IMAGE PROCESSING USING SENSOR NOISE AND
HUMAN VISUAL SYSTEM MODELS

A Dissertation

Presented to the Faculty of the Graduate School
of Cornell University
in Partial Fulfillment of the Requirements for the Degree of
Doctor of Philosophy

by
Keigo Hirakawa
May 2005
Because digital images are subject to noise in the device that captured them and the human visual system (HVS) that observes them, it is important to consider accurate models for noise and the HVS in the design of image processing methods. In this thesis, CMOS image sensor noise is characterized, the chromatic adaptation theories are reviewed, and new image processing algorithms that address these noise and HVS models are presented.

First, a method for removing additive, multiplicative, and mixed noise from an image is developed. An image patch from an ideal image is modeled as a linear combination of image patches from the noisy image. This image model is fit to the image data in the total least square (TLS) sense, because it allows uncertainties in the measured data. The image quality of the output image demonstrates the effectiveness of the TLS algorithms and improvement over existing methods.

Second, we develop a novel technique to combine demosaicing and denoising procedures systematically into a single operation. We first design a filter as optimally estimating a pixel value from a noisy single-color image. With additional constraints, we show that the same filter coefficients are appropriate for demosaicing noisy sensor data. The proposed technique can combine many existing denoising algorithms with the demosaicing operation. The algorithm is tested with pseudo-random noise and noisy raw sensor data from a real digital camera, and the proposed method suppresses CMOS image sensor noise while effectively interpolating the missing pixel components better than when treating demosaicing
and denoising problems independently.

Third, the problem of adjusting the color to match the digital camera output with the scene observed by the photographer’s eye is called white-balance. While most existing white-balance algorithms combine the von Kries coefficient law and an illuminant estimation techniques, the coefficient law has been shown to be an inaccurate model. We instead formulate the problem using induced opponent response theory, the solution to which reduces to a single matrix multiplication. The experimental results verify that this approach yields more natural images than traditional methods. The computational cost of the proposed method is virtually zero.
Keigo Hirakawa received the B.S. degree in electrical engineering from Princeton University, Princeton, NJ, in 2000 and the M.S. degree in electrical and computer engineering from Cornell University, Ithaca, NY, in 2003. He simultaneously pursues an M.M. degree in jazz piano performance at New England Conservatory of Music, Boston, MA, and a PhD in electrical and computer engineering at Cornell University. His research interests include image modeling, image denoising, image interpolation, color science and human visual systems, and asynchronous signal processing.
To my family, old and new.
ACKNOWLEDGMENTS

First and foremost, I would like to express my most genuine gratitude to my research advisor, Professor Thomas W. Parks. He firmly believed that my most valuable strength is my creativity, and through his encouragement, he helped me to also understand this. I am deeply indebted to his invaluable professional and personal mentorship, his sincerity, and his warmth. He has made my experiences at Cornell singularly meaningful, and I humbly acknowledge the contributions Professor Parks has made to my life.

I would like to also thank my committee members, Professor Sheila Hemami, Professor David Delchamps, and Professor Robert Strichartz. I have benefitted immensely from their thoughtful advice and constructive feedbacks throughout my time at Cornell. In particular, Professor Hemami spent much time reviewing my work, and I appreciate her energy and her willingness to share her intellect.

I would also like to take the time to acknowledge Professor Gaurav Sharma at the University of Rochester, Professor Luca Lucchese at Oregon State University, and my colleagues, Karthik Raghupathy, Damon Chandler, Darian Muresan, and Matt Gaubatz, with whom I have consulted and interacted with regards to my Ph.D. research on countless occasions, and their enthusiasm and interest in my research always motivated me to keep moving forward. Tim Brown and my wonderful friends at Agilent Technologies also deserve great recognition. They are responsible for nourishing my appetite for image processing problems, and their contributions to my research are most certainly reflected in this dissertation.

Conversations with Professor Bahadir Gunturk at Louisiana State University, Dr. Xuemei Zhang and Dr. Ramakrishna Kakarala at Agilent Technologies, and Dr. Hsien-Che Lee deepened my understanding of the problems I faced in my
research. Their suggestions affected the research contained herein.

I am indebted to my friends and family, whose love and support have been with me from start to finish. I am very fortunate and grateful to have Adithya, Griff, Emily, Spencer, Eli, Paul, and Mr. Harold Danko in my life. I thank my mother, Michiko; my father, Hiroshi; my brother, Masaaki; and my fiancée, Wenbi Lai for their unconditional love, their emotional support, their patience, their helping hand, and their belief in the crazy life that I lead.

This research was partially supported by Texas Instruments, Agilent Technologies, and the Center for Electronic Imaging Systems. Digital camera evaluation boards were donated to our research group by Agilent Technologies and Texas Instruments.
<table>
<thead>
<tr>
<th>Chapter</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Introduction</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>1.1 Thesis Overview</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>1.2 Thesis Contributions</td>
<td>3</td>
</tr>
<tr>
<td>2</td>
<td>Background</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td>2.1 Color Science</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td>2.1.1 Notational Conventions</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td>2.1.2 Colorimetry</td>
<td>6</td>
</tr>
<tr>
<td></td>
<td>2.1.3 Opponent Color</td>
<td>9</td>
</tr>
<tr>
<td></td>
<td>2.1.4 Chromatic Adaptation</td>
<td>10</td>
</tr>
<tr>
<td></td>
<td>2.2 Image Acquisition Models</td>
<td>13</td>
</tr>
<tr>
<td></td>
<td>2.2.1 Digital Camera</td>
<td>13</td>
</tr>
<tr>
<td></td>
<td>2.2.2 Illuminant Estimation</td>
<td>16</td>
</tr>
<tr>
<td>3</td>
<td>CMOS Image Sensor Noise Characterization and Introduction to Image Denoising</td>
<td>19</td>
</tr>
<tr>
<td></td>
<td>3.1 Noise Characterization</td>
<td>19</td>
</tr>
<tr>
<td></td>
<td>3.2 Perceived Noise</td>
<td>20</td>
</tr>
<tr>
<td></td>
<td>3.3 Introduction to Image Denoising</td>
<td>23</td>
</tr>
<tr>
<td>4</td>
<td>Total Least Squares Method of Image Denoising</td>
<td>27</td>
</tr>
<tr>
<td></td>
<td>4.1 TLS Image Denoising Theory</td>
<td>27</td>
</tr>
<tr>
<td></td>
<td>4.1.1 TLS Image Model</td>
<td>28</td>
</tr>
<tr>
<td></td>
<td>4.1.2 TLS Solution using Noise Model</td>
<td>29</td>
</tr>
<tr>
<td></td>
<td>4.2 Enhancements to TLS Image Model</td>
<td>32</td>
</tr>
<tr>
<td></td>
<td>4.2.1 Affine Approximation</td>
<td>32</td>
</tr>
<tr>
<td></td>
<td>4.2.2 Weighted TLS and Image Patches</td>
<td>33</td>
</tr>
<tr>
<td></td>
<td>4.2.3 Multi-Resolution Image Patches</td>
<td>38</td>
</tr>
<tr>
<td></td>
<td>4.2.4 Redundant Estimation</td>
<td>40</td>
</tr>
<tr>
<td></td>
<td>4.2.5 Pre-Processing</td>
<td>42</td>
</tr>
<tr>
<td></td>
<td>4.3 Implementation and Results</td>
<td>42</td>
</tr>
<tr>
<td></td>
<td>4.3.1 Computational Complexity</td>
<td>52</td>
</tr>
<tr>
<td></td>
<td>4.4 Summary</td>
<td>55</td>
</tr>
<tr>
<td>5</td>
<td>Joint Demosaicing and Denoising</td>
<td>57</td>
</tr>
<tr>
<td></td>
<td>5.1 Filter Design</td>
<td>59</td>
</tr>
<tr>
<td></td>
<td>5.2 Denoising Method</td>
<td>63</td>
</tr>
<tr>
<td></td>
<td>5.2.1 TLS Denoising Problem</td>
<td>63</td>
</tr>
<tr>
<td></td>
<td>5.2.2 Solution to TLS</td>
<td>66</td>
</tr>
<tr>
<td></td>
<td>5.2.3 Denoising Improvements</td>
<td>68</td>
</tr>
<tr>
<td></td>
<td>5.2.4 Pre-Processing</td>
<td>69</td>
</tr>
<tr>
<td></td>
<td>5.3 Implementation and Results</td>
<td>70</td>
</tr>
</tbody>
</table>
5.4 Summary ....................................................... 76

6 Chromatic Adaptation and White-Balance Problem 77
6.1 White-Balance Algorithm .................................. 78
   6.1.1 White-Balance Problem ............................... 78
   6.1.2 Common Approaches to White-Balance ............... 81
   6.1.3 Jameson-Hurvich Model ............................... 82
6.2 Experimental Results ..................................... 85
6.3 Summary .................................................... 94

7 Conclusion .................................................. 96

A The Solution to TLS Problem ............................. 98

B Review of Generalized Homomorphic Filtering ............. 100

Bibliography .................................................. 102
## LIST OF TABLES

<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.1</td>
<td>Error in the assumption (4.10), measured as $|P - \mathcal{E}{P}|_2/|P|_2$. Matrix $P$ is defined as (4.8).</td>
</tr>
<tr>
<td>4.2</td>
<td>Error in the assumption (4.10), measured as $|P - \mathcal{E}{P}|_2/|P|_2$. Matrix $P$ is defined as (4.19).</td>
</tr>
<tr>
<td>4.3</td>
<td>Denoising methods evaluated using PSNR. Images corrupted by noise generated by $(k_0, k_1) = (25, 0), (25, 0.1), (25, 0.2)$, respectively. Method in [9] was combined with methods in [51] [49] [54] [45] to account for signal-dependency of noise (see text).</td>
</tr>
<tr>
<td>4.4</td>
<td>Denoising methods evaluated using SSIM. Images corrupted by noise generated by $(k_0, k_1) = (25, 0), (25, 0.1), (25, 0.2)$, respectively. Method in [9] was combined with methods in [51] [49] [54] [45] to account for signal-dependency of noise (see text).</td>
</tr>
<tr>
<td>5.1</td>
<td>Performance of demosaicing and denoising algorithms on the “parrots” image, evaluated using average SCIELAB error [75]. Noise levels considered were $(k_0, k_1) = (0, 0), (25, 0), and (10, 0.1)$. n/a means not available or not necessary.</td>
</tr>
</tbody>
</table>
LIST OF FIGURES

2.1 Opponent color space. ............................................. 9
2.2 Typical digital camera image processing pipeline. .............. 13
2.3 With no color space conversion (left), with color space conversion (right). .................................................. 15

3.1 Graphs of standard deviation of noise v.s. image sensor value (for a fixed PGA). From top-to-bottom, left-to-right, green sensors, red sensors, blue sensors, green sensors. ................................................. 21
3.2 The histogram of noise when $X = 160$. .......................... 21
3.3 $(X + (k_0 + k_1 X))^{1/3} - X^{1/3}$ plotted against $X$. See text. .... 22
3.4 Denoising examples. (a) Noisy image $(k_0, k_1) = (25, 0)$. (b) Low-pass filtering. (c) Median filtering. (d) Bilateral filtering. (e) Wiener filtering. (f) Wavelet thresholding. .......................... 24

4.1 Example demonstrating the effectiveness of adaptive weights. Noisy input image cropped from “Lena” (left), output image without adaptive weights (middle), output image with adaptive weights (right). ................................................................. 38
4.2 Image patches from the noisy image in the spatial vicinity of $s_0$. 39
4.3 Image patches from decimated noisy image in the spatial vicinity of $s_0$. ................................................................. 39
4.4 Example demonstrating the sharpening of edges due to the inclusion of decimated image patches. Noisy input image cropped from “Lena” (left), output image using $\{x_i^{(1)}\}$ only (middle), output image using $\{x_i^{(1)}\}$ and $\{x_i^{(2)}\}$ (right). ................................................................. 40
4.5 Noise $(k_0, k_1) = (25, 0.2)$ and average error magnitude as a function of signal value. Error magnitude in the proposed algorithm is signal-independent. ................................................................. 43
4.6 Example with signal-independent noise, $k_0 = 25$. First column: noisy input image, and output from proposed algorithm, and output from method in [51]. Second column: output from method in [49], output from method in [54], and output from method in [45]. 44
4.7 Example with signal-independent noise, $k_0 = 50$. First column: noisy input image, output from proposed algorithm, and output from method in [51]. Second column: output from method in [49], output from method in [54], and output from method in [45]. .... 45
4.8 Example with signal-dependent noise, $k_0 = 15, k_1 = 0.4$. First column: noisy input image, output from proposed algorithm, and output from method in [51] (with [9]). Second column: output from method in [49] (with [9]), output from method in [54] (with [9]), and output from method in [45] (with [9]). .......................... 46
4.9 SSIM for images denoised using fewer image patches by omitting image patches that are poor representations of \( s_0 \). Noise generated with \((k_0, k_1) = (25, 0)\). Circle represents ‘lena’ image, triangle represents ‘barbara’ image. See text. .............................................. 55

5.1 An example of \( n \times n \) window, \( y_0 \) cropped from noisy sensor output. Cartesian coordinate is used to label the pixel locations, and \( y_0(0, 0) \) is the center pixel. .......................................................... 60

5.2 Red image (left), green image (middle), difference image (right). 61

5.3 Example using “parrot” image with noise \((k_0, k_1) = (25, 0)\): reconstruction using method in [16], method in [20], method in [16] and [49], method in [20] and [49], method in [16] and [51], method in [20] and [51], method in [16] and [21], method in [20] and [21], proposed method. ............................................. 72

5.4 Example using “parrot” image with noise \((k_0, k_1) = (10, 0.1)\): reconstruction using method in [16], method in [20], method in [16] and [21], method in [20] and [21], proposed method. ......................... 73

5.5 Example using an image captured with Agilent Technologies’ CMOS APS digital camera: method in [16], method in [20], proposed method. ................................................................. 74

6.1 Two different viewing conditions. The purpose of the white-balance algorithm is to make the two scenes appear identical. ......................... 80

6.2 Macbeth Color Chart used to calibrate the camera. ......................... 85

6.3 Images taken with no white-balance algorithms. In each respective scene, the solid colors represent illuminants detected by (left) gray-world [35] and (right) method in [13]. ......................... 86

6.4 Experiments using TI CCD camera. ............................................. 87

6.5 Experiments using TI CCD camera. ............................................. 88

6.6 Experiments using TI CCD camera. ............................................. 89

6.7 Experiments using TI CCD camera. ............................................. 90

6.8 Experiments using Agilent CMOS camera. .................................... 91

6.9 Experiments using Agilent CMOS camera. .................................... 92

6.10 \( \Delta_{ab} \) distance between the white panel and a neutral white. The numbers in abscissa correspond to the figure numbers. ......................... 94
CHAPTER 1
INTRODUCTION

1.1 Thesis Overview

It is becoming increasingly important to identify problems in digital cameras and develop long term strategies for improving the image quality, as the consumer use of digital cameras, driven by digital still cameras and cell phone camera markets, grows at an unprecedented rate. In 2003, for example, Sony introduced a four-color image sensor array, which uses red, green, blue, and emerald color filters to reduce the color reproduction errors—conventional digital cameras use only three colors (red, green, blue) [58]. The industry’s decision to sacrifice the spatial resolution for color fidelity reflects the reality that the shrinking photoreceptor size will eventually override the loss of resolution, while the evolving fabrication technology will not fix the problems with color reproduction. Likewise, the rapid decrease in the size of sensors and the growth of pixel count makes the color artifacts due to demosaicing algorithms insignificantly small, and enabled mosaicing-based cost-effective single-cameras to remain competitive against full-color pixel technology introduced by Foveon. However, the photoreceptors suffer from a low signal-to-noise ratio because each sensor receives less photons, and thus the low-light digital photography is still far from reality. Increasingly, the digital camera critics are evaluating the camera performance based on noise, rather than on the color artifacts.

Two key areas of problems in digital camera algorithm research today are sensor noise and white-balance. Sensor noise deteriorates the digital camera output image. In order to design an effective noise removal method, a precise characterization of noise is needed. However, while image denoising is an active field of general
research, most do not consider realistic noise models. In addition, because the images captured by digital cameras undergo a series of image processing algorithms, the effect of the noise is often too complicated to describe mathematically at the output. Thus it is impractical to design an image denoising method for images captured by a digital camera unless sensor noise is addressed as an integral part of the camera’s image processing pipeline.

Given \textit{a priori} knowledge that the end-user of the digital camera is a human eye, the \textit{appearance} of the output color is also critical. In order to compensate for the effects of the human visual system (HVS), it is important to understand the mechanics of the HVS first. In addition to assuming an inaccurate HVS model, most existing white-balance methods pose the color problem poorly. Perhaps this is a testimony to the disconnectedness between the engineering field and cognitive science. A white-balance method is ineffective unless it incorporates an accurate model for how the human eye interprets color.

This thesis develops new image processing techniques motivated by sensor noise characterization and the HVS modeling. The signal-dependent nature of the CMOS image sensor noise is studied and the noise issue is addressed explicitly as a part of digital camera image processing pipeline. While the research results presented are formulated from device-specific noise models, the approaches taken are widely applicable to general problems. Likewise, chromatic adaptation models derived from psychology experiments are reviewed, the white-balance problem is posed according to a viewing model, and the solution to the problem using the HVS models is presented.
1.2 Thesis Contributions

The following is a list of original contributions in this thesis.

**CMOS image sensor noise characterization** The dependency of noise to the signal in CMOS image sensor is studied. Using image processing techniques, we independently verify the noise model proposed and measured by hardware experts.

**Total least squares image denoising algorithm** A method for removing additive, multiplicative, and mixed noise from an image is developed. An image patch from an ideal image is modeled as a linear combination of image patches from the noisy image. This image model is fit to the image data in the total least square (TLS) sense, because it allows uncertainties in the measured data.

**A method to combine demosaicing and denoising method** We develop a novel technique to combine demosaicing and denoising procedures systematically into a single operation. We first design a filter as optimally estimating a pixel value from a noisy single-color image. With additional constraints, we show that the same filter coefficients are appropriate for demosaicing noisy sensor data. The method is general and many existing denoising algorithms can be combined with the demosaicing procedure.

**Joint demosaicing and denoising using total least squares** The TLS image denoising algorithm, optimized for CMOS sensor noise model, is combined with the demosaicing procedure. The combined algorithm estimates the pixel values from sparsely sampled noisy sensor data.
White-balance solution using induced opponent response theory The problem of adjusting the color to match the digital camera output with the scene observed by the photographers eye is called white-balance. We formulate the white-balance problem precisely using a viewing model. While most existing white-balance algorithms solve this problem by combine the von Kries coefficient law and an illuminant estimation techniques, the coefficient law has been shown to be an inaccurate model. We instead solve the problem using induced opponent response theory. This solution reduces to a single matrix multiplication, and the computational cost is virtually zero.

The thesis is organized as follows. Color science, basic digital camera image processing pipeline, and existing image denoising methods are reviewed in chapter 2. Methodology to characterize CMOS sensor noise is also described in chapter 2, and the noise appearance is discussed. TLS denoising method is derived in chapter 4, and experimental results for the CMOS sensor noise model as well as the popular signal-independent noise are shown. A general strategy to combine demosaicing and image denoising is derived in chapter 5. As a proof-of-concept, a new demosaicing/denoising algorithm using TLS denoising method is developed using this technique. Experiments are performed on color images with pseudo-random noise and on digital camera raw data, and the performance is compared to . Finally, white-balance problem is formulated using the viewing model and solved assuming induced opponent response theory in chapter 6.
CHAPTER 2
BACKGROUN

2.1 Color Science

In the following subsections, readers will be familiarized with the concepts behind color science. We also assume in this thesis that the readers are not familiar with color science and the psycho-visual experiments, and we will carefully examine relevant HVS models. In citing many psychology papers, the author does not intend to enter into debates about different HVS theories and models. Instead, we make use of the theoretical framework set forth by color scientists. Section 2.1.2 reviews basic math behind colorimetry. While colorimetry deals primarily with the acquisition of the color by our photoreceptor, section 2.1.3 and 2.1.4 discuss the low-level HVS image processing theories.

