

POISSON NOISE PARAMETER ESTIMATION AND COLOR IMAGE DENOISING  
FOR REAL CAMERA HARDWARE

Dissertation

Submitted to

The School of Engineering of the

UNIVERSITY OF DAYTON

In Partial Fulfillment of the Requirements for

The Degree of

Doctor of Philosophy in Engineering

By

Chen Zhang

Dayton, Ohio

December, 2019



**University of  
Dayton**

POISSON NOISE PARAMETER ESTIMATION AND COLOR IMAGE DENOISING  
FOR REAL CAMERA HARDWARE

Name: Zhang, Chen

APPROVED BY:

---

Keigo Hirakawa, Ph.D.  
Advisory Committee Chairman  
Associate Professor, Electrical and  
Computer Engineering

---

Russell Hardie, Ph.D.  
Committee Member  
Professor, Electrical and Computer  
Engineering

---

Raúl Ordoñez, Ph.D.  
Committee Member  
Professor, Electrical and Computer  
Engineering

---

Ryan Kappedal, Ph.D.  
Committee Member  
Professor, Effective Communicator

---

Robert J. Wilkens, Ph.D., P.E.  
Associate Dean for Research and Innovation  
Professor  
School of Engineering

---

Eddy M. Rojas, Ph.D., M.A., P.E.  
Dean, School of Engineering

© Copyright by

Chen Zhang

All rights reserved

2019

# ABSTRACT

## POISSON NOISE PARAMETER ESTIMATION AND COLOR IMAGE DENOISING FOR REAL CAMERA HARDWARE

Name: Zhang, Chen  
University of Dayton

Advisor: Dr. Keigo Hirakawa

Noise is present in all images captured by real-world image sensors. The distribution of real camera sensor data is well approximated by Poisson, and the estimation of the light intensity signal from the Poisson count data plays a prominent role in digital imaging. Multi-scale Poisson image denoising techniques have processed Haar frame and wavelet coefficients—being enabled by Skellam distribution analysis. Previous work has solved the minimum risk shrinkage operator (MRSO) that produces denoised wavelet coefficients with best achievable Mean Squared Error (MSE) for gray scale image. We extend the idea of MRSO to denoise color sensor data in color-opponent space, improving the quality of denoised color images. In addition, the stable representation of color is to use ratios which we denote by chromaticities. Thus we propose a new Bayes estimator for color image denoising in log-chromaticity coordinate. Using full resolution real R/G/B camera images, we verified that the proposed denoising is more stable than the state-of-art color denoising techniques, yielding higher image quality result. Furthermore, the noise parameters that characterize the level of noise in an image or video frame are required for effective denoising. We develop a novel technique to estimate the noise parameters from natural scenes by exploiting the global joint statistics across multiple video frames, which can be interpreted as a binomial random variable that is insensitive to textures and scene contents. We verify experimentally that

the proposed noise parameter estimation method recovers noise parameters more accurately than the state-of-art noise parameter estimation techniques.

*To my dear friends at UD*

Thy friendship makes us fresh

## ACKNOWLEDGMENTS

The denoising and noise parameter estimation works are funded by Japan Broadcasting Corporation (NHK).

## TABLE OF CONTENTS

ABSTRACT . . . . .	iii
DEDICATION . . . . .	v
ACKNOWLEDGMENTS . . . . .	vi
LIST OF FIGURES . . . . .	ix
LIST OF TABLES . . . . .	xiii
I. INTRODUCTION . . . . .	1
1.1 A: Introduction of Poisson Parameter Estimation . . . . .	1
1.2 B: Introduction of Color Image Denoising . . . . .	2
II. BACKGROUND AND RELATED WORK . . . . .	5
2.1 A: Background and Related Works of Parameter Estimation . . . . .	5
2.1.1 Noise Distribution and Parameterization . . . . .	5
2.1.2 Noise Parameter Estimation Techniques . . . . .	8
2.2 B: Background and Related Works of Color Image Denoising . . . . .	10
2.2.1 Review: Noise Model and Denoising Technique for Gray-scale Image	10
2.2.2 Image Denoising of Each Color Channel. . . . .	13
2.2.3 Image Denoising in Color-opponent Space. . . . .	14
III. A:PROPOSED THEOREMS OF PARAMETER ESTIMATION . . . . .	17
3.1 Theoretical Results: Binomial Statistics . . . . .	17
3.2 Application: Noise Parameter Estimation . . . . .	21
3.2.1 Motivation . . . . .	22
3.2.2 Lower-Bound Function . . . . .	22
3.2.3 Improvements: Pixel Registration and Selection . . . . .	25
IV. B:PROPOSED THEOREMS OF COLOR IMAGE DENOISING . . . . .	30
4.1 Background: Aitchison Geometry . . . . .	30
4.1.1 Elementary Operations . . . . .	30
4.1.2 Centered Log-Ratio Transform . . . . .	31
4.1.3 Log-chromaticity Coordinate Representation of Color . . . . .	34
4.1.4 Time-Frequency Analysis Of CLRT Simplex coefficients . . . . .	35
4.2 Multivariate Inhomogeneous Poisson Sequence and Bayes Estimate in Chromaticity Coordinates . . . . .	36
4.2.1 Poisson-Multinomial Likelihood Function and Minimum Mean Square Error Estimator in Chromaticity Coordinates . . . . .	36
4.2.2 Linearized Poisson-Multinomial Likelihood Function in Wavelet Domain . . . . .	39
4.2.3 Practical MMSE Denoising in Chromaticity Coordinates . . . . .	42

V.	EXPERIMENTAL RESULTS AND DISCUSSION . . . . .	47
5.1	A:Experimental Results and Discussion of Parameter Estimation . . . . .	47
5.1.1	Spatial Correlation of Noise . . . . .	47
5.1.2	Dataset and Experiment Setup . . . . .	48
5.1.3	Noise Parameter Estimation Performance . . . . .	52
5.1.4	Discussion . . . . .	54
5.2	B:Experimental Results and Discussion of Color Image Denoising . . . . .	57
5.2.1	Parameter Tuning Methodology for Proposed Denoising . . . . .	57
5.2.2	Verification Using Synthetic Images . . . . .	60
5.2.3	Real Camera Sensor Data Experiment . . . . .	62
VI.	CONCLUSIONS . . . . .	68
6.1	A: Conclusions of Parameter Estimation . . . . .	68
6.2	B: Conclusions of Color Image Denoising . . . . .	68
	BIBLIOGRAPHY . . . . .	69
APPENDICES		
A.	Appendix for Color Image Denoising . . . . .	73
A.1	Proof of (4.58) . . . . .	73
A.2	Proof of (4.59) . . . . .	74

