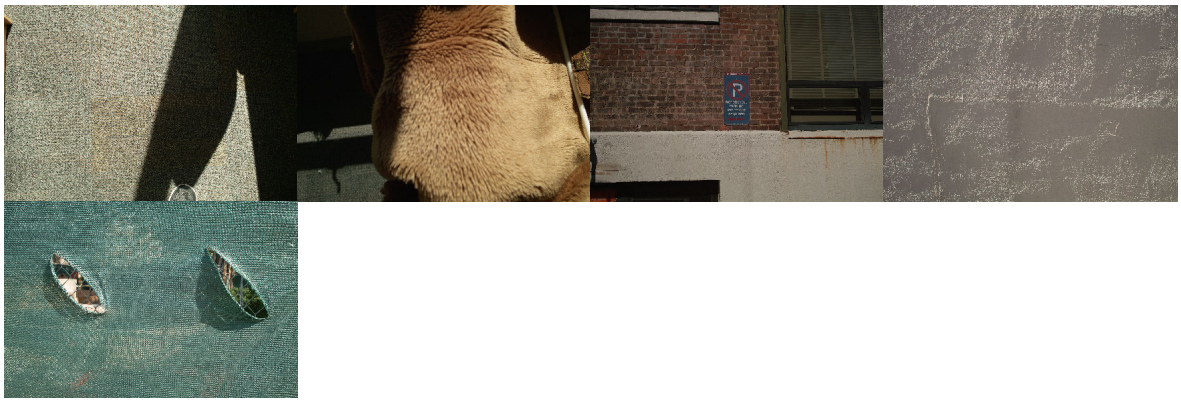


Figure A-3: Thumbnails of figures 49-53 from the Sigma dataset.

Figure A-4: Thumbnails of figures 1-20 from the Pentax dataset.

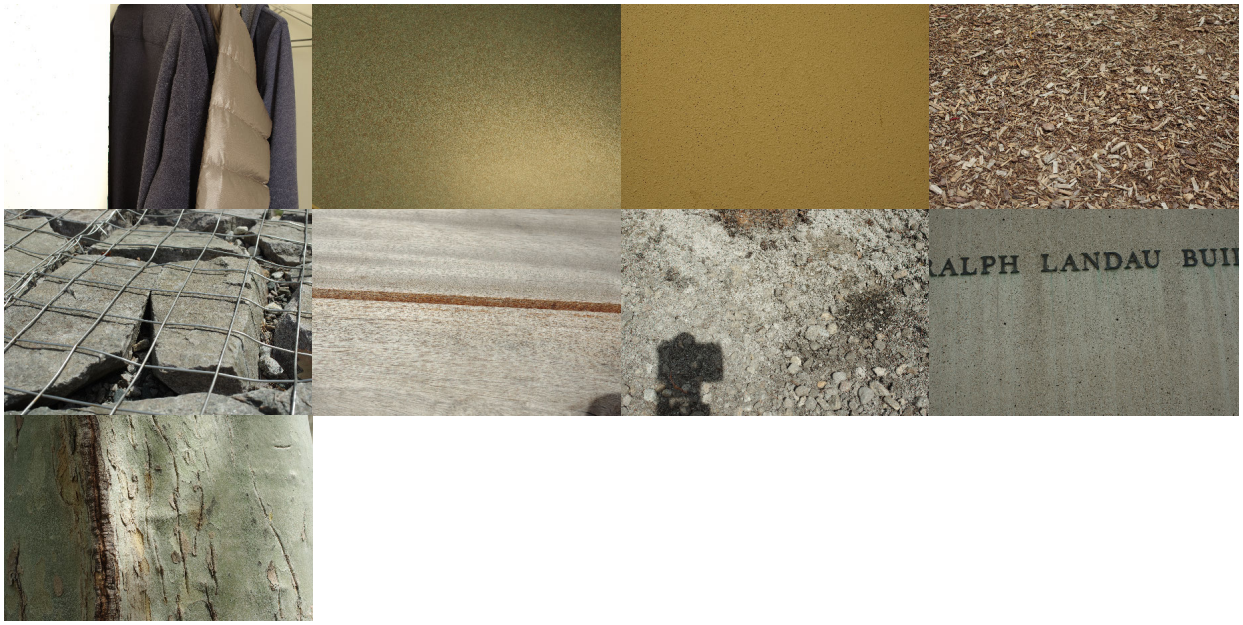


Figure A-5: Thumbnails of figures 21-29 from the Pentax dataset.