2.1.1 Notational Conventions

Let $\lambda \in \mathbb{R}$ represent frequency in the spectrum of light. Unless otherwise specified, the following notational conventions are used to talk about color science (i.e. chapters 4 and 5 do not follow these conventions):

<table>
<thead>
<tr>
<th>variable type</th>
<th>example</th>
<th>usage</th>
</tr>
</thead>
<tbody>
<tr>
<td>lower case</td>
<td>$l(\lambda)$</td>
<td>spectral distribution function ($\mathbb{R} \rightarrow \mathbb{R}$)</td>
</tr>
<tr>
<td>lower case with arrow</td>
<td>$\vec{u}$</td>
<td>vector in $\mathbb{R}^3$</td>
</tr>
<tr>
<td>upper case</td>
<td>$M$</td>
<td>$\mathbb{R}^3$ matrix</td>
</tr>
<tr>
<td>lower case greek</td>
<td>$\phi(\lambda)$</td>
<td>spectral analysis function ($\mathbb{R} \rightarrow \mathbb{R}$)</td>
</tr>
<tr>
<td>upper case greek</td>
<td>$\Phi(l)$</td>
<td>a map relating a spectral function to a vector $\mathbb{R}^3$</td>
</tr>
</tbody>
</table>
It is also implied that $j \in \{1, 2, 3\}$ is used to index the three color types, unless otherwise specified.

### 2.1.2 Colorimetry

*Colorimetry* is the science of measuring color. More complete details of colorimetry are found in [73]. Here, we cover only the basic equations necessary to understand the contents of the thesis.

**Cones and Rods**

There are four types of photoreceptors, one *rod* and three *cones*. Let the cone types be denoted by $L, M, S$. Cones are sensitive to the colors, and they are concentrated in the center region of the pupil. Rods are sensitive to the intensities of the light and movements, but insensitive to color. Rods are found in the outer region of the pupil. Under well-lit viewing conditions (photopic vision), cones are highly active and colors appear vivid to a human eye. In poorly-lit viewing condition (scotopic vision), rods are active, and colors are not very well perceived. In this thesis, we strictly assume photopic vision.

**Photoreceptor Model**

Let $l(\lambda) \in \mathbb{R}_+$ be the spectrum distribution of a color at frequency $\lambda \in \mathbb{R}$. Let $\phi_j(\lambda) \in \mathbb{R}_+, j \in \{1, 2, 3\}$ be the cone sensitivity function of $L, M, S$ photoreceptors, respectively. The cone sensitivity functions are centered around 550 nm ($L$), 540 nm ($M$), and 450 nm ($S$). Then the following inner product models the response vector $\Phi(l) = [\Phi_1(l), \Phi_2(l), \Phi_3(l)]^T \in \mathbb{R}^3$ of the cone photoreceptors to the spectrum
distribution $l(\cdot)$:

$$
\Phi_j(l) = \langle \phi_j, l \rangle = \int_{-\infty}^{\infty} \phi_j(\lambda) l(\lambda) d\lambda.
$$

If $\Phi(l_1) = \Phi(l_2)$ for given spectrum distributions $l_1(\cdot)$ and $l_2(\cdot)$, then they appear to have the same color, and $l_1 - l_2 \perp \text{span}\{\phi_1, \phi_2, \phi_3\}$ is invisible. In this case, $l_1(\cdot)$ and $l_2(\cdot)$ are said to be *metameric*.

**Color Space and Tristimulus Values**

Let $\{p_1(\lambda), p_2(\lambda), p_3(\lambda)\}$ be three spectrum distributions of colors, where $\Phi(p_j)$ are linearly independent of each other. Then it is easy to verify that

$$
\Phi(w_1 p_1 + w_2 p_2 + w_3 p_3) = \Phi(l)
$$

if and only if

$$
\vec{w} = [\Phi(p_1), \Phi(p_2), \Phi(p_3)]^{-1} \Phi(l)
$$

where $\vec{w} = [w_1, w_2, w_3]^T$. The weights $\vec{w}$ is often referred to as the *tristimulus values* of $l(\cdot)$ in *color space* defined by the *primary colors* $\{p_1, p_2, p_3\}$.

**Chromaticity Coordinates**

It should also be noted that to a human eye, $\Phi(l)$ and $k\Phi(l)$ ($k$ is a constant) differ only by brightness. For this reason, given tri-stimulus values $\vec{w}$, we define *chromaticity coordinates*:

$$
\vec{w}_* = \frac{1}{\|\vec{w}\|_{L^1}} \vec{w}
$$

where $\| \cdot \|_{L^1}$ is the $L^1$ norm. Suppose $l_3 = \alpha_1 l_2 + \alpha_2 l_1$, where $\alpha_1, \alpha_2 \geq 0$. Then it can be shown that chromaticity coordinates of $l_3$ is a convex combination of the chromaticity coordinates of $l_1$ and $l_2$. 
Reflection Models

Let us assume that an object surface is an opaque, inhomogeneous medium and that it is uniformly colored. Typically when a light source illuminates a surface, it is partially absorbed and partially scattered. If the light is reflected at the interface, it is the specular component or interface reflection; if the light penetrates the interface and interacts with the colorant particles before being re-emitted through the same interface is called diffuse component or body reflection [55] [36].

In many applications, the body reflection model is adequate. The reflected light \( l_r(\lambda) \) is usually modeled as:

\[
l_r(\lambda) = r(\lambda)l_s(\lambda),
\]

where \( l_s(\lambda) \geq 0 \) is the spectral density of the illuminant (source) and \( r(\lambda) \geq 0 \) is the reflectance of the object surface at the frequency \( \lambda \) [56].

The dichromatic reflection model uses both interface and body reflection models. It states:

\[
l_r(\lambda) = \alpha l_s(\lambda) + r(\lambda)l_s(\lambda), \tag{2.2}
\]

where \( 0 \leq \alpha \leq 1 \) is a constant, and \( \alpha l_s(\lambda) \) term models the interface reflection [55]. In other words, the reflected light \( l_r(\cdot) \) is a mixture of interface and body reflections. Specular highlight is an example of dichromatic reflection, where the reflection on a smooth surface appears to be white because of high \( \alpha \) value [36].

Substituting (2.2) to (2.1) we have

\[
\Phi_j(l_r) = \int_{-\infty}^{\infty} \phi_j(\lambda) (\alpha + r(\lambda)) l_s(\lambda) d\lambda
\]

\[
= \alpha \Phi_j(l_s) + \Phi_j(r l_s), \quad j \in \{1, 2, 3\}.
\]
2.1.3 Opponent Color

Section 2.1.2 described the spectral density of a light is measured by the photoreceptors. Hering proposed that there are two levels of interpreting a color in the HVS: at the receptor level, and in the opponent color space [18]. Since his proposal, there have been numerous experiments confirming that low-level image processing (spatial [6] [50] [57], temporal [47], chromatic adaptation [29] [30]) is being performed in this opponent color space.

The colors red, green, yellow, and blue are called the unique hues; the opponent color theory asserts that red neutralizes green, and yellow neutralizes blue. Hurvich and Jameson’s experiment is credited with giving conclusive evidence that the opponent color system exists in the HVS, and this theory continues to have many proponents [27]. For example, Poirson and Zhang showed evidences that low level spatial processing is performed in the HVS in the opponent color space [50] [74] [75] [76]. Likewise, CIELAB space maintains red-green blue-yellow relationship. See fig. 2.1.

Let $l(\lambda)$ be the spectral distribution of a light at the frequency $\lambda$, as before. At
the basic level, opponent color representation is assumed to be formed by taking a linear combination of the cone response $\Phi(l) \in \mathbb{R}^3$ (though recent work reveals that the transformation is slightly more complicated than that [6]). That is, let $M \in \mathbb{R}^{3 \times 3}$ be a non-singular color conversion matrix from $L, M, S$ cone responses to the opponent color space. Then

$$\vec{v} = M\Phi(l),$$

where $\vec{v} = [v_1, v_2, v_3]^T$, consisting of a achromatic channel $v_1$ and chromatic channels $v_2$ (red-green) and $v_3$ (yellow-blue). Large positive $v_2$ ($v_3$) values have large red (yellow) values; large negative $v_2$ ($v_3$) values have large green (blue) values.

The opponent color system is difficult to derive precisely. Typically, the $M \in \mathbb{R}^{3 \times 3}$ matrix is computed from measurements made in psycho-visual experiments involving color registration or spatial color contrast. Ingling compares different experimental measurements [25]. In our study, we used the color spaces defined by Jameson and by Poirson, but there are no significant differences in the outcome [29] [50]. According to Jameson, coefficients for the matrix $M$ is:

$$M = \begin{bmatrix} 0.8524 & 0.1538 & 0 \\ 1.6643 & -2.2299 & 0.3676 \\ 0.3410 & 0.0615 & -0.7130 \end{bmatrix}. \quad (2.3)$$

### 2.1.4 Chromatic Adaptation

Chromatic adaptation is a study of change in the photoreceptive sensitivity of the human visual system (HVS) under various viewing conditions, such as illumination. Generally, the chromatic adaptation mechanism has the effect of discounting the illuminant, and thus metameric colors under one illuminant often appear metameric under another illuminant. Human vision is said to have a color constancy property
if a color of an object appears invariant to the illuminant. There is a considerable amount of literature reported on how the HVS sensitivity to color changes when the human eye adjusts to the chromaticity of the background light, and some have sought to characterize it [30] [68].

There are two main approaches to studying the chromatic adaptation phenomenon. First, psycho-visual experiments are performed in effort to characterize the HVS’s response to the changing environments. Some of the most influential chromatic adaptation models include [3] [6] [24] [28] [29] [30] [35] [50] [57] [68]. Second, psychophysical explanations are given by [69].

The von Kries coefficient law is a theory that describes the relationship between the illuminant and the HVS sensitivity [68] and it accounts for the approximate color constancy in the HVS [3] [35]. The coefficient law has been popularized by many [43] [68] [71]. It asserts that the sensitivity of each cone type adapts to changes in viewing conditions not by altering the shape of its spectral distribution, but by controlling its amplitude [68]. That is, suppose $l_s(\cdot)$ is the spectral distribution of the surrounding field color that a human eye has adopted to, and let $l_F(\cdot)$ be the spectral distribution of the focal field color that are observing. According to the von Kries coefficient law, the HVS response to the focal field color is

$$
\Psi_{K,j}(l_F, l_S) = \int_{-\infty}^{\infty} d_j(\lambda) l_F(\lambda) d\lambda = d_j(\Phi_j(\lambda)),
$$

(2.4)

where $\phi_j(\cdot), \Phi(\cdot)$ is defined as before, and $d_j$ the proportionality constant. Furthermore, (2.4) is attributed to chromatic adaptation mechanism by assuming that the magnitudes of $d_1, d_2, d_3$ are inversely proportional to $\Phi_1(l_S), \Phi_2(l_S), \Phi_3(l_S)$, respectively [68]. A more generalized form of (2.4) was proposed by [3].

However, Hess, Pretori, and Wallach demonstrate that (2.4) is false under the context of chromatic adaptation [28] [29] [30]. Instead, from a series of psycho-
visual experiments, Jameson and Hurvich proposed a different mechanism to account for chromatic adaptation [29], which they refer to as induced opponent response. The induce opponent response model states that the surrounding activity induces an opposite physiological response through incremental processes [29] [30]. As before, let $l_F(\cdot)$ be the spectral distribution of the focal field color and $l_S(\cdot)$ be the spectral distribution of the surrounding field color. The HVS response to the focal ($\Psi_F$) and the surrounding ($\Psi_S$) are

$$
\Psi_F(l_F, l_S) = M^{-1}(c(M\Phi(l_F))^n - \vec{i}_F)
$$

$$
\Psi_S(l_S, l_F) = M^{-1}(c(M\Phi(l_S))^n - \vec{i}_S),
$$

(2.5)

where $\vec{u}^n$ means $[u_1^n, u_2^n, u_3^n]^T$. Here, $c$ is a constant, $n$ is the transducing constant, and $\vec{i}_F$ and $\vec{i}_S$ are the induced activities in the focal and surrounding, respectively. The matrix $M \in \mathbb{R}^{3\times3}$ represents a color space transformation from $L, M, S$ to opponent color space. The induced responses are proportional to the HVS response to the inducing fields:

$$
\vec{i}_F = kM\Psi_S(l_S, l_F)
$$

$$
\vec{i}_S = kM\Psi_F(l_F, l_S),
$$

(2.6)

where $k$ is some constant which usually depends on the size of the inducing field.

Experimentally, it was found that when the focal and surrounding stimuli are isoluminant then $n = 1$ [28]. When the stimuli are neutral, instead, $n = 1/3$ [24] [30]. Interactions between the several inducing elements can be approximated as a weighted average [28].

Induced opponent response theory is still an active of research today, and although some minor modifications to (2.5) have been suggested, the general principles behind the induced opponent response theory are consistent with recent pub-
Figure 2.2: Typical digital camera image processing pipeline.

applications [50] [57]. For example, Chichilnisky argues that the increment-decrement opponency (IDO) model explains the non-symmetrical response in the color opponency [6].

2.2 Image Acquisition Models

As were the case for human visual systems, there are basic mathematical models for digital photography and digital imaging devices.

2.2.1 Digital Camera

Internal workings of a typical cost-effective single-sensor digital camera is shown in fig. 2.2. At each pixel location, the photoreceptor in the image sensor array measures the intensity of the light. In order to measure the pixel values with full-color representation, the light that enters the photoreceptor is first filtered by an array of color filters, arranged in alternating colors. The arrangement shown in
Figure 2.2 is the most popular pattern called *Bayer* pattern [1].

In this thesis, we make several assumptions about the image sensor. Let \( l(\lambda) \) be the spectral distribution of a light at frequency \( \lambda \). The sensor response \( \Theta(l) \in \mathbb{R}^3 \) to light \( l \) is modeled as

\[
\Theta_j(l) = \langle \theta_j, l \rangle = \int_{-\infty}^{\infty} \theta_j(\lambda)l(\lambda)d\lambda,
\]

where \( \theta_j(\lambda) \in \mathbb{R}_+ \), \( j \in \{1, 2, 3\} \) are the sensor sensitivity functions for red, green, and blue pixel components characterized by the CFA, respectively. Furthermore, the sensor is called *colorimetric* if \( \text{span}\{\phi_1, \phi_2, \phi_3\} = \text{span}\{\theta_1, \theta_2, \theta_3\} \). That is, for all \( l \), there exists a matrix \( M_{\theta,\phi} \in \mathbb{R}^{3 \times 3} \) such that

\[
\Phi(l) = M_{\theta,\phi} \Theta(l).
\]

In this thesis, we assume that the sensor response is colorimetric.

The demosaicing algorithm (sometimes called CFA interpolation) interpolates the missing pixel components of the sensor data to reconstruct a full three-color image representation by exploiting spatial redundancies (see [16] [17] [20] [22] [40] [41] [72]). In the presence of high frequency components, the output from these algorithms may suffer from a type of color artifacts called *zippering*. The artifact produces an alternating pattern of pixels with colors irrelevant to the scene, because the CFA is also arranged in alternating colors. While the research in demosaicing algorithms focus mainly on the reduction of zippering artifacts, increasing pixel count also makes the color artifacts less significant.

The color space defined by the CFA may differ from the RGB color spaces used by the display devices and compression standards. The color space conversion step is needed to convert the *sensor color space* image into an image represented in a standardized color space. This usually requires multiplying each pixel color, \( \Theta(l) \),
As stated in the previous section, chromatic adaptation mechanism generally has the effect of discounting the illuminant, and thus metameric colors under one illuminant often appear metameric under another illuminant. In particular, a piece of white paper is believed to appear white regardless of the illuminant. This phenomenon poses a particularly challenging problem in digital color imaging. Because digital cameras measure the light intensity corresponding to the pixel positions in the scene, the captured image often appears different from the scene the photographer sees. The process of adjusting the image appearance to a different viewing condition is commonly known as the white-balance problem. The white-balance step is typically a matrix multiplication. It appears after color space conversion in the figure 2.2, but some designs choose to compensate for the effects of the illuminant before demosaicing.

The output display device often has a nonlinear response. The nonlinearity in CRT monitor is often modeled as $p^\gamma$, where $p$ is the pixel value, and $\gamma$ is a constant. In the gamma correction step, the pixel values are adjusted $(p^{1/\gamma})$ to cancel out the effects of nonlinearity. The $\gamma$ value for a typical display device falls between

Figure 2.3: With no color space conversion (left), with color space conversion (right).
1.8 and 2.5.

There are some (optional) digital camera features not shown in figure 2.2. For example, due to manufacturing variabilities, the image sensor array may contain isolated defective sensors (commonly referred to as hot or dead pixels). The manufacturing yield of the sensors can be increased by applying a simple denoising algorithm to hide these defective sensors before the demosaicing step. Compression algorithms may also follow gamma correction for data storage or communication.

2.2.2 Illuminant Estimation

As we shall see in later chapters, the solution to the white-balance problem using the traditional methods, the color of the illuminant is required. Because of this, the study of white-balance algorithm in the engineering field is often synonymous with the problem of illuminant estimation. Many computational methods to illuminant estimation, derived from physics and statistical modeling of natural scenes, have been proposed [36] [37] [12] [13] [63] [64] [65] [66]. In particular, dichromatic reflectance model proposed in [55] proved useful in this field. Illuminant estimation is also relevant to research in the HVS because human eye is capable to detect the illuminant in most cases [36]. Maloney hypothesizes that the illuminant estimation is not performed as a result of an unique cue in the scene, but as a combination of multiple cues [44]. In this section, we will briefly review existing techniques.

Let \( \Omega \) be the set of all pixels in an image and \( |\Omega| \) be the size of the set \( \Omega \). Let \( l_s(\cdot) \) be the illuminant of the scene we are trying to estimate. If \( r_i(\lambda) \) is the reflectance of the object represented at pixel location \( i \in \Omega \), then pixel value at \( i \) is \( \Theta(\alpha_i l_s + r_i l_s) \) (see (2.2)).

An illuminant estimation technique called gray-world is widely implemented in
commercial digital cameras. It uses the retinex theory \[3\] \[35\], which states that the average chromaticity value of a natural scene is neutral. Mathematically, the spectral distribution of the average reflectance \(\bar{r} = (1/|\Omega|) \sum_i r_i(\lambda)\) is constant. It implies that given a digitally captured image, its average pixel value serves as a good approximation to a neutral color:

\[
\bar{v}_{GW} = \frac{1}{|\Omega|} \sum_{i \in \Omega} \Theta(\alpha_i l_s) + \Theta(r_i l_s)
\]

where \(\alpha\) and \(c\) are scalar. In other words, the pixel values averaged over \(\Omega\) is proportional to \(\Theta(l_s)\) \[3\]. Note, however, that while \(M_{\theta,\phi}\bar{v}_{GW}\) captures the chromatic content of the illuminant, \(c\) remains unknown and it is impossible to estimate the intensity of the color \(l_s\). Gray-world illuminant estimation \(\bar{v}_{GW}\) is used in almost all of the digital cameras today, perhaps because it is extremely efficient. It has a disadvantage that when the image scene is dominated by one color, the gray-world assumption is violated and \(\bar{v}_{GW}\) is less meaningful.

Another popular and efficient illuminant estimation technique is called the maximum pixel value assumption:

\[
\bar{v}_{\text{max}} = \frac{1}{|\Omega_{\text{max}}|} \sum_{i \in \Omega_{\text{max}}} \Theta(\alpha_i l_s) + \Theta(r_i l_s)
\]

where \(\Omega_{\text{max}}\) refers to \(K\) brightest pixels in the image. The assumption is based on two ideas. First, (2.2) and the dichromatic reflection model states that the specular highlight contains most of the illuminant color (i.e. \(\alpha_i\) is large). Second, MacAdam’s reflectance efficiency theory state that neutral colors have higher reflectance values than the colors with strong chromatic content \[42\].