## LIST OF FIGURES

2.1	Evidence of heteroskedastic noise and noise parameter drift. (a) Observed video sensor frame $\tilde{X}$ with Colorchecker. (b) The empirical mean/variance of the Colorchecker patches on Day 1. (c) The empirical mean/variance on Day 2, taken with the same sensor parameters as the plot in (b). The difference between (b) and (c) indicate the noise parameter drift over time. (d) The empirical mean/variance of the recovered Poisson image using (2.7). The mean and variance are equal. 3.1. The cyan line denotes the estimated relation computed by the proposed theorem. . . . .	6
2.2	Color denoising comparison of real camera sensor data. The plots represent the pixel values of a line crossing black/white/grey patches. Scene was captured using Sony $\alpha 7RIII$ with pixel-shifting mode to capture the full resolution R/G/B data. Plots in (a,b,c,d) correspond to the long-exposure image, which we use as a proxy of cleaner reference. Plots in (e,f,g) illustrate image denoising process in each color channel. Plots in (h,i,j,k) illustrate image denoising process in color-opponent space. Plots (e,l,m,n) illustrate the proposed image denoising process in log-chromaticity space. More specifically, for the plots in (a,e,f,j,m), the red/green/blue lines represent the R/G/B values of sensor data, respectively. For the plots in (b,g,k,n), the red/green/blue lines represent the R/G/B values after white balance. For the plots in (c,h,i), the purple lines denote the chrominance values R-G of sensor data. Similarly, the gold lines denote the G-B values. And the emerald-green lines denote the R-B values. For the plots in (d,l), the magenta lines represent the log-chromaticity value corresponding to $\log \frac{R}{R+G+B}$ . The darker-green lines correspond to $\log \frac{G}{R+G+B}$ . And the cyan lines correspond to $\log \frac{B}{R+G+B}$ . From (b), we know the latent pixel intensities are in black/white/grey. Any pixels with obvious imbalance in R/G/B after white balance will create artificial color artifacts. Clearly, the performance of (n) > (k) > (g). The (i,j,k) and (l,m,n) share the same denoised luminance component, which does not show in (c,d,h,i,l). Because luminance represents the brightness only, which is separated from hue components. In this figure, we mainly focus on the comparison of color processing. . . . .	11
3.1	Block diagram of the proposed Poisson noise parameter estimation method. The binomial statistics in (3.27) are computed in the orange block. The main steps for estimating the noise parameters (lower-bound linear regression in (3.31) and parameter estimation in (3.26)) appear in green blocks. The blue blocks carry out the pixel registration and selection steps outlined in Section 3.2.3 to ensure that the proposed method is robust to foreground and background motion. . .	17

3.2	<p>Demonstration of the proposed noise parameter estimation. The blue points denote the empirical binomial statistics <math>\hat{\sigma}_{\tilde{X} \tilde{Z}_n=\tilde{z}}^2</math> in (3.27). The linear regression in (3.28) and lower envelope fit in (3.29) are shown by green and red lines, respectively. The cyan line denotes the proposed robust lower-bound function <math>f(\tilde{z}) = s\tilde{z} + t</math> computed using (3.31), based on which we recover the noise parameters using (3.26). For reference, we included the magenta line corresponding to the noise parameters from calibration. Qualitatively, we may assess the accuracy of the parameter estimation by observing the closeness of green/red/cyan lines to the magenta line. . . . .</p>	21
3.3	<p>Illustration of how proposed method gains robustness to foreground and background motions. (a,b) Two adjacent frames of a video sequence (which we refer to as “frame 1” and “frame 2”, respectively). (c,d) Difference of frames 1 and 2 (<math>\tilde{D}_n = \tilde{X}_n - \tilde{Y}_n</math>) before and after frame registration, respectively. Frame registration largely eliminates the background motion caused by camera motion, but the foreground motion remains. (e) Pearson product-moment correlation coefficient of <math>sgn(\tilde{D})</math> computed on (d). It determines the regions of image affected by foreground motion without relying on the magnitude of <math>\tilde{D}</math>. (f) Empirical binomial statistics with no motion compensation (as shown in (c)). (g) Empirical binomial statistics after frame registration (as shown in (d)). (h) Empirical binomial statistics after frame registration and thresholding (as shown in (e)). The improvement of addressing foreground and background motion is evidenced by comparing the tightening of clusters in (f), (g), and (h). See text for explanation. . . . .</p>	23
4.1	<p>Demonstration of chromaticity coordinate. Equation (4.23) maps color channels <math>\mathcal{R} \mathcal{G} \mathcal{B}</math> onto the triangle plane in (a)—the chromaticity coordinate. By coloring the chromaticity coordinate triangle, (b) demonstrates the relation between color and color ratios . . . . .</p>	34
4.2	<p>Analysis of prior density <math>Pr[\mathbb{B}]</math> using long exposure image as the proxy of clean reference. (a,b), Log-histogram (blue plots) of the marginal prior density <math>Pr[\mathbf{B}_1]</math> and <math>Pr[\mathbf{B}_2]</math>, respectively. The red-line plots illustrate the approximation of the marginal prior with mixture-Gaussian model. The green-line plots illustrate the approximation with Laplacian model. (c), 2D-histogram of <math>Pr[\mathbb{B}]</math>. (d), Approximation of <math>Pr[\mathbb{B}]</math> using 2-D mixture-Gaussian model as described in (4.55). (e), the top-down view of (c). We can see <math>\mathbf{B}_1</math> and <math>\mathbf{B}_2</math> are less correlated. (f), the approximation of (e) using 2-D mixture-Gaussian model. In this model, <math>\mathbf{B}_1</math> and <math>\mathbf{B}_2</math> are dependent but uncorrelated. . . . .</p>	43
5.1	<p>Frame examples of video sequence. Each scene has two recordings—first was shot on tripod, second one shot with camera held by hand. . . . .</p>	48

5.2	Performance evaluation of the noise parameter estimation algorithms. Each plot represents one experiment with 69 consecutive frame pairs of scenes in Figure 5.1, respectively. The plots in (a) correspond to the test video recorded on tripod, while (b) correspond to the tests with a camera that is held by hand. The results of the proposed robust lower-bound regression are shown by cyan and blue lines, while the “ground truth” line (according to the calibration) is shown by the magenta line. For comparison, we map the noise parameter estimates of methods in [1–3] to the binomial statistics plot as well. Each thin yellow/red/green line corresponds to a parameter estimate from one of the 70 frames in a video sequence. The thick line corresponds to their average. Qualitatively, we may assess the accuracy of the parameter estimation by observing the closeness of these lines to the magenta line. . . . .	49
5.3	(a) An example frame from a noisy texture patch video sequence. (b) The empirical binomial statistics $\hat{\sigma}_{\tilde{X} \tilde{Z}_n=\tilde{z}}^2$ in (3.27). The plot follows the legends used in Figure 5.2. The recovered S-MSE/N-MSE scores from the proposed method are 0.0062/0.0020. Compare this to the S-MSE/N-MSE scores of 0.4429/0.1733 by method in [1], 0.0911/0.0383 by method in [3], and 1.7271/0.7209 by method in [2]. . . . .	55
5.4	Successful and non-successful examples of binomial statistics computed from a single frame with $(\kappa, \gamma_1, \gamma_2, \eta) = (10, 0.5, 0.002, 15)$ . In the scene in (a), the proposed robust lower-bound function regression successfully estimates the noise parameter. In (b), the noise parameter estimation is more difficult. The topic of single frame noise parameter estimation is left for future work. See text. . .	56
5.5	Behavior analysis for tunable parameter $\eta_{y_{\tilde{B}_1}}$ and $\eta_{y_{\tilde{B}_2}}$ . (a) Relatively clean image taken by increasing the exposure time. (b) Noisy image taken under low illumination. (c) Denoised result with very low value of $y$ under three levels of Haar wavelet decomposition. (d) Denoised result with very high value of $y$ under three levels decomposition. (e) Denoised result with very low value of $y$ under four levels decomposition. (f) Denoised result with very high value of $y$ under four levels decomposition. (g) Denoised result with fine tuned value of $y$ under four levels decomposition. . . . .	57
5.6	The Relatively clean image taken by increasing the illumination. Scenes were captured using Sony $\alpha$ 7RIII with pixel-shifting mode to capture the full resolution R/G/B data. Clean images are used as the proxies of latent color intensities. .	62

5.7	Color denoising results comparison. The reference image correspond to Figure5.6(a) and (e). Noisy images were taken under low illumination using same camera as Figure5.6, but three different camera settings. The acquired less/moderate/very noisy images are displaced in a same picture to demonstrate the degradation change according to noise level. (a,b) Noisy images. (c,d) Denoising each color channel independently using MRSO [4]. (e,f) Denoising in color-opponent space using (2.22)–(2.27). (g,h) Proposed color denoising in log-chromaticity coordinate. (i,j) Denoising using CBM3D [5]. (k,l) Denoising in color-opponent using dnCnn [6]. . . . .	63
5.8	Some other examples of color denoising results comparison. The reference image correspond to Figure5.6(c) and (f). The figure arrangement follows Figure 5.7.	64
5.9	Some other examples of color denoising results comparison. The reference image correspond to Figure 5.6(b) and (d). The figure arrangement follows Figure 5.7.	65