Figure A-6: Thumbnails of figures 1-24 from the Kodak dataset.

Appendix B

Derivation of the Theoretical Limits of Noise Reduction

The general equation for the red channel of a noisy image that has had the modeling constraints imposed is

$$\hat{R}'(x) = R'_{avg}(x) + \frac{1}{3}\left(\tilde{R}'(x) - R'_{avg}(x) + \tilde{G}'(x) - G'_{avg}(x) + \tilde{B}'(x) - B'_{avg}(x)\right) \quad \text{(B.1)}$$

$$= \frac{2}{3}R'_{avg}(x) + \frac{1}{3}\left(\tilde{R}'(x) + \tilde{G}'(x) - G'_{avg}(x) + \tilde{B}'(x) - B'_{avg}(x)\right) \quad \text{(B.2)}$$

Assuming the noise is additive, we have

$$\tilde{R}'(x) = R'(x) + n_R(x) \quad \text{(B.3)}$$

$R'_{avg}(x)$ is computed as

$$R'_{avg}(x) = \frac{1}{|W|}\sum_{k\in W}\tilde{R}'(x+k) \quad \text{(B.4)}$$

$$= \frac{1}{|W|}\sum_{k\in W}R'(x+k) + n_R(x+k) \quad \text{(B.5)}$$

where W is the set of pixels over which the average is taken. If it is further assumed that all the image values are the same (a best-case scenario for noise reduction via

modeling), this becomes

$$R'_{avg}(x) = R'(x) + \frac{1}{|W|}\sum_{k\in W} n_R(x+k) \tag{B.6}$$

Substituting (B.6) and (B.3) into (B.2):

$$\hat{R}'(x) = \frac{2}{3}\left(R'(x) + \frac{1}{|W|}\sum_{k\in W} n_R(x+k)\right) + \frac{1}{3}(R'(x) + n_R(x) + G'(x) + n_G(x) - (G'(x) + \frac{1}{|W|}\sum_{k\in W} n_G(x+k)) + B'(x) + n_B(x) - (B'(x) + \frac{1}{|W|}\sum_{k\in W} n_B(x+k))) \tag{B.7}$$

After some algebraic manipulation, this becomes

$$\hat{R}'(x) = R'(x) + \frac{2}{3|W|}\sum_{k\in W} n_R(x+k) - \frac{1}{3|W|}\sum_{k\in W} n_G(x+k) - \frac{1}{3|W|}\sum_{k\in W} n_B(x+k) + \frac{1}{3}n_R(x) + \frac{1}{3}n_G(x) + \frac{1}{3}n_B(x) \tag{B.8}$$

We are now interested in calculating the expected squared error:

$$\mathbb{E}\left[\left(\hat{R}'(x) - R'(x)\right)^2\right] = \mathbb{E}\left[\left(\hat{R}'(x)\right)^2 - 2\hat{R}'(x)R'(x) + (R'(x))^2\right] \tag{B.9}$$

$$= \mathbb{E}\left[\left(\hat{R}'(x)\right)^2\right] - 2R'(x)\mathbb{E}\left[\hat{R}'(x)\right] + (R'(x))^2 \tag{B.10}$$