Other more sophisticated illuminant estimators using physics and statistical modeling are proposed. Some use correlation between the possible illuminants and
the observed pixels [12] [63] [64]. Lee exploits convexity property in chromatic-
ity coordinates (see section 2.1.2) [36] [37]. The chromatic coordinate of pixel
Θ(\(\alpha_i l_S + r_i l_S\)) is a convex combination of the chromaticity coordinate of \(\Theta(l_S)\) and
\(\Theta(r_i l_S)\). If we assume that \(\alpha_i\) varies more rapidly than \(r_i\) within an object, a plot of
\(\Theta(\alpha_i l_S + r_i l_S)\) in the chromaticity coordinate forms a line pointing toward the chro-
maticity coordinate of \(\Theta(l_S)\). A variation of this technique is used by [11] [13] [62].
Note again that while it is possible to estimate the chromaticity coordinate of the
illuminant \(l_S\), its intensity remains unknown.
CHAPTER 3
CMOS IMAGE SENSOR NOISE CHARACTERIZATION AND INTRODUCTION TO IMAGE DENOISING

3.1 Noise Characterization

In order for the denoising method to be effective, it is important to understand the noise characteristics in an image sensor. The CMOS photodiode active pixel sensor (APS) typically uses a photodiode and three transistors, all major sources of noise. While investigating the source of noise is beyond the scope of this thesis, hardware analysis suggest that the readout noise takes three main forms: a fixed-pattern noise, defective pixels, and a mixture of independent additive and multiplicative Gaussian noise [60],

\[ Y(i, j) = X(i, j) + (k_0 + k_1X(i, j))\delta(i, j), \]

where \(X(i, j)\) and \(Y(i, j)\) are the ideal and measured sensor values at pixel location \((i, j)\), respectively, \(\delta(i, j) \sim N(0, 1)\) is noise, and \(k_0, k_1 \in \mathbb{R}\) are parameters.

We independently verified the relationship in (3.1) by calibrating Agilent Technologies camera evaluation board HDCP-2000, equipped with a 300K pixel CMOS sensor [2] using image processing techniques. We have the control over all internal registers, including the exposure time and the programmable gain amplifier (PGA). During the calibration experiment, all registers were fixed and all images are captured in unprocessed raw sensor data format.

Define \(\mathcal{E}\{\cdot\}\) as the expectation operator. Inside a room with controlled lighting, the Macbeth color chart is placed in the view of the camera in a fixed position. Assuming that \(\mathcal{E}\{Y\} = X\) and that the colors inside the squares on the color chart
are uniform, the average and the variance of 400 points (from one color channel) measured from one square are taken to be the true $X$ value and the noise variance for that $X$ value, respectively. We repeat this experiment with varying levels of lighting to measure the noise variances for many different $X$ values. We assumed that the variation among the pixel sensors is small compared to the level of noise.

In figure 3.1, the standard deviation of the sensor values is plotted against the estimated $X$ value. Red pixels, blue pixels, and the two different orientations of green pixels are plotted separately. The behavior of the noise in the red, green, and blue channels seem identical. It is clear from these graphs that the standard deviation of the noise and the pixel values are roughly related by an affine equation, as in (3.1). We also note that in low-light photography, pixel values are often no larger than 50, and the level of noise is thus significant (relative to these small pixel values). Moreover, the histogram of the 400 data points measured from the same square in Macbeth color chart reveals that for each $X$, it is not unreasonable to call the shape of the noise distribution Gaussian. See figure 3.2.

### 3.2 Perceived Noise

In one sense, the dark regions of an image are more difficult to process because the signal-to-noise ratio is smaller when the signal value is small, due to $k_0 > 0$. In another sense, it is more difficult to process the image in the bright regions of the image because the level of the noise is greater, due to $k_1 > 0$. More importantly, the signal-dependent noise model (3.1) means that the dark regions of the image appears most noisy to a human eye. To understand this, consider the definition of
Figure 3.1: Graphs of standard deviation of noise v.s. image sensor value (for a fixed PGA). From top-to-bottom, left-to-right, green sensors, red sensors, blue sensors, green sensors.

Figure 3.2: The histogram of noise when $X = 160$. 
Figure 3.3: \((X + (k_0 + k_1 X))^{1/3} - X^{1/3}\) plotted against \(X\). See text.

CIE-Lab color space [26]:

\[
S_L = 116 \left( \frac{S_Y}{T_Y} \right)^{1/3} - 16
\]
\[
S_a = 500 \left( \frac{S_X}{T_X} \right)^{1/3} - \left( \frac{S_Y}{T_Y} \right)^{1/3}
\]  \( (3.2) \)
\[
S_b = 200 \left( \frac{S_Y}{T_Y} \right)^{1/3} - \left( \frac{S_Z}{T_Z} \right)^{1/3}
\]

where \([S_X, S_Y, S_Z]\) are tri-stimulus values in XYZ color space, and \([T_X, T_Y, T_Z]\) are the reference white color, measured also in XYZ color space. Research has shown that CIE-Lab is a uniform color space [26]. That is, the Euclidean distance between two points in CIE-Lab color space is proportional to the perceived difference between two stimuli.

The mathematical feature of interest in (3.2) is the cube-root function. Suppose you plot \((X + (k_0 + k_1 X))^{1/3} - X^{1/3}\) against \(X^{1/3}\), as in figure 3.3. It is clear that the perceived level of noise in the low-signal region is larger than that of the high-signal region. The graph also emphasizes the need for considering the signal-dependency of the noise in the design of image processing algorithm because the human eye is sensitive to the perturbations when the pixel values are small.
3.3 Introduction to Image Denoising

In this section, we are interested in reviewing the general problem of image denoising and discuss existing image processing techniques to gain an intuition for working with the complexity of image signals.

In real-world digital imaging devices, the images we are interested in are often corrupted by device-specific noise. Basic research in image denoising, therefore, would prove useful to wide ranges of applications such as low-light photography and lossy compression. Image sensor is a very important special case to such imaging devices that suffer from noise. While effective methods to remove fixed-pattern noise and defective pixels have been proposed [52], removing noise of the form (3.1) proves difficult.

Many image denoising papers in the literature prefer working with an independent additive white Gaussian noise model [45] [51] [49] [53] [54] [59] [38]. That is, instead of (3.1), an ideal image \( s \) is corrupted by noise according to the following formula:

\[
x = s + k_0 \delta. \tag{3.3}
\]

Note that (3.3) is a special case of (3.1) by setting \( k_1 = 0 \). Although the mathematical elegance and simplicity makes (3.3) attractive for the complex task of designing a denoising algorithm and describing a natural image, this noise model often requires additional techniques to describe the real-world systems. For example, an approximate relationship between (3.3) and (3.1) can be found using a non-linear compander designed with homomorphic filtering [48] or generalized homomorphic filtering [9].

Consider the noisy image in figure 3.4a. Smoothing the noise by taking the spa-
Figure 3.4: Denoising examples. (a) Noisy image \((k_0, k_1) = (25, 0)\). (b) Lowpass filtering. (c) Median filtering. (d) Bilateral filtering. (e) Wiener filtering. (f) Wavelet thresholding.
tial average of neighboring pixels is neither a reasonable method to estimate pixel values nor an effective method to remove noise (figure 3.4b). The low-pass filter smoothes out the high frequency components from the image signal that defines edges while the noisy appearance of the image is still prevalent. Order statistical filtering is a slight improvement over spatial smoothing. In median filtering, for example, each pixel is replaced with the median of its $N \times N$ neighborhood $\eta$ [67]:

$$\hat{x} = \text{median}\{y_i\}_{i \in \eta}. \quad (3.4)$$

The edge details are better preserved, although the level of noise reduction is poor. See figure 3.4c.

Bilateral filtering, which is related to anisotropic diffusion methods, replaces each pixel with a weighted average of pixels in its $N \times N$ neighborhood. The weights are determined by how similar the pixels are to the pixel of interest $y_0$:

$$\hat{x} = c_1 \sum_{i \in \eta} y_i e^{-\left(\frac{u-y_0}{c_2}\right)^2}, \quad (3.5)$$

where $c_1$ and $c_2$ are constants. Image details such as the lines on the scarf are preserved better than median filtering, although it is still not satisfactory. See figure 3.4d.

Pixel-wise adaptive filtering uses $N \times N$ neighborhoods to estimate the local image mean and standard deviation:

$$\hat{x} = \mu + \frac{\sigma^2}{\sigma^2 + k_0^2}(y - \mu) \quad (3.6)$$

where $\sigma$ and $\mu$ are the standard deviations of the image and the noise, respectively [39]. This method, which is also implemented in Matlab as $\text{wiener2}$ function, successfully smoothes the flat regions, but the edges are jagged. This is not surprising since the stationarity assumption needed to estimate the standard deviation of the image signal fails at these edges. See figure 3.4e.
Finally, Donoho et al. set a new trend in image denoising research with the introduction of wavelet shrinkage techniques. Given the energy compaction property of wavelet transforms in image signals, large wavelet coefficients are assumed due to edges and small wavelet coefficients are assumed due to noise. The proposed algorithm eliminates small coefficients, while retaining the large:

\[
\hat{w}_x = \text{sign}(w_y) \max(|w_y| - \tau, 0),
\]

where \( w_x \) and \( w_y \) are the wavelet coefficients of \( X \) and \( Y \), respectively. It is more effective at smoothing the flat regions and producing continuous edges. See figure 3.4f. In the recent literature, statistical modeling of wavelet coefficients has been the most popular approach to describing a natural image. The probabilistic models for the wavelet coefficients considered include Gaussian mixture [7], Gaussian scale mixture [51], and circular-symmetric Laplacian [53] [54]. The study of interdependencies of wavelet coefficients across scale, especially, has gained strong momentum, and pair-wise processing of a coefficient and its parent is common. Contributions from new wavelet techniques, such as undecimated wavelets [38] [23], steerable pyramid [14], complex wavelets [33] [34], and curvelets [4], helped spawn many new developments. While wavelets share some behavioral characteristics with the neurological response of a human eye, in most cases the statistical modeling of wavelets has been derived heuristically.
CHAPTER 4

TOTAL LEAST SQUARES METHOD OF IMAGE DENOISING

We develop a model relating the noisy image to an ideal image in the total least squares (TLS) sense, taking into account the stochastic nature of the noise and allowing small perturbations in the system. Furthermore, we develop a denoising algorithm that, while effective in removing additive white Gaussian noise in (3.3) (i.e. $k_1 = 0$), removes the signal-dependent noise of the form (3.1) well.

This section is organized as follows. In section 4.1, we present our deterministic image model, and introduce the basics of TLS denoising algorithm. The TLS problem is solved for both signal-independent and signal-dependent noise. Generalizations on the algorithm is made in section 4.2, and we demonstrate how they improve the image quality. In section 4.3 we demonstrate the performance of the proposed method on images corrupted by signal-dependent and signal-independent noise. We compare results with the state-of-the-art denoising algorithms.

4.1 TLS Image Denoising Theory

In this section, we introduce the TLS image denoising theory at its basic level. In the image denoising problem, only the noisy image data is observed. We develop an image model relating the noisy image to a clean image based on the TLS framework (section 4.1.1). We solve this TLS problem for the case that an image is corrupted by signal-dependent noise (section 4.1.2). See appendix A for a review of general total least squares.
4.1.1 TLS Image Model

Suppose we are given an ideal clean image, $s$, and a noise corrupted version, $x$. Let $s_0 \in \mathbb{R}^m$ be an image patch from $s$ (i.e. $\sqrt{m} \times \sqrt{m}$ vector cropped from an image) and $\{x_i \in \mathbb{R}^m\}_{i \in \{1, \ldots, n\}}$, $m \geq n + 1$ be a collection of image patches taken from $x$. To find the relationship between $s_0$ and the noisy image, $x$, we would like to represent $s_0$ as a linear combination of $\{x_i\}$:

$$s_0 = X\alpha,$$  \hspace{1cm} (4.1)

where $X = [x_1, \ldots, x_n]$. While (4.1) is a simplistic model to describe the real image signals, it has practical uses. To see this, consider an edge in an image. Image patches over this image feature are invariant to the spatial translations parallel to the edge orientation. Therefore, $s_0$ can be represented efficiently as a linear combination of a collection of noisy image patches $\{x_i\}$ taken by translating image patches in the vicinity of $s_0$. This approach is cited in [46] and is illustrated in figure 4.2. The strategies for selecting $x_i$ are detailed in section 4.2.2.

In general, however, there is no such $\alpha$ that makes (4.1) true because $s_0 \notin \text{span}\{x_i\}$. Suppose we allow a small perturbation $e_0$ in the system so that

$$s_0 + e_0 = X\alpha.$$  \hspace{1cm} (4.2)

The vector $\alpha$ that satisfies (4.2) with the smallest perturbation $e_0$ in the $L^2$ sense is commonly known as the least square (LS) solution. However, the inherent flaw in the above system is that the perturbation is confined to $s_0$, even though there is noise in $X$.

Instead, we propose to allow small perturbations in both $s_0$ and $X$:

$$s_0 + e_0 = (X + E)\alpha.$$  \hspace{1cm} (4.3)
The vector $\alpha$ satisfying (4.3) while minimizing $\| [E, e_0] \|_F^2$ is known as the total least square solution, denoted $\alpha_{\text{TLS}}$. Here, $\| \cdot \|_F$ is the Frobenius norm. In general, the perturbation in $X$ makes the perturbation in $s_0$ smaller.

The solution to (4.3) is well documented [15] [8]. First, examine $[X, s_0]$ using singular value decomposition (SVD)

$$[X, s_0] = U \Sigma V^T,$$  \hspace{1cm} (4.4)

where $\Sigma = \text{diag}(\sigma_1, \ldots, \sigma_{n+1})$, $\sigma_i^2 > \sigma_{i+1}^2$. Then

$$\alpha_{\text{TLS}} = - \begin{bmatrix} v_{1,n+1} \\ \vdots \\ v_{n,n+1} \end{bmatrix}^{-1} u_{n+1,n+1}$$  \hspace{1cm} (4.5)

$$[E, e_0] = -\sigma_{n+1} \begin{bmatrix} u_{1,n+1} \\ \vdots \\ u_{n+1,n+1} \end{bmatrix} \begin{bmatrix} v_{1,n+1} & \cdots & v_{n+1,n+1} \end{bmatrix},$$ \hspace{1cm} (4.6)

where $[u_{1,n+1}, \ldots, u_{n+1,n+1}]^T$ and $[v_{1,n+1}, \ldots, v_{n+1,n+1}]^T$ are the left and right singular vectors corresponding to $\sigma_{n+1}$, respectively.

\subsection*{4.1.2 TLS Solution using Noise Model}

The solution (4.5) requires the knowledge of the clean image patch $s_0$, but this is not available in a denoising problem. In this section, we develop a method to compute $\alpha_{\text{TLS}}$ using the noise model (3.1), and $s_0$ is not provided.

Define $s_i$ as an image patch from ideal image $s$ corresponding to $x_i$, and assume $s_0 \in \{s_i\}$. Then

$$x_i = s_i + k_0 \delta_i + k_1 \text{diag}(s_i) \delta_i,$$ \hspace{1cm} (4.7)
where $\delta_i \in \mathbb{R}^m$ is a noise vector and $\text{diag}(s_i)$ is a diagonal matrix whose diagonal entries are the entries of $s_i$. Recall that when $k_1 = 0$, (4.7) represents an image corrupted by a signal-independent noise described in (3.3).

We solve for $\alpha_{\text{TLS}}$ without $s_0$ given and taking into account the stochastic nature of $\delta_i$. Consider the following:

$$P = [X, s_0]^T [X, s_0]$$

$$= (U \Sigma V^T)^T (U \Sigma V^T) = V \Sigma^2 V^T,$$

(4.8)

where $[X, s_0] = U \Sigma V^T$ is the SVD in (4.4). Our strategy is to estimate $P$ and obtain the right singular vector $V$ through its eigen decomposition, as in (4.8). Once $V$ is found, $\alpha_{\text{TLS}}$ is computed using (4.5).

Define $\mathcal{E}\{\cdot\}$ as the expectation operator and assume

$$\mathcal{E}\{\delta_i\} = 0$$

$$\mathcal{E}\{\delta_i \delta_j^T\} = \begin{cases} I & i = j \\ 0 & i \neq j, \end{cases}$$

(4.9)

where $I$ is the identity matrix. We estimate $P$ by making the assumption that

$$P \approx \mathcal{E}\{P\},$$

(4.10)

when $m \gg n+1$. The relationship (4.10) is generally true because noise is spatially uncorrelated and a large number of averaging occurs in $X^T X$ due to the height of the matrix $X \in \mathbb{R}^{m \times n}$. A numerical analysis and verification of this assumption is detailed in section 4.3.
For now, let $P = \mathcal{E}\{P\}$. Then

$$P = \mathcal{E}\{[X, s_0]^T[X, s_0]\}$$

$$= \begin{bmatrix} \mathcal{E}\{X^TX\} & \mathcal{E}\{X^T s_0\} \\ \mathcal{E}\{s_0^T X\} & \mathcal{E}\{s_0^T s_0\} \end{bmatrix} = \begin{bmatrix} P_{XX} & S^T s_0 \\ s_0^T S & s_0^T s_0 \end{bmatrix}$$

where $P_{XX} = \mathcal{E}\{X^T X\}$. Simplifying $P_{XX}$,

$$P_{XX} = \mathcal{E}\{X^T X\}$$

$$= S^T S + \mathcal{E}\{(k_0 \Delta^T + k_1 \Delta_S^T)(k_0 \Delta + k_1 \Delta_S)\}$$

$$= S^T S + k_0^2 \mathcal{E}\{\Delta^T \Delta\} + k_0 k_1 \mathcal{E}\{\Delta^T \Delta_S\} + k_0 k_1 \mathcal{E}\{\Delta_S^T \Delta\} + k_1^2 \mathcal{E}\{\Delta_S^T \Delta_S\}$$

$$= S^T S + mk_0^2 I + k_1^2 \sum_{i=1}^{m} \text{diag}(s_{i,1}^2, \ldots, s_{i,n}^2) + 2k_0 k_1 \sum_{i=1}^{m} \text{diag}(s_{i,1}, \ldots, s_{i,n}),$$

where $\Delta = [\delta_1, \ldots, \delta_n]$ and $\Delta_S = [\text{diag}(s_1)\delta_1, \ldots, \text{diag}(s_n)\delta_n]$. When $m \gg n + 1$, we can also approximate $\sum_i s_{i,j}$ as $\sum_i x_{i,j}$, which is computable. Therefore,

$$P_{XX} = S^T S + mk_0^2 I + k_1^2 \sum_{i=1}^{m} \text{diag}(s_{i,1}^2, \ldots, s_{i,n}^2) + 2k_0 k_1 \sum_{i=1}^{m} \text{diag}(s_{i,1}, \ldots, s_{i,n}).$$

Using the fact that the $j$th diagonal entry of $S^T S$ is $\sum_{i=1}^{m} s_{i,j}^2$, $S^T S$ can be estimated using the following procedure:

1. Compute $P_{XX} = X^T X$.

2. Compute $P_{XX} - k_0^2 m I - 2k_0 k_1 \sum_i \text{diag}(x_{i,1}, \ldots, x_{i,n})$.

3. Multiply the diagonal entries of matrix in step 2 by $(1 + k_1^2)^{-1}$.

$S^T s_0$ and $s_0^T s_0$ can be estimated by taking the appropriate rows and columns from the above $S^T S$ estimate. Therefore, the matrix $P$ is fully computable. The new $\alpha_{\text{TLS}}$ is computed from (4.5), where $[v_{1,n+1}, \ldots, v_{n+1,n+1}]^T$ is the eigen vector corresponding to the smallest eigen value of $P$. Our best estimate for $s_0$ is

$$\hat{s}_0 = X \alpha_{\text{TLS}}.$$  (4.11)
4.2 Enhancements to TLS Image Model

In this section, we offer a number of different generalizations to the TLS image models developed in section 4.1. In some cases, variables are redefined to match the improved behaviors of these generalized algorithms.