## LIST OF TABLES

3.1	Spatial correlation of noise in various commercial cameras and smartphones are <b>uncorrelated</b> . We collected 100 raw images (Bayer color filter array patterned sensor) of maximally blurred blank walls taken for each camera setup placed on a tripod. We computed Pearson product-moment correlation coefficient of spatially adjacent pixels (which have two different color filters, denoted “cross color”) and two pixels apart (with same color filters). Maximum correlation coefficient magnitude is only 0.0969, most are smaller than 0.05. . . . .	29
5.1	Quality of estimated Poisson noise parameters $(\hat{\alpha}, \hat{\beta})$ assessed by S-MSE and N-MSE scores. The proposed algorithm yields one estimate for the entire video sequence (so we only report one MSE score). For methods in [1–3] that yield noise parameters for each frame, we report the minimum/maximum/mean/median MSE scores of all frames, which are subsequently averaged over all video sequences. We partitioned our dataset into low, medium, and high gain settings; and assessed the performance for stationary and hand-held cameras separately.	50
5.2	Summation MSE score in log chromaticity coordinates averaged over 63 samples. Each score is computed as $\sum_m \gamma[\log \hat{\theta}_m(\mathbb{F})]$ . The maximum intensity of $\mathbb{F}$ indicates the noise level. $\max(\mathbb{F}) = 20$ corresponds to the moderate noisy scenario. $\max(\mathbb{F}) = 10$ corresponds to the high noisy scenario. $\max(\mathbb{F}) = 5$ corresponds to the extremely noisy scenario. Each noisy image was denoised with four different methods. The method with score in bolder face has the minimum $\ell^2$ error, indicating the best obtained performance in each noise level. . . . .	61

# CHAPTER I

## INTRODUCTION

### 1.1 A: Introduction of Poisson Parameter Estimation

The goal of this work is to characterize the random noise in a video sensor by estimating noise parameters. Noise parameters are needed in most of image and video denoising methods, as the level of noise determines the degree to which the denoising functions smooth out the spatial features [7–22]. Noise in video sensor measurements are dynamic, as they are affected by the gain factor and sensor/environment temperatures. See Figure 2.1(b,c) showing an example of noise parameters drifting for a fixed camera setting. The parameters governing the noise model can be computed by a simple calibration experiment, which requires that we capture images/video sequences of known calibration targets, like the one shown on Figure2.1(a). But calibration is not desirable or possible in many cases—example scenarios include image sensor becoming hot during a long video capture, and the sensor gain adjusting in real-time to compensate for the dynamic range of the scene. It is also inconvenient and impractical to require consumers to calibrate their video camera frequently.

Estimating the noise parameter without calibration targets is challenging due to the complexity of scene content. Existing noise parameter estimation techniques are “patch based” methods, determining regions of the image that are homogeneous (resembling the calibration targets). These methods fail when scene content is inhomogeneous (e.g. textured) or if the noise dominates, making it difficult to determine the homogeneous regions. With wide-angle lens or when observing scenes far away, homogeneous patches are small and unusable.

Video presents an opportunity to develop a noise parameter estimation technique that is invariant to scene content by exploiting scene redundancy across video frames. Specifically, joint statistics across multiple frames can be interpreted as a binomial random variable, and we develop a method to estimate noise parameters based on properties of binomial statistics. The proposed algorithm draws on global binomial statistics (unlike the patch based methods which are inherently spatially local) that are insensitive to textures and scene contents, and achieves performance better than the state-of-the-art noise parameter estimation techniques.

The remainder of this paper is organized as follows. We briefly review image/video sensor noise models, and existing techniques for parameterizing them in Section 2.1. In Section 3.1, we develop the theory of binomial statistics that help characterize the joint statistics between pixels in multiple frames. In Section 3.2 we develop a novel noise parameter estimation technique based on the binomial statistics we investigated in Section 3.1. In Section 5.1 we present a dataset we collected to assess the noise parameter estimation techniques, and verify quantitatively that the proposed method outperform the existing techniques. Finally, we make concluding remarks in Section 6.1.

## 1.2 B: Introduction of Color Image Denoising

The goal of this research is to develop color denoising technique for full resolution R/G/B raw sensor data of color video camera. Pixel sensor is a type of integrating detector, many evidence have proved that the distribution of sensor measurement is well modeled as Poisson [7, 23]. The normal approximation to Poisson breaks down especially during low-photon scenarios because the camera noise is heteroskedastic. Thus, the intense demand for shrinking sensor size and increasing image resolution encourages the development of

Poisson images denoising [11, 12, 15, 17, 19, 24–26]. The Poisson based denoising has been well-studied for gray-scale image. It can be extended for denoising color image by treating each color channel as independent gray-scale image. However, as the Figure 2.2 (g) shows, the undesired under/over-smoothing in different color channel is also heteroskedastic. In other word, this method is not capable to keep the spatial dependency between different color channels. Consequently, the denoising result creates artificial color defects that are spatially inconsistent.

Alternatively, with the notions of luminance component and chrominance component, denoising in color-opponent space is another way to overcome the spacial consistency. Color-opponent space is commonly used in many color image processing applications, including demosaicking, color video broadcasting, printer and projector design, etc. However, as shown in the Figure 2.2 (a), the disequilibrium of spectral sensitivities of red/green/blue channels exists in every modern camera. Normally, it is compensated by white balance which is illustrated by the change from Figure 2.2 (a) to (b). The denoising of chrominance component recovers the step size relationship between different color channels so that it maintains the spatial dependency across different channels. However, as shown in Figure 2.2 (f,g), this method fails after white balance because it can hardly maintain the color ratios, especially when the brightness is low and the spatial contrast is high. Therefore, denoising in color-opponent space is not ideal unless the luminance is constant or can be normalized.

Color scientific research shows that hue is invariant to brightness [27]. Therefore, it is better to model the color noise with the brightness being normalized, which is usually inconvenient and challenge. CIE xyY color space describes color in two parts: luminance “Y” and chromaticity “x and y”. Originally, the latter pair is defined in terms of color

ratios computed from the values in CIE XYZ space, and used to draw the CIE chromaticity diagram. Chromaticity is a stable representation of Hue. Therefore, in order to overcome the issues of spatial consistency and brightness invariance in color denoising of sensor data, we analyze the color behavior in chromaticity space. Then we develop a minimum mean square error (MMSE) estimator to estimate the clean chromaticity in log domain from noisy observations. Using real camera noisy images, we testify that the proposed method provides color denoising with stable performance than the denoising in each color channel method and denoising in color-opponent space. Using synthetic noisy images, we experimentally verify that the proposed method can achieve better PSNR performance than the state-of-the-art color denoising techniques.

We hereby emphasize that this work bypassed the influence of demosaicking to study the noise behavior of raw sensor data. The acquisition of R/G/B/ full resolution image will be explained in the future section. The remainder of this paper is organized as follows. We briefly review the gray scale Poisson image denoising, and propose a practical color denoising strategy in color-opponent space in Section 2.2. In Section 4.1 we develop the theory of Centered Log-Ratio Transform (CLRT) and established the basic idea of log-chromaticity coordinate analysis. In Section 4.2, we establish a framework to analyze multivariate inhomogeneous Poisson sequence using CLRT and propose the Bayes estimator in log-chromaticity coordinates. We verify the significance of proposed method in Section 5.2. Finally, we make concluding remarks in Section 6.2.