Where in going from (B.9) to (B.10) we have used the linearity of expectation and the fact that the original image is a deterministic signal. Let us break up (B.10) and

examine a piece at a time. First, we compute

$$\begin{aligned}
\mathbb{E}\left[\hat{R}'(x)\right] &= \mathbb{E}[R'(x) + \frac{2}{3|W|}\sum_{k\in W} n_R(x+k) - \frac{1}{3|W|}\sum_{k\in W} n_G(x+k) - \\
&\quad \frac{1}{3|W|}\sum_{k\in W} n_B(x+k) + \frac{1}{3}n_R(x) + \frac{1}{3}n_G(x) + \frac{1}{3}n_B(x)] \\
&= \mathbb{E}[R'(x)] + \frac{2}{3|W|}\sum_{k\in W} \mathbb{E}[n_R(x+k)] - \frac{1}{3|W|}\sum_{k\in W} \mathbb{E}[n_G(x+k)] - \\
&\quad \frac{1}{3|W|}\sum_{k\in W} \mathbb{E}[n_B(x+k)] + \frac{1}{3}\mathbb{E}[n_R(x)] + \frac{1}{3}\mathbb{E}[n_G(x)] + \frac{1}{3}\mathbb{E}[n_B(x)] \\
&= R'(x)
\end{aligned}$$

Where the second step is due to the linearity of expectation and the final step is due to the fact that the noise is zero mean and the original image is deterministic. Substituting this back into (B.10):

$$\mathbb{E}\left[\left(\hat{R}'(x) - R'(x)\right)^2\right] = \mathbb{E}\left[\left(\hat{R}'(x)\right)^2\right] - 2(R'(x))^2 + (R'(x))^2 \tag{B.11}$$

$$= \mathbb{E}\left[\left(\hat{R}'(x)\right)^2\right] - (R'(x))^2 \tag{B.12}$$

Let us now examine $\mathbb{E}\left[\left(\hat{R}'(x)\right)^2\right]$. Using (B.8), we have

$$\begin{aligned}
\mathbb{E}\left[\left(\hat{R}'(x)\right)^2\right] &= \mathbb{E}[(R'(x) + \frac{2}{3|W|}\sum_{k\in W} n_R(x+k) - \frac{1}{3|W|}\sum_{k\in W} n_G(x+k) - \\
&\quad \frac{1}{3|W|}\sum_{k\in W} n_B(x+k) + \frac{1}{3}n_R(x) + \frac{1}{3}n_G(x) + \frac{1}{3}n_B(x))^2]
\end{aligned} \tag{B.13}$$

Exploiting the linearity of expectation, the deterministic nature of $R'(x)$, the zero-mean nature of the noise, and the fact that the noise is independent across color

channels as we expand, we find:

$$\mathbb{E}\left[\left(\hat{R}'(x)\right)^2\right] = (R'(x))^2 + \mathbb{E}\left[\left(\frac{2}{3|W|}\sum_{k\in W} n_R(x+k)\right)^2\right] + \mathbb{E}\left[\frac{2}{3}n_R(x)\frac{2}{3|W|}\sum_{k\in W} n_R(x+k)\right]$$
$$+\mathbb{E}\left[\left(\frac{1}{3|W|}\sum_{k\in W} n_G(x+k)\right)^2\right] - \mathbb{E}\left[\frac{2}{3}n_G(x)\frac{1}{3|W|}\sum_{k\in W} n_G(x+k)\right] + \mathbb{E}\left[\left(\frac{1}{3|W|}\sum_{k\in W} n_B(x+k)\right)^2\right]$$
$$-\mathbb{E}\left[\frac{2}{3}n_B(x)\frac{1}{3|W|}\sum_{k\in W} n_B(x+k)\right] + \mathbb{E}\left[\frac{1}{9}(n_R(x))^2\right] + \mathbb{E}\left[\frac{1}{9}(n_G(x))^2\right] + \mathbb{E}\left[\frac{1}{9}(n_B(x))^2\right] \tag{B.14}$$

Before continuing, it is worth pointing out a slight subtlety regarding the independence of the noise. It is important for this analysis not only that the noise samples in the image are independent, but that they are independent *conditional on the fact that they are in* W. It is possible to design W such that this is not true. For example, if all pixels in W were required to have the same value and the underlying image was known to be constant, the noise values would be known to all be the same, and obviously they would not be independent. A less dramatic example is a W where all the pixel values are required to be within some threshold of each other. In practice, this is a useful way to design the set. Such a design can help W avoid including pixels on both sides of an edge, for example. However, for this analysis, we are assuming the underlying image is constant. If this is known, there is no reason to set a threshold, and doing so would make the analysis intractable. Therefore, we will continue to assume independence of the noise.