4.2.1 Affine Approximation

A variation to the TLS problem (4.3) using an affine approximation model was solved by de Groen [8]. He showed that \( \| [E, e] \|_F^2 = \sigma_{n+1}^2 \) is reduced greatly when the column-means of \([X, s_0]\) are subtracted from their respective columns first, suggesting a better model fit. More specifically, instead of (4.3), we solve for \( \alpha \) in the following system that minimizes \( \| E, e_0 \|_F^2 \):

\[
\tilde{s}_0 + e_0 = (\tilde{X} + E)\alpha \quad (4.12)
\]

where \( \tilde{s}_0 = s_0 - \bar{s}_0 \), \( \tilde{x}_i = x_i - \bar{x}_i \) (ith column of \( \tilde{X} \)), and \( \bar{s}_0, \bar{x}_i \in \mathbb{R} \) are the average values of elements in \( s_0 \) and \( x_i \), respectively. Likewise, let \( \tilde{s}_i = s_i - \bar{s}_i \), \( \tilde{S} = [\tilde{s}_1, \ldots, \tilde{s}_n] \). Note that for \( m \gg 1 \), \( \bar{s}_i \approx \bar{x}_i \), and so

\[
\bar{x}_i = \tilde{s}_i + k_0 \delta_i + k_1 \text{diag}(s_i) \delta_i.
\]

The solution to (4.12) is still given by (4.5) and (4.6), except using the SVD of \([\tilde{X}, \tilde{s}_0] = U \Sigma V^T \). To solve for the right singular vectors \( V \), let \( P = [\tilde{X}, \tilde{s}_0]^T [\tilde{X}, \tilde{s}_0] = V \Sigma^2 V^T \), and assume \( P = \mathcal{E}\{P\} \) for \( m \gg n + 1 \) as before. Then

\[
P = \mathcal{E}\{[\tilde{X}, \tilde{s}_0]^T [\tilde{X}, \tilde{s}_0]\} = \begin{bmatrix}
P_{XX} & \tilde{S}^T \tilde{s}_0 \\
\tilde{s}_0^T \tilde{S} & \tilde{s}_0^T \tilde{s}_0
\end{bmatrix},
\]
where

\[ P_{XX} = \mathcal{E}\{\tilde{X}^T \tilde{X}\} \]

\[ = \tilde{S}^T \tilde{S} + \sum_{i=1}^{m} \text{diag}(k_0 + k_1 s_{i,1}, \ldots, k_0 + k_1 s_{i,n})^2 \]

\[ = \tilde{S}^T \tilde{S} + \sum_{i=1}^{m} \text{diag}(k_0 + k_1 (\tilde{s}_1 + \tilde{s}_{i,1}), \ldots, k_0 + k_1 (\tilde{s}_n + \tilde{s}_{i,n}))^2 \]

\[ = \tilde{S}^T \tilde{S} + \sum_{i=1}^{m} \text{diag}(k_0 + k_1 \tilde{s})^2 + \sum_{i=1}^{m} k_1^2 \text{diag}(\tilde{s}_{i,1}, \ldots, \tilde{s}_{i,n})^2 \]

\[ + 2 \sum_{i=1}^{m} k_1 \text{diag}(k_0 + k_1 \tilde{s}) \text{diag}(\tilde{s}_{i,1}, \ldots, \tilde{s}_{i,n}). \]

Again, using the substitution \( \sum_i s_{i,j} = \sum_i x_{i,j} \), \( \tilde{S}^T \tilde{S} \) is estimated from subtracting \( \sum_{i=1}^{m} \text{diag}(k_0 + k_1 \tilde{s})^2 + 2 \sum_{i=1}^{m} k_1 \text{diag}(k_0 + k_1 \tilde{s}) \text{diag}(\tilde{s}_{i,1}, \ldots, \tilde{s}_{i,n}) \) from \( P_{XX} \) and multiplying the diagonal entries by \( (1 + k_1^2)^{-1} \). The matrix \( P \) and our solution to (4.12) are found as explained in section 4.1.2. Our best estimate for \( \tilde{s}_0 \) is \( \tilde{X}\alpha_{\text{TLS}} \), and thus our estimate for \( s_0 \) is

\[ \hat{s}_0 = \tilde{X}\alpha_{\text{TLS}} + \bar{x}_0, \]

where \( \bar{x}_0 \) is the average of the noisy image patch corresponding to \( s_0 \).

### 4.2.2 Weighted TLS and Image Patches

In this section, an alternative TLS cost function is introduced. Adaptive methods to selecting relevant noisy image patches and to determining the appropriate shape of the patches are developed by solving the new TLS optimization problem.

Recall \( E \) and \( e_0 \), the matrix perturbations in \( X \) and \( s_0 \). Notice that each row of \( [E, e_0] \) corresponds to the perturbation in a particular pixel position within image patches, and each column of \( [E, e_0] \) refers to perturbation in a corresponding image patch. Let \( A = \text{diag}(a_1, \ldots, a_m) \), \( B = \text{diag}(b_1, \ldots, b_{n+1}) \), \( B \) non-singular.
The TLS image model can be modified so that $\alpha$ is chosen to satisfy (4.12) while minimizing

$$\|A[E, e_0]B\|_F^2$$

instead of $\|E_0\|_F^2$. Below, $a_i$ and $b_i$ will be referred to as weights.

Above, $\{b_1, \ldots, b_{n+1}\}$ scale the columns of $[E, e_0]$, individually controlling the degree of perturbation allowed in each image patch. Less perturbation is allowed in the image patches corresponding to larger weights, and thus the estimated ideal image patch, $\hat{s}_0 = X\alpha$, is closer to these image patches. Alternatively, the matrix $B$ can limit the contribution from irrelevant image patches by assigning smaller weights to them.

In section 4.1.1, we motivated the image patch model by suggesting that $s_0$ can be represented efficiently as a linear combination of noisy image patches $\{x_i\}$ collected by spatially translating patches in the direction parallel to the edge orientation, in the spatial vicinity of $s_0$. While determining the precise edge orientation is difficult, especially when the image signal is noisy, the weighting scheme in (4.13) offers a systematic way to adaptively discriminate between various spatial translations of the image patches.

More specifically, suppose $\{x_i\}$ is a set of all noisy image patches in the spatial vicinity of $s_0$. Owing to the techniques developed for bilateral filtering [61], each spatial translation is evaluated and larger weights are assigned to image patches whose spatial translations are more relevant. Let $\text{dist}_B(x_i, x_0)$ be a range distance function between image patches $x_i$ and $x_0$ (returns a smaller number if $x_i$ and $x_0$
are similar). Then define

\[ b_i = \begin{cases} \exp\left(-\text{dist}_B(x_i, x_0)^2/k_B\right), & \forall i \leq n \\ \gamma, & \forall j > n \end{cases} \tag{4.14} \]

where \( \gamma, k_B \) are constants, and \( x_0 \) is a noisy image patch whose spatial location corresponds exactly to that of \( s_0 \). We chose to work with exponentials because of the fast roll-off as \( \text{dist}(\cdot, \cdot) \) becomes large. Notice that the proposed weighting scheme favors spatial translation of image patches in the direction parallel to the edge orientation because image patches are similar in that direction. Thus \( b_i \) is a soft selection of image patches.

In the results presented in this thesis, we use:

\[ \text{dist}_B(\phi, \psi) = \|H(\phi - \psi)\|_2, \]

where \( H = \text{diag}(h_1, \ldots, h_m) \) and \([h_1, \ldots, h_m]\) is a Gaussian envelope centered at the center of the \( \sqrt{m} \times \sqrt{m} \) image patch. \( H \) is needed because \( m \gg n \) is large. There are many other choices for the design of distance metrics, including the use of perceptual distortion metrics, and we leave this question as an open research problem.

Recall \( A = \text{diag}(a_1, \ldots, a_m) \). In (4.13), \( \{a_1, \ldots, a_{n+1}\} \) scale the rows of \([E, e_0]\), individually controlling the degree of perturbation allowed for each pixel position within an image patch. Analogous to above, less perturbation is allowed in the pixel locations corresponding to larger weights, and the matrix \( A \) can limit the contribution from certain pixels by assigning smaller weights to them.

In section 4.1.1, an image patch is defined as a \( \sqrt{m} \times \sqrt{m} \) vector cropped from an image. However, the square shape of the patch is much too restrictive for the general image patch modeling problem, where an abstract image patch shape may
be preferred. The weighting scheme in (4.13) offers a systematic way to adaptively determine the shape of the image patches.

More specifically, suppose $x_i$ is a $\sqrt{m} \times \sqrt{m}$ vector cropped from the noisy image. Owing to the techniques developed for bilateral filtering again \cite{61}, each pixel location in an image patch is compared to the pixel at the center of image patch, and assigned a weight $a_j$ according to its relevance to the center pixel. Let $\text{dist}_A(y_i, y_0)$ be the Euclidean distance function between the vectors $y_i$ and $y_0$ (returns a smaller number if $y_i$ and $y_0$ are similar). Then define

$$a_j = \exp\left(-\text{dist}_A(y_j, y_0)^2/k_A\right)$$

where $k_A$ is a constant, $y_j$ is the $j$-th row of $X$, and $y_0$ is the row in $X$ corresponding to the center pixel of $\sqrt{m} \times \sqrt{m}$ image patch. Once again, exponentials were used because of the fast roll-off. The proposed weighting scheme favors pixels that belong to the same object being represented by the center pixel of the image patch. Thus $a_j$ is a soft shape of image patches, $\{x_i\}$. As before, there are many other choices for the design of adaptive image patch shape, and we leave this question as an open research problem.

Mathematically, the $\alpha$ value that minimizes $\|A[E, e_0]B\|$ while satisfying $\tilde{s}_0 + e_0 = (\tilde{X} + E)\alpha$ is

$$\alpha_{\text{TLS}} = -\frac{1}{v_{n+1,n+1}b_{n+1}} \text{diag}(b_1, \ldots, b_n) \begin{bmatrix} v_{1,n+1} \\ \vdots \\ v_{n,n+1} \end{bmatrix}, \quad (4.15)$$

where $A[\tilde{X}, \tilde{s}_0]B = U\Sigma V^T$ is a singular value decomposition ($\sigma_i^2 > \sigma_{i+1}^2$ as before). See appendix A for details. From (4.15) we see that $B$ also has the effect of reducing the magnitudes of $\alpha$ coefficients corresponding to the image patches in
\{x_i\} that poorly describes the image features in the region of interest. To solve for the right singular vector matrix \( V \), let 
\[
P = (A[\tilde{X}, \tilde{s}_0]B)^T (A[\tilde{X}, \tilde{s}_0]B) = V\Sigma^2V^T,
\]
and assume \( P = \mathcal{E}\{P\} \) for \( m \gg n + 1 \) as before. Then
\[
P = \mathcal{E}\{(A[\tilde{X}, \tilde{s}_0]B)^T (A[\tilde{X}, \tilde{s}_0]B)\}
\]
\[
= B \left[ \begin{array}{cc}
P_{XX} & \tilde{S}^T A^2 \tilde{s}_0 \\
\tilde{s}_0^T A^2 \tilde{S} & \tilde{s}_0^T A^2 \tilde{s}_0
\end{array} \right] B,
\]
where
\[
P_{XX} = \mathcal{E}\{\tilde{X}^T A^2 \tilde{X}\}
\]
\[
= \tilde{S}^T A^2 \tilde{S} + \sum_{i=1}^{m} a_i^2 \text{diag}(k_0 + k_1 s_{i,1}, \ldots, k_0 + k_1 s_{i,n})^2
\]
\[
= \tilde{S}^T A^2 \tilde{S} + \sum_{i=1}^{m} a_i^2 \text{diag}(k_0 + k_1 (\tilde{s}_1 + \tilde{s}_{i,1}), \ldots, k_0 + k_1 (\tilde{s}_n + \tilde{s}_{i,n}))^2
\]
\[
= \tilde{S}^T A^2 \tilde{S} + \sum_{i=1}^{m} a_i^2 \text{diag}(k_0 + k_1 \tilde{s})^2 + \sum_{i=1}^{m} a_i^2 k_1^2 \text{diag}(\tilde{s}_{i,1}, \ldots, \tilde{s}_{i,n})^2
\]
\[
+ 2 \sum_{i=1}^{m} a_i^2 k_1 \text{diag}(k_0 + k_1 \tilde{s}) \text{diag}(\tilde{s}_{i,1}, \ldots, \tilde{s}_{i,n}).
\]
Using the substitution \( \sum_i a_i^2 s_{i,j} = \sum_i a_i^2 x_{i,j} \) for \( m \gg n + 1 \), \( \tilde{S}^T A^2 \tilde{S} \) is estimated from subtracting \( \sum_{i=1}^{m} a_i^2 \text{diag}(k_0 + k_1 \tilde{s})^2 + 2 a_i^2 k_1 \text{diag}(k_0 + k_1 \tilde{s}) \text{diag}(\tilde{s}_{i,1}, \ldots, \tilde{s}_{i,n}) \) from \( P_{XX} \) and multiplying the diagonal entries by \( (1 + k_1^2)^{-1} \). \( \tilde{S}^T A^2 \tilde{s}_0 \) and \( \tilde{s}_0^T A^2 \tilde{s}_0 \) can be estimated by taking the appropriate rows and columns from the \( S^T A^2 S \) estimate. Therefore, the matrix \( P \), whose eigenvectors are used to calculate \( \alpha \) in (4.15), is fully computable.

\( A \) and \( B \) amount to a very significant contribution to the overall image quality of the output image. See an example in figure 4.1. The image denoised with weights \( A \) and \( B \) has sharper details and less noticeable edge artifacts, and the behavior of the algorithm is often more stable. More sophisticated weighting schemes (e.g. non-diagonal matrices) may be considered in our future research.
4.2.3 Multi-Resolution Image Patches

In section 4.1.1, the estimated image patch $\hat{s}_0$ is described as a weighted average of a collection of noisy image patches, $\{x_i\}$. If each $x_i$ contains an image feature that also appears in $s_0$, then this feature can be preserved in $\hat{s}_0$ because it survives the averaging. Therefore in order that our image model (4.12) be effective, the objective is to find and choose the set $\{x_i\}$ such that image features in $s_0$ are well captured. In sections 4.1.1 and 4.2.2, spatial locality was exploited by taking the $\sqrt{m} \times \sqrt{m}$ vectors cropped from the noisy image $x$ in the spatial vicinity of $s_0$. We will refer to a set of image patches collected this way as $\{x_i^{(1)}\}$ and it is illustrated in figure 4.2.

In this section, motivated by multi-resolution analysis and the self-similarity properties in a natural image, we propose to take the $\sqrt{m} \times \sqrt{m}$ vectors in the spatial vicinity of $s_0$ from a noisy image decimated by two in horizontal and vertical directions. This is illustrated in figure 4.3. A set of image patches collected from the decimated image, $\{x_i^{(2)}\}$, are also effective for modeling large image features,
corners, and gradual (blurry) edges. To see this, suppose $s_0$ contains a corner. Then the noisy image patch taken from the decimated noisy image at the same spatial location also contains a corner with the same angle and orientation. While it is desirable to apply an anti-aliasing filter before decimation, we did not consider it in this thesis because modeling the signal-dependency of noise after filtering is difficult.

In this thesis, we use $\sqrt{m} \times \sqrt{m}$ vectors generated using both methods. Through experimental results, we see that the inclusion of $\{x_i^{(2)}\}$ leads to sharpening of sharp edges and smoothing of smooth surfaces, contributing to a significant image quality gain. See figure 4.4 for comparison.
Figure 4.4: Example demonstrating the sharpening of edges due to the inclusion of decimated image patches. Noisy input image cropped from “Lena” (left), output image using \( \{x_i^{(1)}\} \) only (middle), output image using \( \{x_i^{(1)}\} \) and \( \{x_i^{(2)}\} \) (right).

### 4.2.4 Redundant Estimation

Let us generalize the TLS image model (4.12) further. Let \( \tilde{S}_0 = [\tilde{s}_1, \ldots, \tilde{s}_p] \), where \( \{s_i\} \) is a collection of image patches from \( s \). Then our new TLS system is modified as follows:

\[
\tilde{S}_0 + E_0 = (\tilde{X} + E)\alpha, \tag{4.16}
\]

where the perturbation \( E_0 \) is now \( m \times p \), and \( \alpha \in \mathbb{R}^{n \times p} \).

Let \( A[\tilde{X}, \tilde{S}_0]B = U \Sigma V^T \) be SVD, where \( \Sigma = \text{diag}(\sigma_1, \ldots, \sigma_{n+p}) \), \( \sigma_1^2 > \sigma_{i+1}^2 \).

Partition \( U \) and \( V \) as follows:

\[
U = \begin{bmatrix} U_1 & U_2 \end{bmatrix}, \quad V = \begin{bmatrix} V_{1,1} & V_{1,2} \\ V_{2,1} & V_{2,2} \end{bmatrix}
\]

\[
\begin{array}{cc}
U & n \\
p & p
\end{array}
\]

\[
\Sigma_2 = \text{diag}(\sigma_{n+1}, \ldots, \sigma_{n+p}).
\]

Then the matrix \( \alpha \) that minimizes \( \|A[E, E_0]B\|_F^2 \) while satisfying (4.16) is [15] [8]:

\[
\alpha_{\text{TLS}} = -\text{diag}(b_1, \ldots, b_n)V_{1,2}V_{2,2}^{-1}\text{diag}(b_{n+1}, \ldots, b_{n+p})^{-1} \tag{4.17}
\]

\[
A[E, E_0]B = -U_2\Sigma_2[V_{1,2}^T, V_{2,2}^T]. \tag{4.18}
\]
See appendix A for a sketch of a proof.

To solve for the right singular vector matrix \( V \), let

\[
P = (A[\bar{X}, \tilde{S}_0]B)^T(A[\bar{X}, \tilde{S}_0]B) = V \Sigma^2 V^T, \quad (4.19)
\]

and assume \( P = \mathcal{E}\{P\} \) for \( m \gg n + 1 \) as before (see section 4.3). Then

\[
P = \mathcal{E}\{(A[\bar{X}, \tilde{S}_0]B)^T(A[\bar{X}, \tilde{S}_0]B)\} = B \begin{bmatrix} P_{XX} & \tilde{S}_0^T A^2 \tilde{S}_0 \\ \tilde{S}_0^T A^2 \tilde{S} & \tilde{S}_0^T A^2 \tilde{S}_0 \end{bmatrix} B
\]

where \( P_{XX} = \mathcal{E}\{\bar{X}^T A \bar{X}\} \). Using the procedure outlined in section 4.2.2 yields an estimate of \( \tilde{S}_0^T A^2 \tilde{S} \) from \( P_{XX} \). Furthermore, \( \tilde{S}_0^T A^2 \tilde{S} \) and \( \tilde{S}_0^T A^2 \tilde{S}_0 \) are estimated by taking the appropriate rows and columns from \( \tilde{S}_0^T A^2 \tilde{S} \) estimate. Therefore, the matrix \( P \), whose eigenvectors are used to calculate \( \alpha \) in (4.17), is fully computable.

Working with (4.16) has several advantages over (4.12). First, by choosing to minimize the perturbation in multiple image patches \( \{s_i\} \) simultaneously, the algorithm becomes more robust against noise. To see this, note that \( A[E, E_0]B \) in (4.18) is rank \( p \), which offers more freedom over the perturbation than (4.6) allows. This is also in a sharp contrast to the analogous LS system,

\[
\tilde{S}_0 + E_0 = \bar{X} \alpha
\]

because the LS solution to the above system that minimizes \( \|E_0\|_F^2 \) will be no different than if each columns of \( E_0 \) were minimized independently.