## CHAPTER II

### BACKGROUND AND RELATED WORK

#### 2.1 A: Background and Related Works of Parameter Estimation

##### 2.1.1 Noise Distribution and Parameterization

The camera noise is heteroskedastic. The variance of noise is dependent on the intensity of the pixel value we measure. Studies have confirmed that the noise variance increases linearly with the signal [1,3,7,21,22,28]. This phenomenon is attributed to photon emission (the randomness associated with photons in light) and photon recapture (the stochastic process of generating electrical current in photodiode) [7]. The thermal noise in dark current contribute to additional noise. It is largely signal-independent, and scales with the gain as well as the temperature of the sensor. Combined, signal strength and noise variance are typically related by an affine relationship, as shown by Figure 2.1. The photon emission, photon recapture, and thermal noise are *non-repeatable* variables in sensor measurement, and is the focus of our paper.

Other sources of noise include fixed pattern noise stemming from non-uniform characteristics of A/D converters and pixels (whose influence can be minimized by extensive calibration), quantization noise that affects very low signals (typically overcome by increasing sensor gain), and reset noise caused by the inadequate discharging of capacitors (handled by subtracting the bias). However, we exclude these from the considerations of this paper as they are largely repeatable sources of noise (i.e. not dynamic).

There are three noise models for noise in image and video sensors commonly used in image denoising methods [7–23]. Each of them is consistent with the affine relationship between the signal strength and noise variance, as evidenced in Figure 2.1. First noise

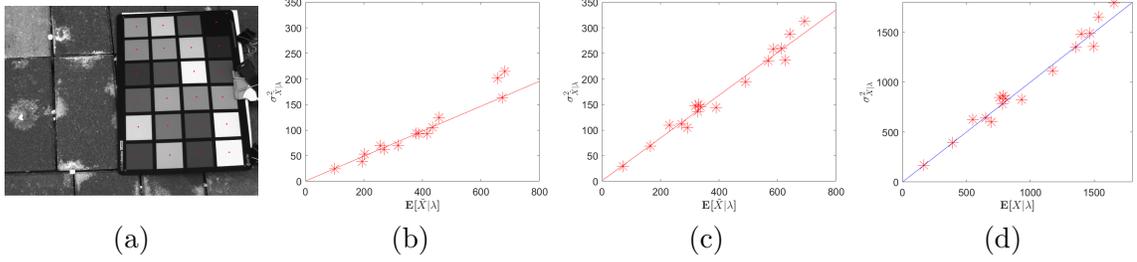


Figure 2.1: Evidence of heteroskedastic noise and noise parameter drift. (a) Observed video sensor frame  $\tilde{X}$  with Colorchecker. (b) The empirical mean/variance of the Colorchecker patches on Day 1. (c) The empirical mean/variance on Day 2, taken with the same sensor parameters as the plot in (b). The difference between (b) and (c) indicate the noise parameter drift over time. (d) The empirical mean/variance of the recovered Poisson image using (2.7). The mean and variance are equal. 3.1. The cyan line denotes the estimated relation computed by the proposed theorem.

model of the observed image/video sensor output  $\tilde{X}$  is based on Poisson:

$$\tilde{X} = \alpha X + \beta \tag{2.1}$$

$$X|\lambda, \xi \sim \mathcal{P}(\lambda + \xi)$$

where  $\lambda$  represents the underlying light intensity (the “signal”), and  $\xi$  is the offset that accounts for thermal noise that is independent of the signal. Alternatively, the Gaussian-Poisson noise model uses additive white Gaussian noise to represent the signal-independent noise component (often attributed to thermal noise). Consider the following:

$$\tilde{X} = a\Lambda + N_{\mathcal{P}} + N_{\mathcal{N}}, \tag{2.2}$$

where  $\Lambda + N_{\mathcal{P}} \sim \mathcal{P}(\Lambda)$  represent the signal dependent noise as defined by Poisson distribution, and  $N_{\mathcal{N}} \sim \mathcal{N}(0, b)$  represent additive white Gaussian noise. Lastly, a “normal” approximations to (2.2) coupling the noise variance to the pixel intensity yields:

$$\tilde{X}|\Lambda \sim \mathcal{N}(a\Lambda, \Lambda + b) \tag{2.3}$$

$$\tilde{X} = a\Lambda + \sqrt{\Lambda + b} \cdot N_{\mathcal{N}_0},$$

where  $N_{\mathcal{N}_0} \sim \mathcal{N}(0, 1)$ . As a side note, a generalized model for signal dependent noise of the form

$$\tilde{X}|\Lambda \sim \mathcal{N}(a\Lambda, \Lambda^\tau + b) \quad (2.4)$$

has been proposed also, where the additional parameter  $\tau$  controls the dependence on the pixel value [2, 3, 29–33]. However, Figure 2.1 clearly suggests  $\tau = 1$  for real image/video sensor data. The model with  $\tau \neq 1$  is more appropriate for describing the noise behavior in post-demosaicking, post-gamma-correction image data, which is outside of the scope of this paper. We also exclude the considerations for censoring, which we handle by excluding the pixels that are close to saturation.

Thus noise parameters  $(\alpha, \beta)$  in (2.8) or  $(a, b)$  in (2.2) and (2.3) above must be estimated in order to characterize the noise behavior in Figure 2.1. Specifically, both sets of parameters establish the affine relationship between the observed signal  $\mathbb{E}[\tilde{X}]$  and the noise variance  $\text{var}[\tilde{X}]$  in the following ways:

$$\begin{aligned} \text{var}[\tilde{X}] &= \alpha \mathbb{E}[\tilde{X}] - \beta \\ &= a^{-1} \mathbb{E}[\tilde{X}] + b. \end{aligned} \quad (2.5)$$

The slope and the intercept of the above affine equation establish that the noise parameters  $(\alpha, \beta)$  and  $(a, b)$  are related as follows:

$$\alpha = a^{-1}, \quad \beta = -b. \quad (2.6)$$

By and large,  $(\alpha, \beta)$  and  $(a, b)$  are equivalent. For example, the ‘‘Generalized Anscombe transform’’ technique developed in [9] for the Gaussian-Poisson model in (2.2) is mathematically identical to applying ordinary Anscombe to the Poisson noise model  $\hat{X} = (\tilde{X} - \beta)\alpha^{-1}$  in (2.8).

Most image and video denoising techniques designed to handle heteroskedastic noise require the noise parameters. These parameters determine the degree to which the denoising functions smooth out the image/video details. Methods leveraging the normal approximation treat the noise according to the estimated noise model parameters  $(\hat{a}, \hat{b})$  to infer the variance of noise  $(\sqrt{\Lambda + \hat{b}})$  at each pixel. Due to the lack of direct access to the pixel value  $\Lambda$ , it is often necessary to approximate the actual noise model by a proxy for  $\Lambda$ , such as observed  $(\Lambda \approx \tilde{X}/\hat{a})$  or denoised pixel values. Alternatively, Poisson-based denoising techniques are designed to estimate clean image  $\lambda$  from a Poisson image  $X$ . In this scenario, we reconstruct  $X$  using estimated parameters  $(\hat{\alpha}, \hat{\beta})$ , as follows [34]:

$$\hat{X} = \frac{\tilde{X} - \hat{\beta}}{\hat{\alpha}}. \quad (2.7)$$

The recovered Poisson image  $\hat{X}$  is subsequently used as an input to the denoising function. The advantage to this type of denoising is that no ad-hoc preprocessing step is required to infer the noise variance at each pixel because the notion of heteroskedastic noise is already captured by the Poisson likelihood function precisely [15, 19, 20, 26, 35]. We draw conclusion that the noise parameters  $(\alpha, \beta)$  or  $(a, b)$  must be known *a priori* or estimated accurately in order to carry out the denoising task effectively.