Now, let us exploit the spatial independence of the noise to simplify (B.14) further:

$$\mathbb{E}\left[\left(\hat{R}'(x)\right)^2\right] = (R'(x))^2 + \frac{4}{9|W|^2}\sum_{k\in W}\mathbb{E}\left[(n_R(x+k))^2\right] + \frac{4}{9|W|}\mathbb{E}\left[(n_R(x))^2\right] +$$
$$\frac{1}{9|W|^2}\sum_{k\in W}\mathbb{E}\left[(n_G(x+k))^2\right] - \frac{2}{9|W|}\mathbb{E}\left[(n_G(x))^2\right] + \frac{1}{9|W|^2}\sum_{k\in W}\mathbb{E}\left[(n_B(x+k))^2\right]$$
$$-\frac{2}{9|W|}\mathbb{E}\left[(n_B(x))^2\right] + \frac{1}{9}\mathbb{E}\left[(n_R(x))^2\right] + \frac{1}{9}\mathbb{E}\left[(n_G(x))^2\right] + \frac{1}{9}\mathbb{E}\left[(n_B(x))^2\right] \tag{B.15}$$

Because the noise is identically distributed with variance σ_n^2 in both space and color channel, this can be simplified further to find that

$$\begin{aligned}
\mathbb{E}\left[\left(\hat{R}'(x)\right)^2\right] &= (R'(x))^2 + \frac{4}{9|W|^2}|W|\sigma_n^2 + \frac{4}{9|W|}\sigma_n^2 + \frac{1}{9|W|^2}|W|\sigma_n^2 - \frac{2}{9|W|}\sigma_n^2 + \frac{1}{9|W|^2}|W|\sigma_n^2 \\
&\quad - \frac{2}{9|W|}\sigma_n^2 + \frac{1}{9}\sigma_n^2 + \frac{1}{9}\sigma_n^2 + \frac{1}{9}\sigma_n^2 \\
&= (R'(x))^2 + \frac{1}{|W|}\sigma_n^2\left(\frac{4}{9} + \frac{4}{9} + \frac{1}{9} - \frac{2}{9} + \frac{1}{9} - \frac{2}{9}\right) + \sigma_n^2\left(\frac{1}{9} + \frac{1}{9} + \frac{1}{9}\right) \\
&= (R'(x))^2 + \sigma_n^2\left(\frac{2}{3|W|} + \frac{1}{3}\right)
\end{aligned}$$

Substituting this result back into B.12:

$$\mathbb{E}\left[\left(\hat{R}'(x) - R'(x)\right)^2\right] = (R'(x))^2 + \sigma_n^2\left(\frac{2}{3|W|} + \frac{1}{3}\right) - (R'(x))^2 \tag{B.16}$$

$$= \sigma_n^2\left(\frac{2}{3|W|} + \frac{1}{3}\right) \tag{B.17}$$

Let us consider (B.17). Several things can be determined from this. First, the expected squared error is proportional to the variance of the original noise. We would like to make the constant of proportionality as small as possible. As we increase $|W|$, that is, as we include more and more pixels in the local average and so get a better and better estimate of it, that multiplier decreases. In the limit, it becomes $\frac{1}{3}$. This term is due to the fact that there are only three color channels being averaged. If we were dealing with a hyperspectral image, it could go down, but it is set for an RGB image. Therefore, we see that chrominance based denoising will decrease the noise variance by at most three.

To give a sense of some realistic numbers, if the averaging was done using a 3x3 pixel window, $|W| = 9$ and the multiplier would be .4074, leading to an expected PSNR improvement of 3.90dB. If a 7x7 pixel window was used, the multiplier becomes .3469, a PSNR improvement of 4.60dB. The limit of an infinite pixel window has a multiplier of .3333, a PSNR improvement of 4.77dB. In the experiments presented in this thesis, we have used a 7x7 pixel window.