Second, our best estimate for the denoised image patches is

\[
\tilde{S}_0 = \bar{X} \alpha_{\text{TLS}} + \begin{bmatrix} 1 \\ \vdots \\ [\bar{x}_1, \ldots, \bar{x}_p] \\ 1 \end{bmatrix}.
\]
Assuming that \( \{s_1, \ldots, s_p\} \) were picked from the same region of the image \( s \), there will be overlapping regions in the denoised image patches. We benefit from this by combining some or all of estimated pixel values that are available at each position. With this technique, the edge artifacts are reduced and smooth surfaces become significantly smoother, while the sharpness of the edges are preserved.

### 4.2.5 Pre-Processing

The effectiveness of the TLS denoising algorithm depends on our ability to estimate \( P \) matrix accurately. Given the noise characteristics (4.7), there will be one or two pixels occasionally that stand out because the value of \( \delta \) at that pixel position is far greater than the standard deviation of the noise. This is problematic because many entries in \( X \) appear more than once, degrading our estimate for \( P \) greatly.

To work around this problem, we propose to prune the outliers. In this paper, the following pre-processing procedure was used. For each pixel location in \( x \),

1. Crop a \( 5 \times 5 \) vector from \( x \). We will call it \( y \).

2. Sort all of the pixels in \( y \), and find the \( N \)th largest and \( N \)th smallest pixel values in \( y \).

3. If the center pixel in \( y \) is larger (smaller) than the \( N \)th largest (smallest) pixel value in \( y \), replace the center pixel value with the \( N \)th largest (smallest) pixel value in \( y \).

### 4.3 Implementation and Results

We performed simulation experiments on 8-bit gray-scale test images. The parameters were \( m = 23 \times 23 = 529 \) and \( n_1 = 5 \times 5 = 25 \), \( n_2 = 5 \times 5 = 25 \), where \( n_1 \) and
Figure 4.5: Noise \((k_0, k_1) = (25, 0.2)\) and average error magnitude as a function of signal value. Error magnitude in the proposed algorithm is signal-independent.

\(n_2\) refer to the number of vectors in \(\{x^{(1)}\}\) and \(\{x^{(2)}\}\), respectively. The columns of \(\tilde{S}_0 = [\tilde{s}_1, \ldots, \tilde{s}_p]\), used in the redundant estimation of \(s\), come from image patches in \(s\) corresponding to \(\{x^{(1)}_i\}\). In all cases, we assumed that the parameters \(k_0\) and \(k_1\) were available a priori. In the presence of signal-independent noise, algorithms to estimate the noise variance has been proposed [31]. In real-world CMOS sensors, \(k_0\) and \(k_1\) depend closely on the programmable amplifiers in the A/D converter. We may assume that the gain for the amplifier is provided, and that relationship between the gain and \(k_0, k_1\) is understood from the calibration experiments.

Given the parameters above, we first verified our claim (4.10) \(P = \mathcal{E}\{P\}\). Tables 4.1 and 4.2 show the average values for \(\|P - \mathcal{E}\{P\}\|_2 / \|P\|_2\) given well-known test images, where \(\| \cdot \|_2\) is a matrix \(L^2\) norm. These tables give a rough estimate of the magnitude of the error in the above assumption, relative to the matrix norm of \(P\). When \(P\) is defined as (4.8), the differences between \(P\) and \(\mathcal{E}\{P\}\) are almost...
Figure 4.6: Example with signal-independent noise, $k_0 = 25$. First column: noisy input image, and output from proposed algorithm, and output from method in [51]. Second column: output from method in [49], output from method in [54], and output from method in [45].
Figure 4.7: Example with signal-independent noise, $k_0 = 50$. First column: noisy input image, output from proposed algorithm, and output from method in [51]. Second column: output from method in [49], output from method in [54], and output from method in [45].
Figure 4.8: Example with signal-dependent noise, $k_0 = 15, k_1 = 0.4$. First column: noisy input image, output from proposed algorithm, and output from method in [51] (with [9]). Second column: output from method in [49] (with [9]), output from method in [54] (with [9]), and output from method in [45] (with [9]).
Table 4.1: Error in the assumption (4.10), measured as $\|P - \mathcal{E}\{P\}\|_2/\|P\|_2$. Matrix $P$ is defined as (4.8).

<table>
<thead>
<tr>
<th>$(k_0, k_1)$</th>
<th>Lena</th>
<th>Barbara</th>
<th>Boats</th>
<th>House</th>
<th>Peppers</th>
<th>F Print</th>
</tr>
</thead>
<tbody>
<tr>
<td>(25,0.0)</td>
<td>0.000021</td>
<td>0.000020</td>
<td>0.000022</td>
<td>0.000024</td>
<td>0.000014</td>
<td>0.000015</td>
</tr>
<tr>
<td>(25,0.1)</td>
<td>0.000030</td>
<td>0.000030</td>
<td>0.000032</td>
<td>0.000036</td>
<td>0.000021</td>
<td>0.000024</td>
</tr>
<tr>
<td>(25,0.2)</td>
<td>0.000039</td>
<td>0.000041</td>
<td>0.000042</td>
<td>0.000049</td>
<td>0.000029</td>
<td>0.000033</td>
</tr>
</tbody>
</table>

Table 4.2: Error in the assumption (4.10), measured as $\|P - \mathcal{E}\{P\}\|_2/\|P\|_2$. Matrix $P$ is defined as (4.19).

<table>
<thead>
<tr>
<th>$(k_0, k_1)$</th>
<th>Lena</th>
<th>Barbara</th>
<th>Boats</th>
<th>House</th>
<th>Peppers</th>
<th>F Print</th>
</tr>
</thead>
<tbody>
<tr>
<td>(25,0.0)</td>
<td>0.0873</td>
<td>0.1218</td>
<td>0.1080</td>
<td>0.0931</td>
<td>0.0625</td>
<td>0.0983</td>
</tr>
<tr>
<td>(25,0.1)</td>
<td>0.1301</td>
<td>0.1675</td>
<td>0.1506</td>
<td>0.1458</td>
<td>0.0936</td>
<td>0.1421</td>
</tr>
<tr>
<td>(25,0.2)</td>
<td>0.1733</td>
<td>0.2085</td>
<td>0.1917</td>
<td>0.1966</td>
<td>0.1287</td>
<td>0.1829</td>
</tr>
</tbody>
</table>

insignificant (see table 4.1). When $P$ is defined with adaptive weights, as in (4.19), then the difference between $P$ and $\mathcal{E}\{P\}$ is still small enough to support the claim in (4.10) (see table 4.2), but the magnitude of error considerably larger than the values in table 4.1. This is not surprising, because the assumption (4.10) is based on the fact that a large number of averaging occurs in $X^TX$ when $m \gg n$. However, introducing the weighting matrix $A$ skews averaging, limiting the effectiveness of assuming the law of large numbers. While increasing the size of the image patches (increasing the height of the matrix $X$) can overcome the problems with weighting matrix $A$, an image patch that is too large may include multiple image features, complicating the image model. The optimality of the image patch size is a part of an ongoing research.

We evaluate our work based on the peak signal-to-noise-ratio (PSNR) and on
the structural similarity index (SSIM) [70]. Generally, SSIM yields an image quality and performance measure that is far more reliable than PSNR. Tables 4.3 and 4.4 show PSNR and SSIM values for denoising images corrupted by signal-independent and dependent noise at different variances, respectively. Sample output images from the proposed algorithm, shown in figures 4.6, 4.7, and 4.8, confirm that the algorithm preserves details and sharpness in edges, while the homogeneous regions are smooth.

We compared our method to some works published recently [45] [51] [49] [53] [54]. In our first experiment, images were corrupted by signal-independent noise. All algorithms were provided with a priori parameter, $k_0$. The PSNR values presented in table 4.3 show that the performance of our algorithm, given an image corrupted by signal-independent noise, is comparable to the state-of-the-art denoising methods. In most cases, however, the SSIM values presented in table 4.4 show that the performance of our algorithm, given an image corrupted by signal-independent noise, is better than the state-of-the-art denoising methods. Visual inspection, such as the images in figure 4.6, confirms this fact.

In our second experiment, images were corrupted by signal-dependent noise. Works by [49] [51] [54] were developed under the assumption that images are corrupted by signal-independent noise. Figure 4.5 is an illustration pointing out how an algorithm designed for signal-dependent noise works differently from algorithms designed for signal-independent noise. The solid line represents the standard deviation of signal-dependent noise in an image. The figure reveals that the magnitude of the error from methods in [49] [51] [54] is also dependent on the signal value. In fact, their error graphs are parallel to $k_0 + k_1 s$, indicating that the signal dependent component of the noise is left untouched. On the other hand, the error graph from
Table 4.3: Denoising methods evaluated using PSNR. Images corrupted by noise generated by \((k_0, k_1) = (25, 0), (25, 0.1), (25, 0.2)\), respectively. Method in [9] was combined with methods in [51] [49] [54] [45] to account for signal-dependency of noise (see text).

<table>
<thead>
<tr>
<th></th>
<th>noisy</th>
<th>proposed</th>
<th>[51]</th>
<th>[49]</th>
<th>[54]</th>
<th>[45]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lena</td>
<td>20.17</td>
<td>31.64</td>
<td>31.70</td>
<td>31.18</td>
<td>31.30</td>
<td>30.99</td>
</tr>
<tr>
<td></td>
<td>16.87</td>
<td>29.89</td>
<td>29.46</td>
<td>28.95</td>
<td>29.14</td>
<td>28.66</td>
</tr>
<tr>
<td>Barbara</td>
<td>20.17</td>
<td>29.85</td>
<td>29.13</td>
<td>28.08</td>
<td>28.78</td>
<td>29.12</td>
</tr>
<tr>
<td></td>
<td>17.08</td>
<td>27.62</td>
<td>26.52</td>
<td>25.42</td>
<td>26.37</td>
<td>26.58</td>
</tr>
<tr>
<td></td>
<td>14.85</td>
<td>25.87</td>
<td>24.05</td>
<td>23.19</td>
<td>24.03</td>
<td>24.09</td>
</tr>
<tr>
<td>Boats</td>
<td>20.17</td>
<td>29.16</td>
<td>29.36</td>
<td>29.19</td>
<td>29.01</td>
<td>28.66</td>
</tr>
<tr>
<td></td>
<td>16.71</td>
<td>27.28</td>
<td>27.23</td>
<td>27.02</td>
<td>26.88</td>
<td>26.49</td>
</tr>
<tr>
<td></td>
<td>14.32</td>
<td>25.94</td>
<td>25.00</td>
<td>24.82</td>
<td>24.66</td>
<td>24.31</td>
</tr>
<tr>
<td>House</td>
<td>20.17</td>
<td>31.53</td>
<td>31.37</td>
<td>30.78</td>
<td>30.64</td>
<td>30.77</td>
</tr>
<tr>
<td></td>
<td>16.59</td>
<td>29.38</td>
<td>28.71</td>
<td>28.10</td>
<td>28.17</td>
<td>28.05</td>
</tr>
<tr>
<td></td>
<td>14.23</td>
<td>27.50</td>
<td>25.60</td>
<td>25.11</td>
<td>25.17</td>
<td>24.99</td>
</tr>
<tr>
<td>Peppers</td>
<td>20.17</td>
<td>29.32</td>
<td>29.47</td>
<td>29.36</td>
<td>29.08</td>
<td>28.68</td>
</tr>
<tr>
<td></td>
<td>16.87</td>
<td>27.47</td>
<td>27.15</td>
<td>27.00</td>
<td>26.84</td>
<td>26.47</td>
</tr>
<tr>
<td></td>
<td>14.60</td>
<td>26.05</td>
<td>24.81</td>
<td>24.72</td>
<td>24.50</td>
<td>24.16</td>
</tr>
<tr>
<td>F Print</td>
<td>20.17</td>
<td>27.43</td>
<td>27.44</td>
<td>27.03</td>
<td>27.34</td>
<td>26.86</td>
</tr>
<tr>
<td></td>
<td>16.66</td>
<td>25.00</td>
<td>24.90</td>
<td>24.34</td>
<td>24.78</td>
<td>24.35</td>
</tr>
<tr>
<td></td>
<td>14.28</td>
<td>22.98</td>
<td>22.45</td>
<td>21.79</td>
<td>22.29</td>
<td>21.72</td>
</tr>
</tbody>
</table>
Table 4.4: Denoising methods evaluated using SSIM. Images corrupted by noise generated by \((k_0, k_1) = (25, 0), (25, 0.1), (25, 0.2)\), respectively. Method in [9] was combined with methods in [51] [49] [54] [45] to account for signal-dependency of noise (see text).

<table>
<thead>
<tr>
<th>Image</th>
<th>noisy</th>
<th>proposed</th>
<th>[51]</th>
<th>[49]</th>
<th>[54]</th>
<th>[45]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lena</td>
<td>0.2734</td>
<td>0.8528</td>
<td>0.8514</td>
<td>0.8446</td>
<td>0.8397</td>
<td>0.8278</td>
</tr>
<tr>
<td></td>
<td>0.1784</td>
<td>0.8228</td>
<td>0.8169</td>
<td>0.8066</td>
<td>0.7989</td>
<td>0.7803</td>
</tr>
<tr>
<td></td>
<td>0.1301</td>
<td>0.7969</td>
<td>0.7790</td>
<td>0.7688</td>
<td>0.7483</td>
<td>0.7269</td>
</tr>
<tr>
<td>Barbara</td>
<td>0.4055</td>
<td>0.8657</td>
<td>0.8420</td>
<td>0.8213</td>
<td>0.8379</td>
<td>0.8435</td>
</tr>
<tr>
<td></td>
<td>0.2888</td>
<td>0.8079</td>
<td>0.7737</td>
<td>0.7396</td>
<td>0.7739</td>
<td>0.7765</td>
</tr>
<tr>
<td></td>
<td>0.2181</td>
<td>0.7555</td>
<td>0.7073</td>
<td>0.6664</td>
<td>0.7081</td>
<td>0.7084</td>
</tr>
<tr>
<td>Boats</td>
<td>0.3494</td>
<td>0.7816</td>
<td>0.7856</td>
<td>0.7790</td>
<td>0.7724</td>
<td>0.7567</td>
</tr>
<tr>
<td></td>
<td>0.2293</td>
<td>0.7199</td>
<td>0.7275</td>
<td>0.7187</td>
<td>0.7085</td>
<td>0.6858</td>
</tr>
<tr>
<td></td>
<td>0.1677</td>
<td>0.6742</td>
<td>0.6752</td>
<td>0.6668</td>
<td>0.6479</td>
<td>0.6218</td>
</tr>
<tr>
<td>House</td>
<td>0.2799</td>
<td>0.8378</td>
<td>0.8319</td>
<td>0.8293</td>
<td>0.8045</td>
<td>0.8131</td>
</tr>
<tr>
<td></td>
<td>0.1818</td>
<td>0.8086</td>
<td>0.7981</td>
<td>0.7898</td>
<td>0.7649</td>
<td>0.7709</td>
</tr>
<tr>
<td></td>
<td>0.1300</td>
<td>0.7824</td>
<td>0.7609</td>
<td>0.7491</td>
<td>0.7156</td>
<td>0.7183</td>
</tr>
<tr>
<td>Peppers</td>
<td>0.3542</td>
<td>0.8451</td>
<td>0.8427</td>
<td>0.8510</td>
<td>0.8171</td>
<td>0.8170</td>
</tr>
<tr>
<td></td>
<td>0.2472</td>
<td>0.8050</td>
<td>0.7955</td>
<td>0.8021</td>
<td>0.7608</td>
<td>0.7590</td>
</tr>
<tr>
<td></td>
<td>0.1887</td>
<td>0.7737</td>
<td>0.7514</td>
<td>0.7587</td>
<td>0.7053</td>
<td>0.7035</td>
</tr>
<tr>
<td>F Prints</td>
<td>0.6939</td>
<td>0.9030</td>
<td>0.9038</td>
<td>0.8922</td>
<td>0.9066</td>
<td>0.8897</td>
</tr>
<tr>
<td></td>
<td>0.5168</td>
<td>0.8469</td>
<td>0.8499</td>
<td>0.8302</td>
<td>0.8505</td>
<td>0.8278</td>
</tr>
<tr>
<td></td>
<td>0.3920</td>
<td>0.7779</td>
<td>0.7927</td>
<td>0.7588</td>
<td>0.7879</td>
<td>0.7498</td>
</tr>
</tbody>
</table>
the proposed algorithm flattens out horizontally, suggesting that the magnitude of error is signal-independent.

In order to compare the proposed method to existing algorithms fairly, then, we must account for the signal dependency of the noise. For the rest of the paper, the methods in [45] [51] [49] [53] [54] are combined with generalized homomorphic operator [9]. This operator, \( g(\cdot) \), approximately decouples the noise from the signal. See Appendix B for details. We took the following measures:

1. Transform the corrupted image using a generalized homomorphic operator:
   \[
   x_g = g(x) \approx g(s) + \delta.
   \]

2. Apply the denoising algorithm to \( x_g \) (algorithms were provided with the information that variance is 1). Refer to the output image from this step as \( \hat{s}_g \).

3. Invert the generalized homomorphic operator: \( \hat{s} = g^{-1}(\hat{s}_g) \).

Given signal-dependent noise, the PSNR and SSIM values presented in tables 4.3 and 4.4 show that our algorithm is a clear improvement over other published denoising methods.

Figures 4.6-4.8 highlights some advantages to the proposed algorithm. In our first example (figure 4.6), we show the output images from different algorithms when the input image, cropped from the image commonly known as “Lena,” is corrupted with signal-independent noise, \( k_0 = 25 \). At this noise level, it is difficult see the differences between the algorithms, although the preservation of the texture on the hat is noteworthy. In our second example (figure 4.7), the input image is corrupted with more severe signal-independent noise, \( k_0 = 50 \). The proposed algorithm generally outputs smoother surfaces in homogeneous regions, and
preserves sharper edges in detailed regions. While all denoising algorithms studied in this section suffer from artifacts under severe noise, it is important to point out the differences in the types of artifacts. On occasion, the output images from the proposed algorithm, while preserving the edges, may exhibit a highly structured artifact similar to scratch marks. It occurs when the assumption (4.9) fails because the ensemble average of the noise in an image patch may not be exactly zero. It can be corrected by increasing the size of the image patch. On the other hand, wavelet-based algorithms suffer from low-frequency noise, edge ringing, and blurriness. Methods in [49] and [54] exhibit occasional wavelet “blips,” or isolated instances of large-magnitude wavelet coefficients in the homogeneous regions. The method in [45] suffers from over-smoothing and more severe scratch marks.

Finally, figure 4.8 emphasizes the importance of considering signal-dependent noise when designing a denoising algorithm. Here, the input image is corrupted with severe signal-dependent noise, \( k_0 = 15, k_1 = 0.4 \). Though the edges are not very clean, the performance of the proposed algorithm is acceptable as a denoising algorithm. On the other hand, the methods in [45] [51] [49] [53] [54] fail when the coupling of signal and noise is strong (i.e. large \( k_1 \)). Although the monotonic function, \( g(\cdot) \), approximately decouples the noise from signal, the structure in \( g(s) \) now differ greatly from the original image \( s \). When the signal-dependence of the noise is weaker (e.g. \( k_1 = 0.1 \)), all of the algorithms considered in this section demonstrated acceptable performance.

### 4.3.1 Computational Complexity

Without question, the bottleneck to the proposed algorithm is the eigen decomposition of the positive-definite matrix, \( P = V\Sigma V^T \in \mathbb{R}^{2n \times 2n} \). Generally speaking,
the eigen decomposition of an $2n \times 2n$ symmetric matrix takes $O(4n^2)$ operations.

In this section, we offer two techniques that help speed up this computation.