### 2.1.2 Noise Parameter Estimation Techniques

The conventional method to determine  $(\alpha, \beta)$  or  $(a, b)$  is to calibrate the sensor data using known signal targets, such as Gretag Macbeth Colorchecker shown in Figure 2.1(a) [36]. Taking empirical mean and variance of homogeneous patches as proxies for  $\mathbb{E}[\tilde{X}]$  and  $\text{var}[\tilde{X}]$ , one can use regression to find the slope and intercept in (2.5), as shown in Figure 2.1(b). Besides the fact that the calibration is impractical for consumer applications, however, the thermal noise level is affected by the environment temperature as well as the heat that the

sensor itself generates over extended usage and the gain that dynamically changes to adopt to scene. See Figure 2.1(b,c) for an evidence of noise parameter drift over time—under the same exact camera settings, the calibrated noise parameters differed over two separate days.

Thus noise parameter estimation from natural scenes to adjust denoising for dynamic noise conditions is an attractive alternative to re-calibration after some period of usage. Most existing noise parameter estimation methods rely on homogeneous patch detection [1–3,22,37,38]. The empirical mean and variance of detected homogeneous patches can be used as proxies for  $\mathbb{E}[\tilde{X}]$  and  $\text{var}[\tilde{X}]$ , similar to the calibration steps using the Gretag Macbeth Colorchecker described above. Conventional hypothesis tests for identifying homogeneous regions of sensor data in the presence of noise (such as Student’s-t and Welch) are designed to exclude (a group of) samples  $n \in \ell^2$  with large differences  $\tilde{X}_n - \tilde{X}_{n+1}$  (where  $n$  and  $n+1$  denote adjacent pixel location). However, thresholding noisy pixels is challenging for several reasons—large difference magnitude may be due to noise rather than differences in pixel intensities; and by determining the membership of a patch based on noisy pixel differences, we bias empirical variance to be smaller than the actual  $\text{var}[\tilde{X}]$ . Consequently, methods in [1,3,22,37,38] seek robust alternatives to determining homogeneous patches. In addition, method in [2] use lower envelope fitting to infer noise parameters to safeguard against empirical variance of a segmented region that is not completely homogeneous. However, patch-based methods are sensitive to scene contents, and fail in presence of textured regions.

Alternatively, noise parameter estimation method in [23] used a combination of variance stabilization transfer (VST) and “blank images” (captured with the mechanical shutter closed) to increase robustness to scene contents. While it eliminated the need for detecting homogeneous patches, it is not compatible with cameras with no mechanical shutter, such as video cameras and smartphone cameras.

Recent trends in DSLR and smartphone cameras leverage multi-shot imaging, exploiting temporal redundancy across images taken in a burst or using video frames. Indeed, the noise parameter estimation technique developed in this paper overcome the sensitivities to the scene contents by leveraging the joint statistics across multiple frames that are independent of the scene contents, eliminating the need to detect homogeneous patches. However, this comes at the cost of increased sensitivities to camera and scene motion. As described below, we develop robust techniques to overcome these issues.

## 2.2 B: Background and Related Works of Color Image Denoising

### 2.2.1 Review: Noise Model and Denoising Technique for Gray-scale Image

Poisson distribution accurately models noise caused by photon arrival process in low light imaging (also known as “shot” noise) [23, 28]. Let  $\boldsymbol{\lambda} = [\lambda(1), \dots, \lambda(N)]^\top \in \mathbb{R}^N$  denote the latent light intensities, the independent Poisson counts sequence  $\mathbf{F} = [F(1), \dots, F(N)]^\top \in \mathbb{R}^N$  can be described as:

$$\begin{aligned} F(n)|\lambda(n) &\sim \mathcal{P}(\lambda(n)), \\ Pr[F(n)|\lambda(n)] &= \frac{e^{-\lambda(n)} \lambda(n)^{F(n)}}{F(n)!}, \end{aligned} \quad (2.8)$$

where  $n$  indicates the pixel location. For two independent variables  $F(n)|\lambda(n) \sim \mathcal{P}(\lambda(n))$  and  $F(m)|\lambda(m) \sim \mathcal{P}(\lambda(m))$ ,  $n \neq m$ , there are two key relations with respect to (2.8).

Firstly, the distribution of variable summation is still Poisson:

$$F(n) + F(m)|\boldsymbol{\lambda} \sim \mathcal{P}(\lambda(n) + \lambda(m)). \quad (2.9)$$

Secondly, the distribution of variable difference is called Skellam:

$$\begin{aligned} F(n) - F(m)|\boldsymbol{\lambda} &\sim \mathit{Skellam}(\lambda(n) + \lambda(m), \lambda(n) - \lambda(m)), \\ Pr[F(n) - F(m)|\boldsymbol{\lambda}] &= e^{-(\lambda(n) + \lambda(m))} \left( \frac{\lambda(n)}{\lambda(m)} \right)^{\frac{F(n) - F(m)}{2}} \\ &\times \mathcal{I}_{F(n) - F(m)} \left( \sqrt{(\lambda(n) + \lambda(m))^2 - (\lambda(n) - \lambda(m))^2} \right), \end{aligned} \quad (2.10)$$

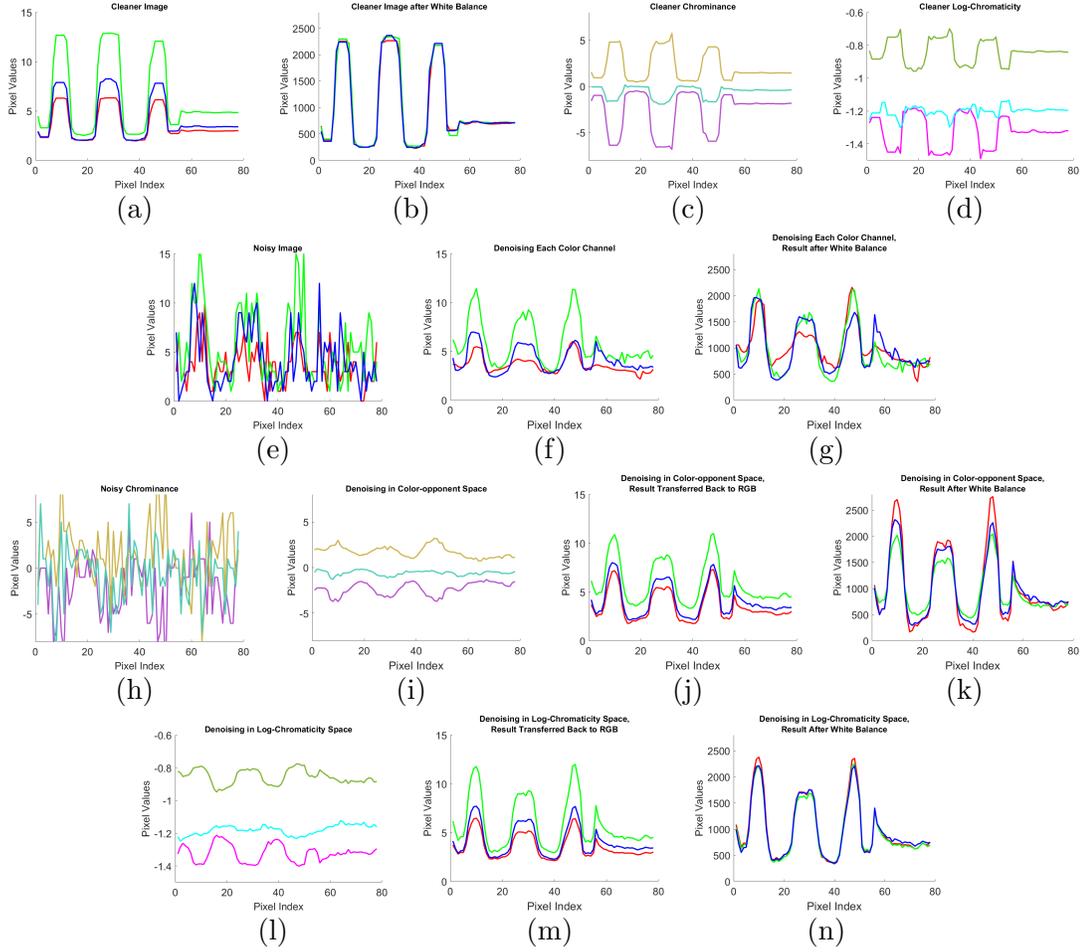


Figure 2.2: Color denoising comparison of real camera sensor data. The plots represent the pixel values of a line crossing black/white/grey patches. Scene was captured using Sony  $\alpha 7R III$  with pixel-shifting mode to capture the full resolution R/G/B data. Plots in (a,b,c,d) correspond to the long-exposure image, which we use as a proxy of cleaner reference. Plots in (e,f,g) illustrate image denoising process in each color channel. Plots in (h,i,j,k) illustrate image denoising process in color-opponent space. Plots (e,l,m,n) illustrate the proposed image denoising process in log-chromaticity space. More specifically, for the plots in (a,e,f,j,m), the red/green/blue lines represent the R/G/B values of sensor data, respectively. For the plots in (b,g,k,n), the red/green/blue lines represent the R/G/B values after white balance. For the plots in (c,h,i), the purple lines denote the chrominance values R-G of sensor data. Similarly, the gold lines denote the G-B values. And the emerald-green lines denote the R-B values. For the plots in (d,l), the magenta lines represent the log-chromaticity value corresponding to  $\log \frac{R}{R+G+B}$ . The darker-green lines correspond to  $\log \frac{G}{R+G+B}$ . And the cyan lines correspond to  $\log \frac{B}{R+G+B}$ . From (b), we know the latent pixel intensities are in black/white/grey. Any pixels with obvious imbalance in R/G/B after white balance will create artificial color artifacts. Clearly, the performance of (n) > (k) > (g). The (i,j,k) and (l,m,n) share the same denoised luminance component, which does not show in (c,d,h,i,l). Because luminance represents the brightness only, which is separated from hue components. In this figure, we mainly focus on the comparison of color processing.

where  $\mathcal{I}_{F(n)-F(m)}(\cdot)$  is the modified Bessel function of the first kind.

Poisson image denoising is often performed in transform domains such as Haar wavelet. Because the denoising gain benefits from the sparse property. Define  $\mathbf{F} = \mathbf{t}^0$ , let  $\mathbf{F} \mapsto (\mathbf{t}^J, \mathbf{X}^J, \dots, \mathbf{X}^1)$  denotes a  $J$ -level Haar wavelet transformation of noisy image. With this transformation, the noisy image  $\mathbf{F}$  is decomposed into multi-level wavelet coefficients  $\mathbf{X}^j = [\dots, X^j(n), \dots]^\top$  and scaler coefficients  $\mathbf{t}^j = [\dots, t^j(n), \dots]^\top$ ,  $\forall j \in [0, 1, \dots, J]$ , which are comprised of:

$$\begin{cases} \text{scaling coefficients: } t^j(n) = t^{j-1}(2n) + t^{j-1}(2n+1) \\ \text{wavelet coefficient: } X^j(n) = t^{j-1}(2n) - t^{j-1}(2n+1). \end{cases} \quad (2.11)$$

Similarly, let  $\boldsymbol{\lambda} \mapsto (\boldsymbol{\mu}^J, \boldsymbol{\kappa}^J, \dots, \boldsymbol{\kappa}^1)$  denotes the Haar wavelet transfer of clean intensity  $\boldsymbol{\lambda}$  (which cannot be observed directly). Combine (2.9), (2.10), and (2.11), naturally, the following relations hold:

$$t^j(n) | \mu^j(n) \sim \mathcal{P}(\mu^j(n)), \quad (2.12)$$

$$X^j(n) | \mu^j(n), \kappa^j(n) \sim \text{Skellam}(\mu^j(n), \kappa^j(n)). \quad (2.13)$$

In turn, the typical wavelet based denoising task is addressed by [4]: 1) let  $\widehat{\boldsymbol{\mu}}^j = \mathbf{t}^j$ , and reconstruct denoised wavelet coefficient  $\widehat{\boldsymbol{\kappa}}^j$ ; 2) proceed the inverse Haar wavelet transform to yield denoising result  $\widehat{\boldsymbol{\lambda}}$ . For convenient, the level index  $j$ 's are omitted in the following discussion. Instead we simply define the clean and noisy scaling coefficients  $\mu = \lambda(2n) + \lambda(2n+1)$  and  $t = F(2n) + F(2n+1)$ , and the clean and noisy wavelet coefficients  $\kappa = \lambda(2n) - \lambda(2n+1)$  and  $X = F(2n) - F(2n+1)$ , respectively. Define a function  $\gamma[\widehat{\boldsymbol{\kappa}}]$  predicts the  $\ell^2$  error of a given wavelet estimator  $\widehat{\boldsymbol{\kappa}}$  such that:

$$\gamma[\widehat{\boldsymbol{\kappa}}] = \mathbb{E} \|\boldsymbol{\kappa} - \widehat{\boldsymbol{\kappa}}\|^2 \quad (2.14)$$

In [4], Wu proposed a wavelet shrinkage operator produces denoised wavelet coefficients with minimum attainable  $\ell^2$  error.

**Theorem 1.** *Let  $\mathbf{F}|\boldsymbol{\lambda}$  as defined in (2.8), and  $\mathbf{X}$ ,  $\mathbf{t}$ ,  $\boldsymbol{\kappa}$ , and  $\boldsymbol{\mu}$  as defined in (2.12) and (2.13). Define minimum risk wavelet shrinkage operator(MRSO)  $\kappa_{MR}(t, X) \in \ell^2(\mathbb{R}) \rightarrow \mathbb{R}$  as:*

$$\widehat{\kappa}_{MR}(t, X) = \begin{cases} \widehat{\psi}_{UMR}(X) & \text{if } \mathbb{E}[t] \approx \frac{1}{N} \sum_n t(n) > \text{threshold} \\ \widehat{\psi}_{BMR}(t, X), & \end{cases} \quad (2.15)$$

where

$$\begin{aligned} \widehat{\psi}_{BMR}(X, t) &= X \frac{Pr(X+1, t+1) + Pr(X-1, t+1)}{2Pr(X, t)} \\ &\quad - (t+2) \frac{Pr(X+1, t+1) - Pr(X-1, t+1)}{2Pr(X, t)}, \end{aligned} \quad (2.16)$$

and

$$\begin{aligned} \widehat{\psi}_{UMR}(X) &= \frac{X+1 + \mathbb{E}[t|X+1]}{2} \frac{Pr[X+1]}{Pr[X]} \\ &\quad + \frac{X-1 - \mathbb{E}[t|X-1]}{2} \frac{Pr[X-1]}{Pr[X]}. \end{aligned} \quad (2.17)$$

Let  $\widehat{\boldsymbol{\kappa}}_{MR}(\mathbf{X}, \mathbf{t}) = [\cdots, \widehat{\kappa}_{MR}(n), \cdots]^\top$ ,  $\forall n \in [1, N/2]$ , this MRSO function minimizes the risk function in (2.14), that is:

$$\forall \widehat{\boldsymbol{\kappa}} \in \mathbb{R}^2, \quad \gamma[\widehat{\boldsymbol{\kappa}}_{MR}] \leq \gamma[\widehat{\boldsymbol{\kappa}}] \quad (2.18)$$

At last, the reconstructed image will be obtained by inverse wavelet transform  $\widehat{\boldsymbol{\lambda}}_{MR}(\mathbf{F}) \leftarrow (\widehat{\mathbf{t}}^J, \widehat{\boldsymbol{\kappa}}_{MR}^J(\mathbf{X}^J, \mathbf{t}^J), \cdots, \widehat{\boldsymbol{\kappa}}_{MR}^1(\mathbf{X}^1, \mathbf{t}^1))$ .