**Matrix Simplification**

In certain cases, we can exploit the internal structure of $P$ matrix to reduce the complexity of its eigen decomposition significantly. Suppose $P$ is composed of the following sub-matrices:

$$
P = \begin{bmatrix} P_{XX} & P_{XX} - k_0I \\ P_{XX} - k_0I & P_{XX} - k_0I \end{bmatrix}.
$$

In this case, instead of computing the eigenvectors of $P$ directly, we propose to compute the eigenvectors of $P_{XX}$, instead. Then

$$
P = \begin{bmatrix} W & 0 \\ 0 & W \end{bmatrix} \begin{bmatrix} T & T - k_0I \\ T - k_0I & T - k_0I \end{bmatrix} \begin{bmatrix} W & 0 \\ 0 & W \end{bmatrix}^T,
$$

where $P_{XX} = WTW^T$ is an eigen decomposition, $T = \text{diag}(\tau_1, \ldots, \tau_n)$. The rows and the columns of this matrix decomposition can be reordered as follows:

$$
P = \begin{bmatrix} W & 0 \\ 0 & W \end{bmatrix} M_1 \text{diag} \left( \begin{bmatrix} \tau_1 & \tau_1 - k_0 \\ \tau_1 - k_0 & \tau_1 - k_0 \end{bmatrix}, \ldots, \begin{bmatrix} \tau_n & \tau_n - k_0 \\ \tau_n - k_0 & \tau_n - k_0 \end{bmatrix} \right) M_1^T,
$$

where $M_1 \in \mathbb{R}^{n\times n}$ is a permutation matrix. We now consider eigen decomposition of $[\tau_i, \tau_i - k_0; \tau_i - k_0, \tau_i - k_0]$, which can be solved in fixed-time. It is therefore clear that a series of Givens rotation matrices will diagonalize the above matrix decomposition [15]:

$$
P = \begin{bmatrix} W & 0 \\ 0 & W \end{bmatrix} M_1 G_1 \ldots G_n M_2 \Sigma M_2^T G_n^T \ldots G_1^T M_1^T \begin{bmatrix} W & 0 \\ 0 & W \end{bmatrix}^T,
$$
where $M_2 \in \mathbb{R}^{n \times n}$ is a permutation matrix that sorts the eigenvalues to descending order, and $G_i \in \mathbb{R}^{n \times n}$ is Givens rotation matrix whose entries are defined by the diagonalization of $[\tau_i, \tau_i-k_0; \tau_i-k_0, \tau_i-k_0]$. Because $W$, the permutation matrices, and Givens rotation matrices are orthonormal transformations,

$$V = \begin{bmatrix} W & 0 \\ 0 & W \end{bmatrix} M_1 G_1 \ldots G_n M_2.$$

Furthermore, the eigen decomposition of $P_{XX}$ can be found efficiently using linear algebra based on rank-one matrix updates and Newton’s method [15] [5]. The rank-one matrix update is based on the fact that most of the adjacent image patches overlap greatly, and it is far more efficient than computing the eigen decomposition of $P$ at every pixel location.

**Image Patch Omission**

The dimension of the matrix $P$ is determined by $n$, the number of image patches present in $\{x_i\}$. However, it is argued in section 4.2.2 that the image patches in $\{x_i\}$ are not always similar to $s_0$. Rather than limiting the contributions from the poorly selected image patches through adaptive weights (soft image patch selection), we may consider omitting them altogether and reducing the computational cost.

Figure 4.9 shows the image quality trade-offs for using fewer image patches in the proposed algorithm. The weights corresponding to the image patches (i.e. $b_i$ in (4.14)) are sorted, and only the image patches with $n$ largest weights are kept. The graph clearly shows that when $n$ is small, the complexity for eigen decomposition of $P$ decreases significantly while the image quality suffers.
Figure 4.9: SSIM for images denoised using fewer image patches by omitting image patches that are poor representations of $s_0$. Noise generated with $(k_0, k_1) = (25, 0)$. Circle represents ‘lena’ image, triangle represents ‘barbara’ image. See text.

4.4 Summary

In this chapter, a new image denoising algorithm based on TLS techniques was presented. An ideal image patch was modeled as a linear combination of vectors cropped from the noisy image, and we fit the model to the real image data by allowing a small perturbation in the TLS sense. A new technique to solve the TLS problem without the knowledge of the ideal image patch when the image is corrupted by signal-dependent noise is developed. The output image quality improved significantly by introducing decimated image patches, adaptive weighting schemes, and redundant estimation techniques.

The output images from the proposed algorithm showed improved image quality, when compared to recently published work. In the case that an image is corrupted by a signal-independent noise, the image quality was comparable according to SSIM values, and it preserved edge structure better under severe noise. In the case that an image is corrupted by a signal-dependent noise, the proposed
algorithm produced acceptable results.

Future research in this field includes reduction of computational complexity and development of a more sophisticated weighting scheme.
CHAPTER 5
JOINT DEMOSAICING AND DENOISING

A typical digital camera is subject to influences from noise in the image sensor. This sensor noise, often characterized as signal-dependent noise, is amplified by a series of image processing steps needed to produce a full-color representation of an image displayable on a computer monitor or a printer. This phenomenon is especially evident when taking a picture in a low-light environment, and it is one of the major problems noticeable in commercial digital cameras today.

A cost-effective digital camera uses a single-chip image sensor with color filter array. Demosaicing algorithm reconstructs a full three-color representation of color image by estimating the missing pixel components from the CFA sampling pattern. The source of conflict between the image processing pipeline and image sensor noise is this demosaicing step. Often in demosaicing, we would like to preserve the sharpness of the edges while interpolating the missing pixel components. In the presence of noise, however, noise patterns form false edge structures, sharpening amplifies high frequency noise, and interpolation adds a structure to the noise too complicated to analyze. Removing noise after demosaicing, therefore, is impractical. Removing noise before the image processing pipeline is equally problematic because determining an image structure, necessary for effective noise reduction, from a sparse sampling lattice is difficult.

Many demosaicing algorithms have been published in recent years [16] [17] [40] [41] [72]. Although the output images from these algorithms are impressive in the absence of sensor noise, none of them address the image sensor noise problem explicitly (to the best of the knowledge of the author). Table 5.1 evaluates the performance of the demosaicing algorithms using S-CIELAB, both in the presence
and the absence of sensor noise. It shows a severe amplification of noise at the output in the presence of sensor noise.

In recent years, a considerable amount of work has been done on denoising an image corrupted by signal-independent noise, and very little attention has been given to denoising of a Bayer pattern image. While they are useful general methods, most algorithms neither take into account signal-dependent noise models nor accommodate CFA sampling patterns. Even so, suppose we apply the state-of-art denoising methods after the demosaicing step. Limiting the study to images corrupted by signal-independent noise, table 5.1 shows the performance of denoising methods when taking the output image from demosaicing algorithms as their input. The denoised images are generally still noisy, although they show improvements.

Noting that image interpolation and image denoising are both estimation problems, this thesis proposes a unified approach to performing demosaicing and image denoising simultaneously. The novelty of our work is the development of a constraint, under which an optimal filter for estimating a pixel value from a noisy single-color image is also an optimal filter for demosaicing given noisy sensor data. Furthermore, many existing image denoising algorithms can be combined with the demosaicing operation using this proposed technique because this constraint is not very restrictive. For example, one may choose bilateral filtering because of computational efficiency, while another may choose a more sophisticated image denoising method for higher image quality.

Performing demosaicing and image denoising simultaneously has various advantages over treating these problems separately. First, image quality is improved because we can tune the filter coefficients such that the edge structures are preserved without amplifying noise. Second, the estimation of the missing pixel com-
ponents may explicitly incorporate the noise characteristics of the sensor. Finally, the combined algorithm reduces the computational complexity compared to performing the procedures independently.

This chapter is organized as follows. Section 5.1 provides motivation for combining denoising and demosaicing methods by posing the problem as a filter design. In section 5.2, an example demosaicing-denoising method is developed using the proposed technique. Section 5.3 presents experimental results using pseudo-random and real sensor noise.

5.1 Filter Design

In this section, motivation for combining denoising and demosaicing methods is considered (the discussion is independent of the choice of the denoising algorithm). In this thesis, the task of estimating pixel values from sparsely-sampled noisy sensor data is treated as a filter-design problem.

Let \( R, G, \) and \( B \) be the noise-free red, green, and blue images, respectively. Define \( R_s, G_s, \) and \( B_s \) as the red, green, and blue pixel values sampled by the image sensor according to the CFA pattern. In this thesis, we work with a Bayer pattern CFA, although the results extend to more general cases [1]. For ease of notation, let \( X = R_s \cup G_s \cup B_s \) be the ideal image sensor output. We assume the signal-dependent noise model in a CMOS image sensor verified in section 3.1:

\[
Y(i, j) = X(i, j) + (k_0 + k_1X(i, j))\delta(i, j),
\]

where \( \delta(i, j) \sim \mathcal{N}(0, 1) \) is noise, and \( k_0, k_1 \in \mathbb{R} \) are parameters. In other words, \( Y \) is the measured value or the noisy image sensor output. Our objective is to estimate \( R, G, \) and \( B \) given \( Y \). In this section, a technique to estimate \( G \) from \( Y \) is presented (estimation of \( R \) and \( B \) is done in the same manner).
Consider a $n \times n$ window cropped from noisy sensor values, $Y$, as in figure 5.1. Let us call this image patch $y_0$ (the pixel at the center of this patch is $y_0(0,0)$), and the corresponding ideal (i.e. noise free) sensor values $x_0$. Suppose the we are interested in estimating the ideal green pixel value at the center ($\hat{G}(0,0)$) by taking a linear combination of the measured values in this window:

$$\hat{G}(0,0) = \sum_{i,j} \alpha(i,j) y_0(i,j)$$  \hspace{1cm} (5.2)$$

Note that even if the center of $y_0$ is green, we must still estimate the noise-free green pixel value. Therefore unlike the demosaicing problem, the above formula applies regardless of the color of the noisy center pixel $y_0(0,0)$ (i.e. we do not draw a distinction between the estimation of a missing pixel component from noisy data and the estimation of ideal pixel value when the noisy pixel value is already given). One obvious approach to choosing $\alpha$ is to treat each color plane separately, i.e. use only green pixels to estimate $G(0,0)$. However, many have argued that this is ineffective because it does not take advantage of the spatial redundancies between the different colors [16] [17] [32] [40] [72].

We instead begin by assuming that the difference images $R - G, B - G, R - B$ are bandlimited signals [20]. For example, figure 5.2 shows $R, G$, and $R - G$
images. While $R$ and $G$ are sharp images, edge information in $R - G$ image is fairly smooth. Thus, we assume

$$\sum_{i,j} h(i,j)(R(i,j) - G(i,j)) \simeq 0$$

$$\sum_{i,j} h(i,j)R(i,j) \simeq \sum_{i,j} h(i,j)G(i,j)$$

where $h(\cdot, \cdot)$ is a highpass filter. This is equivalent to stating that the high-frequency components of $R$ and $G$ (and $B$ likewise) are similar, while the low-frequency components may be dissimilar. Therefore, we impose a constraint that the coefficients corresponding to noisy red and blue values ($\{\alpha(i,j)\}_{i,j\in\{-2,0,2\}}$ and $\{\alpha(i,j)\}_{i,j\in\{-1,1\}}$ in figure 5.1, respectively) add up to 0 when estimating $G(0,0)$, respectively. These coefficients are high-pass filters, effectively, and this guarantees that the low-frequency components of $R_s$ and $B_s$ do not contribute to the estimation of $G$.

But what should the filter coefficients be? Since only the high frequency components of $R_s$ and $B_s$ are passed by the filter $\{\alpha(i,j)\}$, we make the following
substitution:

\[ \hat{G}(0, 0) = \sum_{i,j} \alpha(i, j)y_0(i, j) \]
\[ = \sum_{i,j} \alpha(i, j)[G(i, j) + (k_0 + k_1x_0(i, j))\delta(i, j)]. \]

This substitution suggests that we design filter coefficients \( \alpha(i, j) \) as if we are optimally estimating a pixel value from a noisy single-color image.

Because the single-color image is unavailable from the noisy sensor data \( Y \), we adapt another generalization, motivated by multi-resolution analysis and wavelets [21]. If \( \alpha \) is chosen such that

\[ G(0, 0) \simeq \sum_{i,j} \alpha(i, j)[G(2i, 2j) + (k_0 + k_1x_0(i, j))\delta(i, j)], \]

then

\[ G(0, 0) \simeq \sum_{i,j} \alpha(i, j)[G(i, j) + (k_0 + k_1x_0(i, j))\delta(i, j)]. \]

That is, the filter \( \alpha \) designed to estimate \( G(0, 0) \) from the downsampled green image, \( G(2i, 2j) \), would also yield a satisfactory estimate if applied to full-resolution green image, \( G(i, j) \). Working with \( G(2i, 2j) \) rather than with \( G(i, j) \) is convenient because downsampling \( Y \) by two in horizontal and vertical directions yields two smaller green images.

To summarize, the strategy for choosing the filter coefficients to estimate \( G(0, 0) \), regardless of the color of \( y_0(0, 0) \), consists of three major steps:

1. Design filter \( \alpha \) as if we are estimating \( G(0, 0) \) by taking a linear combination of \( \{G(2i, 2j)\} \).

2. Add a restriction to the filter such that the coefficients corresponding to the noisy red and blue values add up to zero, respectively.
3. Apply the filter to noisy image sensor output $Y$ using (5.2).

We remind the readers that the same technique is used for estimating $R(0,0)$ and $B(0,0)$ from $Y$.

### 5.2 Denoising Method

We are left with the task of designing a denoising algorithm that will fulfill the constraints outlined in section 5.1. There are many existing image denoising algorithms that are compatible with these constraints, offering flexibilities and choices in the design of an image processing pipeline. For example, a simple image denoising method, such as bilateral filtering [61], may be combined with demosaicing procedure when the computational resource is limited. On the other hand, combining a more sophisticated image denoising method, such as [45] and [21], with demosaicing procedure may yield improved image quality.

In this section, we describe a demosaicing algorithm based on a total least squares TLS image denoising method developed in chapter 4. The following discussions are intended as a proof-of-concept case study, and so the choice of denoising method is not unique. Again, we focus exclusively on the linear estimation of $G(0,0)$, although the same techniques are used to estimate $R(0,0)$ and $B(0,0)$.

#### 5.2.1 TLS Denoising Problem

In this section, we are interested in designing a filter $\alpha$ such that $\hat{G}(0,0) = \sum \alpha(i,j)[G(2i,2j) + (k_0 + k_1x_0(i,j))\delta(i,j)]$ is an optimal estimate of $G(0,0)$ in the TLS sense.

Let $G_1(i,j)$ and $G_2(i,j)$ be the two noisy green images obtained from down-
sampling $Y$ by 2 in both horizontal and vertical directions. Define \{${y_1, \ldots, y_m}$\} as a set of vectorized $n \times n$ image patches cropped from $G_1$ and $G_2$, let \{${x_1, \ldots, x_m}$\} be their corresponding ideal green image patches (i.e. noise free), respectively, and

$$z_k = x_k + k_0\delta_k + k_1\text{diag}(x_0)\delta_k,$$  \hspace{1cm} (5.3)

where $\delta_k$ is a noise vector, and $x_0$ is as before. Suppose the filter coefficients $\alpha \in \mathbb{R}^{n^2}$ are designed such that for all $k$, $z_k^T \alpha$ is an optimal estimate for the center value in $x_k$. If this family of image patches is similar to $y_0$ then it is reasonable to assume that $\alpha$ will be a good filter for (5.2), also. A measure of similarity will be introduced below.

Define $X^g = [x_1, \ldots, x_m]^T$, $Y^g = [y_1, \ldots, y_m]^T$, $Z^g = [z_1, \ldots, z_m]^T$, and let $x^g$ be the column in $X^g$ that corresponds to the center pixels of \{${x_1, \ldots, x_m}$\}. In order that $Z^g \alpha$ be an optimal estimate for $x^g$ in the TLS sense, $\alpha$ must solve the following criterion:

$$\min ||A[E, e_0]M^T B||_F^2,$$  \hspace{1cm} (5.4)

subject to

$$(Z^g + E)\alpha = x^g + e_0$$

where $\alpha$ takes the form

$$\alpha = M\beta.$$  \hspace{1cm} (5.5)

Note that $M \in \mathbb{R}^{n^2+1 \times n^2-1}$ restricts $\alpha$ to the subspace spanned by the columns of $M$. This is convenient for constraining $\alpha$ such that coefficients corresponding to red and blue pixels add up to zero, respectively. An example of $M$ matrix is
(assumes $y_k$ is vectorized such that like colors are grouped together):

$$M = \begin{bmatrix} M_R & M_G & M_B & 1 \end{bmatrix},$$

where $M_G = I$, $M_R, M_B = [I; -1, \ldots, -1]$, and $I$ is an identity matrix. While it is possible to solve for $\beta$ that minimizes $\|A[E, e_0]B\|_F$, $M$ is figured into the cost function (5.4) because it simplifies the solution to $\beta$ significantly. Our strategy is to solve for optimal $\beta$, and set $\alpha = M\beta$.

A variation of the TLS problem (5.4) using an affine approximation model was solved by de Groen [8]. He showed that the cost function, $\|A[E, e]MB\|_F^2$, is reduced greatly when the column-means of $A[Z^g, x^g]MB$ are subtracted from their respective columns first, suggesting a better model fit. In this paper, we modify the approach outlined in section 5.1 to take advantage of the affine approximation technique.

More specifically, instead of (5.4), we solve for $\alpha$ in the system that minimizes $\|A[E, e]MB\|_F^2$ subject to $(\tilde{Z}^g + E)\alpha = \tilde{x}^g + e_0$. Here, $\tilde{Z}^g = Z^g - [1, \ldots, 1]^T\bar{z}$ and $\tilde{x}^g = x^g - [1, \ldots, 1]^T\bar{x}^g$, where the entries in $\bar{z}$ are the average values of columns in $Z$, respectively, and $\bar{x}$ is the average value of $x^g$. $\tilde{X}^g$ and $\tilde{Y}^g$ are defined similarly. Note that the average of the column in $Y^g$ corresponding to the center pixel is a good approximation for $\bar{x}$ (see (5.3)). Once $\alpha$ is solved, our optimal estimate for $x^g$ is:

$$\hat{x}^g = \tilde{Z}^g\alpha + \bar{x}^g.$$

More importantly, let $\bar{y}_0 \in \mathbb{R}^{n^2}$ be the vector average of $n \times n$ image patches cropped from noisy sensor output $Y$ that are in the spatial vicinity of $y_0$ and
whose locations of red and blue pixels match that of $y_0$. Our best estimate for $G(0, 0)$ is

$$\hat{G}(0, 0) = \tilde{y}_0 \alpha + \tilde{x}^g,$$

where $\tilde{y}_0 = y_0 - \bar{y}_0$.

### 5.2.2 Solution to TLS

Solving the TLS system above is straightforward [21]. Assume $A = \text{diag}(a_1, \ldots, a_m)$ and $B = \text{diag}(b_1, \ldots, b_{n^2 - 1})$ and let $N = n^2 - 1$. Using singular value decomposition $A[\tilde{Z}^g, \tilde{x}^g]B = U\Sigma V^T$, where $\Sigma = \text{diag}(\sigma_1, \ldots, \sigma_N)$ and $\sigma_k^2 > \sigma_{k+1}^2$, $\beta$ that solves (5.4) is [15]:

$$\beta = -\text{diag}(b_1, \ldots, b_{N-1}) \begin{bmatrix} v_{1,N} \\ \vdots \\ v_{N-1,N} \end{bmatrix} v_{N,N}^{-1} b_N, \quad (5.6)$$

where $[v_{1,N}, \ldots, v_{1,N}]^T$ is the right singular vector corresponding to $\sigma_N$. However, $\tilde{x}^g$ is not available in the denoising problem, thus making it difficult to compute $V$ from singular value decomposition. Instead, define the matrix $P$:

$$P = (A[\tilde{Z}^g, \tilde{x}^g]MB)^T (A[\tilde{Z}^g, \tilde{x}^g]MB)$$

$$= (U\Sigma V^T)^T (U\Sigma V^T) = V\Sigma^2 V^T.$$  

Our strategy is to estimate $P$ and obtain the right singular vector $V$ through its eigen decomposition.

Note that $\mathcal{E}\{\delta_k\} = 0$ and $\mathcal{E}\{\delta_k \delta_l^T\} = I$ if $k = l$ and 0 if otherwise. When
\[ m \gg N, \quad P = \mathcal{E}\{P\}, \text{ and} \]

\[
P = \mathcal{E}\{(A[\tilde{Z}^g, \tilde{x}^g]MB)^T(A[\tilde{Z}^g, \tilde{x}^g]MB)\}
\]

\[
= B^T M^T \begin{bmatrix}
P_{ZZ} & \tilde{X}^g T A^2 \tilde{x}^g \\
\tilde{x}^g T A^2 \tilde{X}^g & \tilde{x}^g T A^2 \tilde{x}^g
\end{bmatrix} MB, \quad (5.7)
\]

where \( P_{ZZ} = \mathcal{E}\{\tilde{Z}^g T A^2 \tilde{Z}^g\} \). With some manipulations, \( P_{ZZ} \) simplifies to:

\[
P_{ZZ} = \tilde{X}^g T A^2 \tilde{X}^g + \text{diag}(k_0 + k_1 x_i)^2 \left( \sum_{i=1}^{m} a_i^2 \right). \quad (5.8)
\]

Given \( Y \), \( P \) can be estimated. Let \( P_{YY} = \tilde{Y}^g T A^2 \tilde{Y}^g \), and \( \tilde{x}_i \) and \( \tilde{y}_i \) are the \( i \)th row of \( \tilde{X}^g \) and \( \tilde{Y}^g \), respectively (hence \( x_i = \bar{x} + \tilde{x}_i \)). For \( m \gg N \), \( P_{YY} = \mathcal{E}\{P_{YY}\} \), and

\[
P_{YY} = \tilde{X}^g T A^2 \tilde{X}^g + \sum_{i=1}^{m} \left( a_i^2 \text{diag}(k_0 + k_1 x_i)^2 \right)
\]

\[
= \tilde{X}^g T A^2 \tilde{X}^g + \sum_{i=1}^{m} \left( a_i^2 \text{diag}(k_0 + k_1 \bar{x} + k_1 \tilde{x}_i)^2 \right)
\]

\[
= \tilde{X}^g T A^2 \tilde{X}^g + \sum_{i=1}^{m} \left( a_i^2 \text{diag}(k_0 + k_1 \bar{x})^2 + \sum_{i=1}^{m} a_i^2 k_1^2 \text{diag}(\tilde{x}_i)^2 \right)
\]

\[
+ 2 \sum_{i=1}^{m} a_i^2 k_1 \text{diag}(k_0 + k_1 \bar{x}) \text{diag}(\tilde{x}_i).
\]

Using the substitution \( \mathcal{E}\{\sum_i a_i^2 \tilde{y}_i\} = \sum_i a_i^2 \tilde{x}_i \) and the fact that the diagonal entries of \( \tilde{X}^g T A^2 \tilde{X}^g \) and \( \sum_i a_i^2 \text{diag}(\tilde{x}_i)^2 \) are identical, \( \tilde{X}^g T A^2 \tilde{X}^g \) can be estimated using the following procedure:

1. Compute \( P_{YY} = \tilde{Y}^g T A^2 \tilde{Y}^g \).

2. Compute \( P_{YY} - \sum a_i^2 [\text{diag}(k_0 + k_1 \bar{x})^2 + 2k_1 \text{diag}(k_0 + k_1 \bar{x}) \text{diag}(\tilde{y}_i)] \).

3. Multiply the diagonal entries of step 2 by \((1 + k_1^2)^{-1}\).
Let us call this estimate $P_{XX}$. The estimates of $\tilde{X}^T A^2 \tilde{x}^g$, $\tilde{x}^g T A^2 \tilde{X}$, and $\tilde{x}^g T A^2 \tilde{x}^g$ are obtained by taking appropriate rows and columns of $P_{XX}$. $P_{ZZ}$ is computed from $P_{XX}$ using (5.8) and exchanging $\tilde{y}_0$ in lieu of $x_0$ (this substitution is justified in the previous chapter). Thus, matrix $P$ is fully computable.

The filter coefficients $\alpha$ are computed from (5.6) and (5.5), where $V$ is given by the eigen decomposition of $P$ in (5.7). This $\alpha$ solves (5.4) subject to $(\tilde{Z}^g + E)\alpha = \tilde{x}^g + e_0$. Our best estimate for $G(0, 0)$ is

$$\hat{G}(0, 0) = \tilde{y}_0^T \alpha + \tilde{x}^g = \tilde{y}_0^T M \beta + \tilde{x}^g.$$ 

Same technique is used to estimate $R(0, 0)$ and $B(0, 0)$ from noisy sensor data, $Y$. Note that the constraint matrix $M$ would be different for estimating $R(0, 0)$ and $B(0, 0)$.

### 5.2.3 Denoising Improvements

Above, $A = \text{diag}(a_1, \ldots, a_m)$ and $B = \text{diag}(b_1, \ldots, b_N)$ are weighting matrices. In this thesis, the $n \times n$ image patches $\{y_1, \ldots, y_m\}$ are taken from the spatial vicinity of $G(0, 0)$ [21]. However, because natural images most certainly are discontinuous signals, not all image patches share the same image attributes with $y_0$. To prioritize $\{y_1, \ldots, y_m\}$ in the order of similarity, larger weight is given (i.e. larger $a_k$) if $H^T y_k$ is similar to $H^T y_0$. More specifically,

$$a_k = \exp(-(y_0 - y_k)^T H H^T (y_0 - y_k)/k_A)$$

where $k_A \in \mathbb{R}$ is a constant. In our simulation, the use of $B$ did not make much difference. The experimental results shown in this thesis has $b_1, \ldots, b_{N-1} = 1$ and $b_N = 0.5$. 
It is well accepted that the edge information in an image is mostly contained within the high frequency components. In the regions of an image that is full of edges, it is reasonable to assume that high frequency components are the dominant features. In these cases, the benefits to incorporating the decimated red and blue images to the estimation of green pixel value may outweigh the possibility that their low frequency components may be dissimilar to that of the green image. Fortunately, we can use the weighting scheme explained above to decide how much of \( R(2i, 2j) \) and \( B(2i, 2j) \) to incorporate into estimation of \( G(0, 0) \). More specifically, we replace \( \tilde{X}^g, \tilde{Y}^g, \text{ and } \tilde{Z}^g \) above with \([\tilde{X}^r; \tilde{X}^g; \tilde{X}^b], [\tilde{Y}^r; \tilde{Y}^g; \tilde{Y}^b] \text{ and } [\tilde{Z}^r; \tilde{Z}^g; \tilde{Z}^b] \), respectively, and \( \tilde{x}^g \) becomes \([\tilde{x}^r; \tilde{x}^g; \tilde{x}^b] \). The idea is to reduce the corresponding weight \( a_k \) when the high frequency components in the red or blue patches are dissimilar to that of \( y_0 \). Contribution from image patches with small weights to the computation of \( \alpha \) is small, assuring that the red and blue patches are used only if there is a potential benefit to incorporating them to the estimation of \( G(0, 0) \). In this thesis, therefore, \( H \) represents a high-pass filter.

Replacing \( \tilde{X}^g \) with \([\tilde{X}^r; \tilde{X}^g; \tilde{X}^b] \) should not be confused with the fact that filter coefficients designed for estimating \( G(0, 0) \) are still different from the coefficients to estimate \( R(0, 0) \) or \( B(0, 0) \). The constraint matrix \( M \) used to estimate \( G(0, 0) \) is different than that of \( R(0, 0) \) or \( B(0, 0) \), guaranteeing that the low frequency components of the estimated pixel comes from the like-colors only.

### 5.2.4 Pre-Processing

The effectiveness of the TLS denoising algorithm depends on our ability to estimate \( P \) matrix accurately. Given \( \delta(i,j) \sim N(0, 1) \), there will be one or two pixels occasionally that stand out because the value of \( \delta \) at that pixel position is far
greater than its standard deviation. This is problematic because the entries in $Y$ appear more than once, degrading our estimate for $P$ greatly. To work around this problem, we propose to prune the outliers. The following pre-processing procedure was used. For each pixel location in $Y$,

1. Let $w$ be a set of pixels in $Y$ that fall within the $L \times L$ neighborhood of the pixel of interest, and whose color is the same as the pixel of interest.

2. Find the $k$th largest and $k$th smallest pixel values in $w$.

3. If the pixel of interest is larger (smaller) than the $k$th largest (smallest) value in $w$, replace it with the $k$th largest (smallest) pixel value in $w$.

The proposed pre-processing procedure is a particularly good match for working with image sensors. Due to variabilities in manufacturing processes, the image sensors often contain a few defective pixel sensors, sometimes referred to as hot or dead pixels. Because the output values from these pixel sensors do not have any relevance to the image (i.e. an outlier), and because most demosaicing algorithms contain filtering, these pixels can potentially degrade the quality of the output image significantly. Although improvements in the manufacturing process is desirable, designing a digital system to remove the defective pixels can increase the yield. Assuming that the defective pixel sensors do not cluster, the pre-processing procedure above should effectively remove them.

5.3 Implementation and Results

Our TLS algorithm is implemented by taking $5 \times 5$ image patches from a $25 \times 25$ neighborhood. Pre-processing had a window size of $11 \times 11$, and we picked the 4th
smallest and largest pixel values. Parameters $k_0$ and $k_1$ were available \textit{a priori}. When $k_1 = 0$, algorithms to estimate the noise variance have been proposed \cite{31}. In real-world CMOS sensors, however, $k_0$ and $k_1$ depend closely on the programmable amplifiers in the A/D converter. We may assume that the gain for the amplifier is provided, and that relationship between the gain and $k_0, k_1$ is understood from the calibration experiments.

Experiments were performed on color images corrupted according to the CMOS noise model (3.1) using pseudo-random noise sampled according to CFA. Figure 5.3 shows a simulation example. To the best of knowledge of the authors, this is the first study of combining demosaicing and denoising algorithms. We therefore compare our results to the state-of-the-art demosaicing algorithms \cite{16} \cite{20} followed by denoising algorithms \cite{49} \cite{51} \cite{21}. Denoising algorithms were performed on each color plane separately. Note that \cite{51} and \cite{49} are incapable of handling the case when the noise is signal-dependent.

Table 5.1 clearly shows the benefits to considering demosaicing and denoising as a single operation. We note, however, that in the absence of noise, the other demosaicing algorithms may sometimes perform better than the proposed algorithm, and the proposed algorithm occasionally suffers from zippering artifacts (only in the absence of noise). Finally, we show an example output image from the state-of-the-art algorithms in figures 5.3 and 5.4. The amplification of noise is seen due to demosaicing, and while applying a denoising algorithm after demosaicing algorithm helps the overall image quality, the proposed algorithm is both sharper and significantly less noisy. The differences are especially pronounced in the smooth regions of the image.

Experiments were also performed on images taken from Agilent Technologies'
Figure 5.3: Example using “parrot” image with noise \((k_0, k_1) = (25, 0)\): reconstruction using method in [16], method in [20], method in [16] and [49], method in [20] and [49], method in [16] and [51], method in [20] and [51], method in [16] and [21], method in [20] and [21], proposed method.
Figure 5.4: Example using “parrot” image with noise \((k_0, k_1) = (10, 0.1)\): reconstruction using method in [16], method in [20], method in [16] and [21], method in [20] and [21], proposed method.
CMOS APS digital camera in low light. Images were captured in a raw-data format with the same programmable gain amplifier setup used during the calibration process. The parameters for the proposed algorithm were \((k_0, k_1) = (3, 0.02)\). After demosaicing, the images were processed with color space conversion and gamma correction \((\gamma = 1.8)\). The illuminant was known \textit{a priori} and it was considered in the color space conversion step. Figure 5.5 shows examples comparing the proposed method to other demosaicing methods. The demosaicing methods in [16] and [20] maintain high contrast, but the \textit{grainy} noise is highly visible in the dark regions of the image. While the proposed algorithm suffers from occasional zipper artifacts, the \textit{grainy} noise is eliminated well by the proposed algorithm.
Table 5.1: Performance of demosaicing and denoising algorithms on the “parrots” image, evaluated using average SCIELAB error [75]. Noise levels considered were \((k_0, k_1) = (0, 0), (25, 0),\) and \((10, 0.1).\) n/a means not available or not necessary.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>(0, 0) (25, 0) (10, 0.1)</td>
<td>(0, 0) (25, 0) (10, 0.1)</td>
<td>(0, 0) (25, 0) (10, 0.1)</td>
</tr>
<tr>
<td>no denoising</td>
<td>0.8108 6.5052 4.6319</td>
<td>0.7768 6.4220 4.6731</td>
<td>0.9922 3.7504 2.9535</td>
</tr>
<tr>
<td>denoising method in [51]</td>
<td>n/a 4.1660 n/a</td>
<td>n/a 4.2926 n/a</td>
<td>n/a n/a n/a</td>
</tr>
<tr>
<td>denoising method in [49]</td>
<td>n/a 4.1166 n/a</td>
<td>n/a 4.2271 n/a</td>
<td>n/a n/a n/a</td>
</tr>
<tr>
<td>denoising method in [21]</td>
<td>n/a 3.8954 3.0283</td>
<td>n/a 4.0702 3.1912</td>
<td>n/a n/a n/a</td>
</tr>
</tbody>
</table>
5.4 Summary

Noting that image interpolation and image denoising are both estimation problems, this chapter presented a unified method to combine demosaicing and image denoising procedures. The filtering coefficients were restricted such that only the high frequency components of the image signals contribute to the estimation of pixel values of different colors. With substitutions, the multi-colored demosaicing/denoising problem was simplified to a single-color denoising problem. Total least squares algorithm was developed as a proof-of-concept, and the algorithm was tested on color images with pseudo-random noise and on raw sensor data from a real CMOS digital camera. The experimental results verify that performing demosaicing and denoising simultaneously is far more effective than treating the demosaicing and denoising problems separately.
The chromatic adaptation phenomenon poses a particularly challenging problem in digital color imaging. For example, digital cameras measure the light intensity corresponding to the pixel positions in the scene. Because the measured light intensity strongly depends on the illuminant, the captured image often appears different from the scene the photographer sees. Similarly, the same image displayed under different media and under different viewing condition affects the perceived image. The process of adjusting the image appearance to a different viewing condition is commonly known as the white-balance problem.

If the end user of the output image is a human eye, it is clearly important to be sensitive to the HVS chromatic adaptation mechanism. Although a series of recent studies in cognitive science verifies that the induced opponent response theory characterizes the chromatic adaptation process more accurately than the coefficient law [28] [29] [30] [6] [50] [57], most white-balance algorithms developed today are a combination of the von Kries coefficient law and an illuminant estimation technique. Likewise, it has been shown that the HVS is not illuminant invariant [44] [30], yet techniques to estimate the reflectance is popular. Perhaps this is a testimony to the disconnectedness between the engineering field and cognitive science.

In this chapter, we examine a model for the viewing conditions for the photographer and the end user of the digital camera, and propose to formulate the white-balance problem using Jameson and Hurvich’s induced opponent response theory. The solution to this problem also requires the knowledge of the illuminant. In section 6.2, we will compare our technique to the coefficient law-based white-
balance solution using the same illuminant estimation method. We show that the images generated with the white-balance algorithm based on induced opponent response appear more natural.

6.1 White-Balance Algorithm

6.1.1 White-Balance Problem

The problem of color constancy poses a difficult challenge to analog and digital photography. The output color from a camera often differs from how it appeared to the eyes of the photographer that took the picture. Problem of correcting the output color from a camera is called white-balance. In analog photography, the silver halide films are chemically calibrated to perform well when the picture is taken under a certain known illuminant. The photographer must choose the correct type of film. In digital photography, we can perform more complex analysis to the acquired image data and numerically correct the color data. Many digital cameras also allow manual selection of the illuminant.

But what is the criteria for color correction in the white-balance problem? The author believes that this is an issue that is very often misunderstood. Recent vision experiments verify that, contrary to the popular belief, the HVS is not invariant to the illuminant [44]. Mathematically, the color constancy is a likewise poorly posed problem because it can be shown that there are no non-trivial operators $D_1, D_2$ (linear or nonlinear) such that

$$D_1 \Phi(r(\lambda)l_1(\lambda)) = D_2 \Phi(r(\lambda)l_2(\lambda))$$

(6.1)

for all choices of $r(\cdot)$. Therefore, the HVS exhibits only an approximate color constancy, and it is not in our interest to set our goals to achieving color constancy
in digital cameras. With the *a priori* knowledge that the observer of the output image is a human eye, the task of white-balance algorithm is to correct the color such that the output image, when viewed under a standard condition, matches *the scene observed by the photographer’s eye*.

To make this point clear, we refer to the example in figure 6.1. Let $\Psi(l_r, l_s)$ be a HVS response to the focal field light $l_r$ when the eye has adapted to the surrounding light $l_s$. The right system in figure 6.1 shows a scene lit by a red illuminant observed by the photographer’s human eye. Assuming that the eye has adapted to the red illuminant, the system’s HVS response to the color $l(\lambda)$ is

$$\Psi(l, \text{red light}).$$

The left system shows a human eye observing display device viewed under a yellow illuminant. Assuming that the eye has adopted to the yellow illuminant, its HVS response is

$$\Psi(w_1 p_1 + w_2 p_2 + w_3 p_3, \text{yellow light}),$$

where $p_1(\lambda), p_2(\lambda), p_3(\lambda)$ are the spectrum distribution of the primary colors of the phosphors used in the CRT monitor, and $\vec{w} = [w_1, w_2, w_3]^T$ is the tristimulus value controlling the intensity of $p_i(\cdot)$, respectively. Note that $\vec{w}$ is the output from the digital camera. The purpose of the white-balance algorithm is to process sensor data inside the digital camera such that the two systems in figure 6.1 are equivalent. That is, we would like to find $\vec{w}$ such that

$$\Psi(l, \text{red light}) = \Psi(w_1 p_1 + w_2 p_2 + w_3 p_3, \text{yellow light}) \quad (6.2)$$

In the following sections, we discuss the solutions to (6.2). In section 6.1.2, we review a common approaches to solving (6.2), while a new alternative method to solving (6.2) is presented in section 6.1.3.
Figure 6.1: Two different viewing conditions. The purpose of the white-balance algorithm is to make the two scenes appear identical.
6.1.2 Common Approaches to White-Balance

Following the example in figure 6.1, let $l_R$ and $l_Y$ be the spectral density of the red and yellow illuminants in (6.2), respectively. In this section, we assume that $l_Y$ and $l_R$ are known.

Nearly all commercial digital cameras sold today assume a variation of the von Kries coefficient law. Its overwhelming popularity comes not only from the simplicity of the implied mathematics, but also from the disconnectedness between color science and engineering. Main disadvantage, however, is that the coefficient law model is inaccurate and inadequate to explain chromatic adaptation mechanism in the HVS [28] [30]. Therefore, we expect that there are limitation to matching the output image with what the photographer saw, regardless of how accurate the illuminant estimation is.