The works in [4] proved that the MRSO denoising  $\widehat{\boldsymbol{\lambda}}_{MR}(\mathbf{F})$  yield visually optimal and quantitatively higher PSNR result for gray-scale image.

## 2.2.2 Image Denoising of Each Color Channel.

Let  $k \in \{\mathcal{R}, \mathcal{G}, \mathcal{B}\}$  be the color channel index. Denote by  $\boldsymbol{\lambda} = [\boldsymbol{\lambda}_{\mathcal{R}}^\top, \boldsymbol{\lambda}_{\mathcal{G}}^\top, \boldsymbol{\lambda}_{\mathcal{B}}^\top]^\top \in \mathbb{R}^{3N}$  a latent multivariate sequence of interest, where  $\boldsymbol{\lambda}_k = [\lambda_k(1), \dots, \lambda_k(N)]^\top \in \mathbb{R}^N$  is a vectored

value referred to as the image intensity of  $k$ -th color channel. In multivariate inhomogeneous Poisson sequence problems, we assume that the true intensities of color image is observed via heteroschedastically noisy measurement  $\mathbb{F}$ . Specifically, denote by  $\mathbb{F} = [\mathbf{F}_{\mathcal{R}}^{\top}, \mathbf{F}_{\mathcal{G}}^{\top}, \mathbf{F}_{\mathcal{B}}^{\top}]^{\top} \in \mathbb{R}^{3N}$  the noisy measurements of  $\lambda$  is an independent Poisson random sequence such as:

$$F_k(n) | \lambda_k(n) \sim \mathcal{P}(\lambda_k(n)). \quad (2.19)$$

It is straightforward to use MRSO function [4] for color image denoising as follows:  $\widehat{\lambda}(\mathbb{F}) = [\widehat{\lambda}_{MR}^{\top}(\mathbf{F}_{\mathcal{R}}), \widehat{\lambda}_{MR}^{\top}(\mathbf{F}_{\mathcal{G}}), \widehat{\lambda}_{MR}^{\top}(\mathbf{F}_{\mathcal{B}})]^{\top} \in \mathbb{R}^{3N}$ . Figure 2.2(e,f,g) illustrates the process of denoising each color channel independently. As we introduced in previous section, the color artifacts in (g) is spatially inconsistent.

### 2.2.3 Image Denoising in Color-opponent Space.

Alternatively, color-opponent space description transfers R/G/B color image into the luminance component by the linear summation of all color channels, and the chrominance component by the linear subtraction of different channels. Usually, the studies based on the additive white Gaussian noise (AWGN) or based on the Gaussian approximation to Poisson conveniently leverage the gray-scale study to luminance and chrominance denoising because the variance of added white noise can linearly pass through the color-opponent transfer, which is no longer hold for Poisson image. Next, we propose a practical Poisson color denoising strategy in color-opponent space.

**Proposition 1.** *Let  $\mathbb{F} | \lambda$  as defined in (2.19). Denote by  $\boldsymbol{\lambda}_{\mathcal{L}} \in \mathbb{R}^N$  and  $\mathbf{F}_{\mathcal{L}} \in \mathbb{R}^N$  the clean and noisy luminance that are computed as:*

$$\boldsymbol{\lambda}_{\mathcal{L}} = \boldsymbol{\lambda}_{\mathcal{R}} + \boldsymbol{\lambda}_{\mathcal{G}} + \boldsymbol{\lambda}_{\mathcal{B}}, \quad (2.20)$$

$$\mathbf{F}_{\mathcal{L}} = \mathbf{F}_{\mathcal{R}} + \mathbf{F}_{\mathcal{G}} + \mathbf{F}_{\mathcal{B}}. \quad (2.21)$$

As illustrated in Figure 2.2(c) and (h), denote by  $\mathbf{C}_j$  and  $\mathbf{D}_j, \forall j \in [1, 2, 3]$  the  $j$ -th clean and noisy chrominance element, the computations of chrominance are described as  $\mathbf{C}_j = \boldsymbol{\lambda}_{k1} - \boldsymbol{\lambda}_{k2} \in \mathbb{R}^N$  and  $\mathbf{D}_j = \mathbf{F}_{k1} - \mathbf{F}_{k2} \in \mathbb{R}^N$ , respectively. Specifically, we define the color-opponent space transfer of noisy color image as example:

$$\begin{bmatrix} \mathbf{F}_{\mathcal{L}}^{\top} \\ \mathbf{D}_1^{\top} \\ \mathbf{D}_2^{\top} \\ \mathbf{D}_3^{\top} \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & -1 & 0 \\ 0 & 1 & -1 \\ 1 & 0 & -1 \end{bmatrix} \times \begin{bmatrix} \mathbf{F}_{\mathcal{R}}^{\top} \\ \mathbf{F}_{\mathcal{G}}^{\top} \\ \mathbf{F}_{\mathcal{B}}^{\top} \end{bmatrix}. \quad (2.22)$$

With the help of the properties described in (2.8), (2.9), and (2.10), the distribution of noisy image in color-opponent space follows:

$$F_{\mathcal{L}}(n)|\lambda \sim \mathcal{P}(\lambda_{\mathcal{R}}(n) + \lambda_{\mathcal{G}}(n) + \lambda_{\mathcal{B}}(n)), \quad (2.23)$$

$$D_1(n)|\lambda, \mathbf{C}_1 \sim \text{Skellam}(\lambda_{\mathcal{R}}(n) + \lambda_{\mathcal{G}}(n), \mathbf{C}_1(n)),$$

$$D_2(n)|\lambda, \mathbf{C}_2 \sim \text{Skellam}(\lambda_{\mathcal{R}}(n) + \lambda_{\mathcal{B}}(n), \mathbf{C}_2(n)), \quad (2.24)$$

$$D_3(n)|\lambda, \mathbf{C}_3 \sim \text{Skellam}(\lambda_{\mathcal{G}}(n) + \lambda_{\mathcal{B}}(n), \mathbf{C}_3(n)),$$

$\forall n \in [1, N]$ . Naturally, the reconstruction of clean luminance  $\widehat{\boldsymbol{\lambda}}_{\mathcal{L}}$  applies the Poisson image denoising. And the reconstruction of clean chrominance  $\widehat{\mathbf{C}}_j$  applies Skellam operator.

Finally, the inverse color transfer yields the denoised color image such that:

$$\begin{bmatrix} \widehat{\boldsymbol{\lambda}}_{\mathcal{R}} \\ \widehat{\boldsymbol{\lambda}}_{\mathcal{G}} \\ \widehat{\boldsymbol{\lambda}}_{\mathcal{B}} \end{bmatrix} = \frac{1}{3} \begin{bmatrix} 1 & 1 & 0 & 1 \\ 1 & -1 & 1 & 0 \\ 1 & 0 & -1 & -1 \end{bmatrix} \times \begin{bmatrix} \widehat{\boldsymbol{\lambda}}_{\mathcal{L}} \\ \widehat{\mathbf{C}}_1 \\ \widehat{\mathbf{C}}_2 \\ \widehat{\mathbf{C}}_3 \end{bmatrix}. \quad (2.25)$$

Here we use MRSO as example to illustrate the Poisson-Skellam based denoising in color-opponent space. As we discussed in Section 2.2.1, the denoised luminance  $\widehat{\boldsymbol{\lambda}}_{MR}(\mathbf{F}_{\mathcal{L}})$  with minimum Skellam risk is calculated as:

$$\widehat{\boldsymbol{\lambda}}_{MR}(\mathbf{F}_{\mathcal{L}}) \leftarrow (\widehat{\mathbf{t}}_{\mathcal{L}}^J, \widehat{\boldsymbol{\kappa}}_{MR}^J(\mathbf{X}_{\mathcal{L}}^J, \mathbf{t}_{\mathcal{L}}^J), \dots, \widehat{\boldsymbol{\kappa}}_{MR}^1(\mathbf{X}_{\mathcal{L}}^1, \mathbf{t}_{\mathcal{L}}^1)). \quad (2.26)$$