Nevertheless, let us solve (6.2) assuming the von Kries coefficient law $\Psi_k(\cdot, \cdot)$. Suppose

$$\Psi(l_F, l_S) = \Psi_k(l_F, l_S) = \text{diag}(d_1, d_2, d_3)\Phi(l_F),$$

where $d_j$ is inversely proportional to $\Phi_j(l_S)$. Substituting this into (6.2),

$$\Psi_k(l, l_R) = \Psi_k(w_1p_1 + w_2p_2 + w_3p_3, l_Y)$$

$$= [\Psi_k(p_1, l_Y), \Psi_k(p_2, l_Y), \Psi_k(p_3, l_Y)]\vec{w},$$

and the solution $\vec{w}$ to (6.2) is given by

$$\vec{w} = [\Psi_k(p_1, l_Y), \Psi_k(p_2, l_Y), \Psi_k(p_3, l_Y)]^{-1}\Psi_k(l, l_R)$$

$$= [\Phi(p_1), \Phi(p_2), \Phi(p_3)]^{-1}\text{diag}(\frac{e_1}{d_1}, \frac{e_2}{d_2}, \frac{e_3}{d_3})\Phi(l),$$

where $d_j$ and $e_j$ are inversely proportional to $\Phi_j(l_Y)$ and $\Phi_j(l_R)$, respectively. Furthermore, $\Phi(p_1), \Phi(p_2),$ and $\Phi(p_3)$ can be pre-computed, and $\Phi(l)$ can be obtained.
directly from the image sensor output using $\Phi(l) = M_{\theta, \phi} \Theta(l)$:

$$\vec{w} = \left[\Phi(p_1), \Phi(p_2), \Phi(p_3)\right]^{-1} \text{diag} \left( \frac{e_1}{d_1}, \frac{e_2}{d_2}, \frac{e_3}{d_3} \right) M_{\theta, \phi} \Theta(l). \quad (6.3)$$

### 6.1.3 Jameson-Hurvich Model

We continue to assume that $l_R$ and $l_Y$ are made available.

Jameson and Hurvich hypothesize that induced opponent response process (2.5) is responsible for discounting the illuminant: this is in contrast to von Kries coefficient law [68]. In this section, we propose an alternative to the existing white-balance algorithms by solving (6.2) assuming a chromatic adaptation model $\Psi_F(\cdot, \cdot)$ (see (2.5) and (2.6)) instead of the von Kries coefficient law [24] [27] [28] [29] [30]. In doing so with a more accurate model for chromatic adaptation, the output from the digital camera will better match what the photographer saw in figure 6.1.

We begin by combining (2.5) and (2.6). For simplicity, define $\Gamma(l) = (M \Phi(l))^n$. Then

$$\Psi_F(l_F, l_S) = M^{-1}(c \Gamma(l_F) - \vec{i}_F)$$

$$= M^{-1}(c \Gamma(l_F) - kM \Psi_S(l_S, l_F))$$

$$= M^{-1}(c \Gamma(l_F) - k(c \Gamma(l_S) - \vec{i}_S))$$

$$= cM^{-1}(\Gamma(l_F) - k\Gamma(l_S)) + k^2 \Psi_F(l_F, l_S)$$

$$= M^{-1}(\Gamma(l_F) - k\Gamma(l_S)),$$

where, without loss of generality, $c = 1 - k^2$. Experiments indicate that $n = 1$ when $l_F$ and $l_S$ are isoluminant [30]. The method for choosing an appropriate value for $n$ in the general case, however, is not very well understood. We, therefore,
approximate the formula by operating as if stimuli are isoluminant (with \( n = 1 \)):

\[
\Psi_F(l_F, l_S) = M^{-1} \left( M\Phi(l_F) - k \left( \frac{m_1\Phi(l_F)}{m_1\Phi(l_S)} \right) BM\Phi(l_S) \right)
\] (6.4)

where \( B = \text{diag}(0, 1, 1) \) and \( m_1 \in \mathbb{R}^{1 \times 3} \) is the first row of \( M \) (hence \( m_1\Phi(l) \) is the achromatic channel value of \( l \) in opponent color space). Above, \( \frac{m_1\Phi(l)}{m_1\Phi(l_S)} \) normalizes the induction response \( M\Phi(l_S) \) using the ratio between the luminance values of the focal and surrounding stimuli, and \( \frac{m_1\Phi(l)}{m_1\Phi(l_S)} = 1 \) when \( l_F \) and \( l_S \) are isoluminant.

Matrix \( B = \text{diag}(0, 1, 1) \) is used in (6.4) because the techniques for estimating the illuminants \( l_Y \) and \( l_R \) are inherently limited to evaluating the chromatic content of the illuminant only (some details are discussed in section 2.2.2).

Because \( m_1\Phi(l_F) \) is scalar, (6.4) simplifies significantly:

\[
\Psi_F(l_F, l_S) = \Phi(l_F) - k \left( \frac{m_1\Phi(l_F)}{m_1\Phi(l_S)} \right) M^{-1}BM\Phi(l_S)
\]  
\[
= \Phi(l_F) - \frac{k}{m_1\Phi(l_S)} M^{-1}BM\Phi(l_S)(m_1\Phi(l_F))
\]  
\[
= \left( I - \frac{k}{m_1\Phi(l_S)} M^{-1}BM \right) \Phi(l_F)
\]  
\[
= L(\Phi(l_S), k)\Phi(l_F),
\]

where \( I \in \mathbb{R}^{3 \times 3} \) is an identity matrix, and \( L(\cdot, \cdot) \in \mathbb{R}^{3 \times 3} \) is

\[
L(\vec{v}, k) = I - \frac{k}{m_1\vec{v}} M^{-1}BM\vec{v}m_1.
\]

Now we are ready to solve the white-balance equation. Substituting \( \Psi(\cdot, \cdot) = L(\Phi(l_S), k)\Phi(l_F) \) to (6.2),

\[
L(\Phi(l_k), k_1)\Phi(l) = L(\Phi(l_Y), k_2)[\Phi(p_1), \Phi(p_2), \Phi(p_3)]\vec{w},
\]

and the \( \vec{w} \) that solves (6.2) is

\[
\vec{w} = [\Phi(p_1), \Phi(p_2), \Phi(p_3)]^{-1}L(\Phi(l_Y), k_2)^{-1}L(\Phi(l_k), k_1)\Phi(l).
\]
As was the case with the von Kries coefficient model, \( \Phi(p_1), \Phi(p_2), \) and \( \Phi(p_3) \) can be pre-computed, and \( \Phi(l) \) can be obtained directly from the image sensor output using \( \Phi(l) = M_{\theta,\phi}\Theta(l) \):

\[
\vec{w} = [\Phi(p_1), \Phi(p_2), \Phi(p_3)]^{-1}L(\Phi(l_Y), k_2)^{-1}L(\Phi(l_R), k_1)M_{\theta,\phi}\Theta(l). \tag{6.5}
\]

The equation above is significant from the computational point-of-view, also. The digital camera output \( \vec{w} \) can be computed from image sensor data \( \Theta(l) \) with a single matrix multiplication. This implies that the white-balance step and the color space conversion, which is also a matrix multiplication, can be combined into a single matrix multiplication procedure, making the computational cost of the white-balance algorithm virtually zero.

Finally, because the sizes of the focal and surrounding fields are unavailable, the parameter values \( k_1 \) and \( k_2 \) are adaptively chosen. Let \( \Omega_{\text{max}} \) be a set of \( K \) brightest pixels in \( \Omega \). Following the examples of MacAdam’s reflectance efficiency theory [42], we are interested in choosing \( k_1 \) and \( k_2 \) such that pixels in \( \Omega_{\text{max}} \) are neutral (i.e. close to \( l_Y \)). Mathematically, we solve the following optimization problem:

\[
\min_{k_1, k_2} \sum_{i \in \Omega_{\text{max}}} \|N[\Phi(p_1), \Phi(p_2), \Phi(p_3)]\vec{w}_i\|^2, \tag{6.6}
\]

where \( \vec{w}_i \) is the camera output at pixel location \( i \in \Omega \) according to (6.5). If we set \( N = BML(\Phi(l_R), 1) \), then the multiplication by \( N \) measures the chromaticity difference between \( [\Phi(p_1), \Phi(p_2), \Phi(p_3)]\vec{w}_i \) and \( l_Y \), normalized by \( m_1\Phi(l_Y) \), as before. The solution to (6.6) has a closed form:

\[
\begin{bmatrix}
   k_1 \\
   k_2
\end{bmatrix} = \begin{bmatrix}
   \vec{w}^T M_1^T N^T N M_1 \vec{w} & -\vec{w}^T M_1^T N^T N M_2 \vec{w} \\
   -\vec{w}^T M_2^T N^T N M_2 \vec{w} & \vec{w}^T M_2^T N^T N M_2 \vec{w}
\end{bmatrix}^{-1} \begin{bmatrix}
   \vec{w}^T N^T N M_1 \vec{w} \\
   -\vec{w}^T N^T N M_2 \vec{w}
\end{bmatrix},
\]

where \( M_1 = L(\Phi(l_R), 1) - I \), \( M_2 = L(\Phi(l_Y), 1) - I \), and \( \vec{w} = \sum_{i \in \Omega_{\text{max}}} \vec{w}_i \).
6.2 Experimental Results

The images used in our experiments are taken from Texas Instruments camera evaluation board DM270 DDS and Agilent Technologies camera evaluation board HDCP-2000. The TI DM270 DDS is equipped with a Sony 3 mega-pixel CCD sensor and Ricoh lens module. Agilent HDCP-2000 is equipped with an Agilent 300K pixel CMOS sensor. Unprocessed raw sensor data is acquired from these cameras, and the experimental results shown below are processed using Matlab codes simulating the image pipeline in a digital camera (see figure 2.2). The color conversion matrix is calibrated using Macbeth color charts shown in figure 6.2, taken on a typical cloudy day in Ithaca, New York (a reader of this thesis should also calibrate the monitor settings using figure 6.2 in order to display the results properly). We assume that CRT monitor with $\gamma = 2.2$ is viewed with an eye adapted to the monitor white.

Figure 6.3 shows indoor images processed without a white-balance algorithm applied. It is easy to see that the images appear unnaturally red or green. Note also that all images contain a Macbeth color chart, and that the white panels
Figure 6.3: Images taken with no white-balance algorithms. In each respective scene, the solid colors represent illuminants detected by (left) gray-world [35] and (right) method in [13].
Figure 6.4: Experiments using TI CCD camera.
Figure 6.5: Experiments using TI CCD camera.
Figure 6.6: Experiments using TI CCD camera.
Figure 6.7: Experiments using TI CCD camera.
Figure 6.8: Experiments using Agilent CMOS camera.
Figure 6.9: Experiments using Agilent CMOS camera.
from it are far from appearing white. The solid colors adjacent to each respective scene are the illuminant colors estimated using (top) the gray-world method and (bottom) the method in [13]. Images in the first two rows were processed from the raw sensor data in the TI CCD camera, while the images in the last row were processed from the raw sensor data in the Agilent CMOS camera.

Figures 6.4 through 6.9 show the same sensor data processed with a variety of white-balance methods. In each respective scene, the images in the first row assumed the illuminant estimated from the gray-world method, while the images in the second row assumed the illuminant color estimated from the method in [13]. The images in the left columns were generated using (6.3), while the images in the right columns use the proposed formula, (6.5).

The colors observed in the white panels in the Macbeth chart give a rough indication of how well the algorithms work—we would like an object, whose reflectance spectral distribution is relatively constant, to appear neutral in the output image. The solid-color squares in the center of each figure show the colors taken from the white panels. Compared to the images generated by (6.3), the images generated by the proposed white-balance formula showed more neutral colors in the white panels, regardless of the method of illuminant estimation assumed. The images generated using (6.3) often suffer from a hazy appearance (figures 6.4 and 6.6 through 6.8), while the colors processed using (6.5) are slightly desaturated (see red shirt in figure 6.7). Overall, the images generated by (6.5) appear more natural than those using (6.3) (though not perfect).

A quantitative evaluation of the white-balance algorithms is difficult. Figure 6.10 shows the $\Delta_{ab}$ distance between the white panel color and a neutral white. We strongly caution that this is only a rough indication of performance because
Figure 6.10: $\Delta_{ab}$ distance between the white panel and a neutral white. The numbers in abscissa correspond to the figure numbers.

The goal of the white-balance problem is to match the appearance of the digital camera output to the scene observed by the photographer. Since the HVS color constancy is only approximate, a measure of color constancy such as the one in figure 6.10 is not a good testimony of the effectiveness of the methods. However, because it is impossible to quantitatively measure the difference between what the photographer saw and what is seen in the display media, we include this graph here as a reference.

### 6.3 Summary

In this section, we formulated the white-balance problem using Jameson and Hurvich’s induced opponent response chromatic adaptation theory. This is in a sharp contrast to the von Kries approach to the white-balance problem. Approximation with a scaling constant was introduced to operate as if the focal and surrounding
fields were isoluminant. The solution to the white-balance problem reduces to a single matrix multiplication. The experimental results, using the basic and the state-of-the-art illuminant estimation methods, verify that the induced opponent response approach to solving the white-balance problem yields more natural looking images than traditional methods. The algorithm is computationally efficient, because it can be combined with the color conversion step in the image pipeline.
CHAPTER 7

CONCLUSION

The following is a summary of results presented in this thesis.

• The CMOS image sensor noise model in [60] was verified using image processing techniques. Noise can be characterized as $Y = X + (k_0 + k_1X)\delta$, where $Y$ is the noisy sensor value, $X$ is the ideal pixel value, $\delta \sim \mathcal{N}(0, 1)$ is noise, and $k_0, k_1 \in \mathbb{R}$ are parameters.

• Analysis of noise characteristics with CIE-Lab color space conversion revealed that the perceived noise in the low-signal region is larger than that of the high-signal region.

• An ideal image patch can be modeled effectively as a linear combination of noisy image patches. The relationship is made exact by allowing perturbations in the ideal and noisy image patches.

• We developed a TLS denoising algorithm by minimizing the perturbation in image patch model in the TLS sense. The output images from the proposed algorithm were compared to the images from the state-of-the-art image denoising methods. In most cases, the proposed algorithm smoothes the noise in the flat regions and preserves the image details better the existing methods given signal-independent and signal-dependent noise. However, the computational cost of the TLS denoising algorithm is high.

• When the denoising filter is constrained such that low-frequency components from red and blue colors do not contribute to the estimation of green, then the same filter can be used to estimate full-color pixel values given a sparsely
sampled noisy image sensor data. Many existing denoising methods can be combined with the demosaicing procedure using this technique.

- We developed a method to estimate pixel values given noisy sensor data using the TLS image denoising technique. The proposed demosaicing/denoising algorithm suppresses CMOS image sensor noise while effectively interpolating the missing pixel components better than when treating demosaicing and denoising problems independently using the state-of-the-art algorithms. The combined algorithm also reduces the computational complexity compared to performing the procedures independently.

- The white-balance problem was formulated precisely using a viewing model, and it was solved using the induced opponent response theory. The output colors from the proposed solution, combined with existing illuminant estimation techniques, is more natural than the output colors from the conventional approach, which combines von Kries coefficient law with the illuminant estimation techniques, and the color in the white panel of the Macbeth color chart appears more neutral. The computational cost of this new method is virtually zero.
APPENDIX A

THE SOLUTION TO TLS PROBLEM

Let $X, E \in \mathbb{R}^{m \times n}, Y, R \in \mathbb{R}^{m \times p}, \alpha \in \mathbb{R}^{n \times p}, A \in \mathbb{R}^{m \times m}, B \in \mathbb{R}^{n+p \times n+p}, A, B$ nonsingular. Then the $\alpha$ value that satisfies

$$
\min_{Y+R=(X+E)\alpha} \|A[E,R]B\|_F^2
$$

(A.1)

is called the total least square solution (denoted $\alpha_{\text{TLS}}$).

The solution to the TLS problem (A.1) is well documented [15] [8]. Let $m \geq n+p$, and $U\Sigma V^T = A[X,Y]B$ be the singular value decomposition of $A[X,Y]B$, where $\Sigma = \text{diag}(\sigma_1, \ldots, \sigma_{n+p})$, $\sigma_i > \sigma_{i+1}^2$. Let $\Sigma_1 = \text{diag}(\sigma_1, \ldots, \sigma_n)$, $\Sigma_2 = \text{diag}(\sigma_{n+1}, \ldots, \sigma_{n+p})$, and partition $U, V,$ and $B$ as follows:

$$
U = \begin{bmatrix} U_1 & U_2 \end{bmatrix}
$$

(A.2)

$$
V = \begin{bmatrix} V_{1,1} & V_{1,2} \\ V_{2,1} & V_{2,2} \end{bmatrix}
$$

(A.3)

$$
B = \begin{bmatrix} B_{1,1} & B_{1,2} \\ B_{2,1} & B_{2,2} \end{bmatrix}
$$

(A.3)

Then the solution to (A.1) is given by

$$
\alpha_{\text{TLS}} = -(B_{11}V_{12}^{-1} + B_{12})(B_{21}V_{12}^{-1} + B_{22})^{-1}
$$

(A.4)

$$
A[E,R]B = -U_2\Sigma_2[V_{12}^T, V_{22}^T].
$$

(A.5)
The sketch proof for (A.4) and (A.5) goes as follows. Since \((X+E)\alpha = Y + R\) (see (A.1)),

\[
\{A[X,Y]B + A[E,R]B\} B^{-1} \begin{bmatrix} \alpha \\ -I_p \end{bmatrix} = 0, \tag{A.6}
\]

where \(I_p \in \mathbb{R}^{p \times p}\) is an identity matrix. Furthermore,

\[
\]

Equation (A.6) implies that (A.7) is at most rank \(n\). The main idea is to find \(A[E,R]B\) with minimum Frobenius norm such that the matrix in the brackets become rank \(n\). Because the Frobenius norm is invariant under unitary transformation, it is clear from (A.7) that

\[
U^T A[E,R]BV = \begin{bmatrix} 0 & 0 \\ 0 & \Sigma_2 \end{bmatrix}. \tag{A.8}
\]

This proves (A.5). The right singular vector matrix corresponding to \(\Sigma_2\) is \([V_{12}^T, V_{22}^T]^T\) so from (A.6) and (A.8)

\[
B^{-1} \begin{bmatrix} \alpha \\ -I_p \end{bmatrix} = \begin{bmatrix} V_{12} \\ V_{22} \end{bmatrix} M_p
\]

for some matrix \(M_p \in \mathbb{R}^{p \times p}\). This proves (A.4).
APPENDIX B

REVIEW OF GENERALIZED HOMOMORPHIC FILTERING

In this section, we review the concept of generalized homomorphing filtering technique [9], developed by Ding et al. (i.e. not the author). The purpose of the method is to design an invertible tone-scale function (point-wise operator) that (approximately) decouples the noise from the signal.

Assume a general signal-modulated noise model,

\[ x = s + f(s)\delta, \]

where \( s \in \mathbb{R} \) is the signal, \( \delta \in \mathbb{R} \) is noise characterized as a signal-independent zero-mean random process with unit variance, \( x \in \mathbb{R} \) is the observed noisy signal, and \( f : \mathbb{R} \rightarrow \mathbb{R}, f(\cdot) > 0 \). We would like to define a strictly monotonic function \( g : \mathbb{R} \rightarrow \mathbb{R} \) such that \( g(x) \approx g(s) + \delta \) (i.e. noise is signal-independent).

Using Taylor expansion,

\[ g(x) = g(s) + g^{(1)}(s)f(s)\delta + \frac{1}{2!}g^{(2)}(s)f^2(s)\delta^2 + \ldots, \]

where \( g^{(i)}(\cdot) \) is the \( i \)th derivative of \( g(\cdot) \). Assuming that the higher derivatives are sufficiently small,

\[ g(x) \approx g(s) + g^{(1)}(s)f(s)\delta, \]

and set \( g^{(1)}(s)f(s) = 1 \). Then

\[ g(x) \approx g(s) + \delta. \]

To solve for \( g(\cdot) \),

\[ g(s) = \int_0^s g^{(1)}(s_0)ds_0 = \int_0^s \frac{1}{f(s_0)}ds_0. \]
Note that if $f(s) > 0$, $g(x)$ is invertible. So long as $f(s)$ is smooth, the generalized homomorphic filter $g(\cdot)$ transforms the signal-modulated noise to an additive signal-independent noise with unit variance,

$$g(x) = g(s + f(s)\delta) \approx g(s) + \delta.$$
BIBLIOGRAPHY


[18] E. Hering, Pfluger’s Arch. ges. Physiol. 10, 519, 1895.


[26] International Commission on Illumination, “Recommendations on uniform color spaces, color difference equations, psychometric color terms,” Supplement No.2 to CIE publication No.15 (E.-1.3.1), TC-1.3., 1971.