Meanwhile, as demonstrated in Figure 2.2(i), the MRSO denoised chrominances are computed as:

$$\begin{aligned}
\hat{\mathbf{C}}_1 &= \hat{\kappa}_{MR}(\mathbf{D}_1, \mathbf{F}_R + \mathbf{F}_G), \\
\hat{\mathbf{C}}_2 &= \hat{\kappa}_{MR}(\mathbf{D}_2, \mathbf{F}_G + \mathbf{F}_B), \\
\hat{\mathbf{C}}_3 &= \hat{\kappa}_{MR}(\mathbf{D}_3, \mathbf{F}_R + \mathbf{F}_B).
\end{aligned}
\tag{2.27}$$

Moreover, Figure 2.2(j) demonstrates the denoised R/G/B value after the inverse color transfer (2.25). And Figure 2.2(k) is the denoised image after white balance. From (h) to (j), the denoising function tries to keep the gap between different chrominances as constant so that the (k) can overcome the spatial inconsistency issue existing in (g). However, compare with (b) and (c), the color-opponent transfer is not capable to normalize the changes of the latent ratio in the “texture” area. Therefore, it may produce false color after white-balance. More examples will be discussed in Section 5.2.3.

From now on, we will gradually introduce the proposed color denoising in log-chromaticity coordinates, and verify that the ratio based color processing is more spatially invariant.

## CHAPTER III

### A: PROPOSED THEOREMS OF PARAMETER ESTIMATION

#### 3.1 Theoretical Results: Binomial Statistics

In this section, we develop the requisite theory of binomial statistics, which describes the joint statistics across multiple frames. We adopt the notational conventions of the Poisson noise parameterization in (2.8). But for notational simplicity, we develop the theory below using  $\lambda$  only, without the loss of mathematical generality when extending to  $\lambda + \xi$  as the affine relationship in (2.5) remains intact the same way. In the subsequent discussion, we let  $\tilde{X}_n$  and  $\tilde{Y}_n$  denote  $n$ th pixel in two adjacent frames in a video sequence. We denote the corresponding latent pixel intensity values by  $\lambda_n$  and  $\mu_n$ , respectively. The subscript  $n$  is omitted whenever it is understood from the context.

**Lemma 1.** *Let  $X|\lambda \sim \mathcal{P}(\lambda)$  and  $Y|\mu \sim \mathcal{P}(\mu)$  be two independent random variables. Define  $Z := X + Y$  and  $\theta := \frac{\lambda}{\lambda + \mu} \in [0, 1]$ . Then  $Z|\lambda, \mu$  and  $X|\lambda, \mu, Z$  are Poisson and binomial*

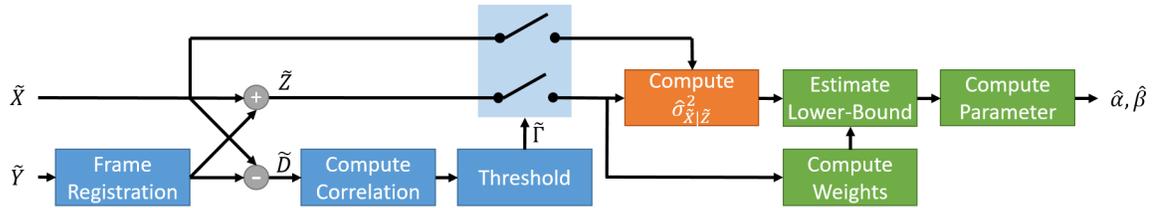


Figure 3.1: Block diagram of the proposed Poisson noise parameter estimation method. The binomial statistics in (3.27) are computed in the orange block. The main steps for estimating the noise parameters (lower-bound linear regression in (3.31) and parameter estimation in (3.26)) appear in green blocks. The blue blocks carry out the pixel registration and selection steps outlined in Section 3.2.3 to ensure that the proposed method is robust to foreground and background motion.

random variables of the form

$$Z|\lambda, \mu \sim \mathcal{P}(\lambda + \mu) \quad (3.1)$$

$$X|\lambda, \mu, Z \sim \mathcal{B}(Z, \theta). \quad (3.2)$$

*Proof.* Recall that the characteristic function  $\Phi_X(\omega) := \mathbb{E}[e^{j\omega X}|\lambda]$  of Poisson random variable  $X|\lambda$  is the discrete time Fourier transform of the conditional probability  $P[X = k|\lambda]$ :

$$\Phi_X(\omega) = \sum_k e^{j\omega k} P[X = k|\lambda] = e^{\lambda(e^{j\omega} - 1)}. \quad (3.3)$$

By the well-known property of characteristic functions, the following relation holds for a sum of independent random variables  $X|\lambda$  and  $Y|\mu$ :

$$\Phi_Z(\omega) = \Phi_X(\omega)\Phi_Y(\omega) = e^{(\lambda+\mu)(e^{j\omega} - 1)}. \quad (3.4)$$

Inverting Fourier transform of  $\Phi_Z(\omega)$  proves (3.1).

By definition of conditional probability,

$$P[X = k|\lambda, \mu, Z = z] = \frac{P[X = k, Z = z|\lambda, \mu]}{P[Z = z|\lambda, \mu]}. \quad (3.5)$$

Substituting (3.1),

$$\begin{aligned} P[X = k|\lambda, \mu, Z = z] &= \frac{P[X = k, Y = z - k|\lambda, \mu]}{P[Z = z|\lambda, \mu]} \\ &= \left( \frac{e^{-\lambda-\mu} \lambda^k \mu^{z-k}}{k!(z-k)!} \right) \left( \frac{e^{-\lambda-\mu} (\lambda + \mu)^z}{z!} \right)^{-1} \\ &= \frac{z!}{k!(z-k)!} \left( \frac{\lambda}{\lambda + \mu} \right)^k \left( \frac{\mu}{\lambda + \mu} \right)^{z-k}. \end{aligned} \quad (3.6)$$

Substituting  $\theta := \frac{\lambda}{\lambda + \mu}$  proves (3.2).  $\square$

If  $X$  and  $Y$  are noisy measurements of the true pixel intensity values  $\lambda$  and  $\mu$ , then  $\theta = \frac{\lambda}{\lambda + \mu}$  is a ratio describing the *relative* intensities of  $\lambda$  and  $\mu$ . That is,  $\theta = \frac{1}{2}$  means  $\lambda$  and

$\mu$  have equal pixel intensity;  $\theta > \frac{1}{2}$  implies  $\lambda > \mu$  and  $\theta < \frac{1}{2}$  implies  $\lambda < \mu$ . Interpreting  $\theta$  as a random variable,

$$\mathbb{E}[\theta|Z] = \frac{1}{2} \quad (3.7)$$

is a reasonable assumption, because the sum  $Z = X + Y$  does not provide any meaningful information about whether  $\lambda$  or  $\mu$  is greater on average.

**Theorem 2.** *The variance of  $X|Z$  is*

$$\sigma_{X|Z}^2 = \frac{Z}{4} + \sigma_{\theta|Z}^2(Z^2 - Z). \quad (3.8)$$

*Proof.* By the law of total variance, the following relation holds:

$$\sigma_{X|Z}^2 = \mathbb{E} \left[ \text{Var}(X|Z, \theta) \middle| Z \right] + \text{Var} \left( \mathbb{E}[X|Z, \theta] \middle| Z \right). \quad (3.9)$$

The mean and the variance of the Binomial random variable  $X|\lambda, \mu, Z$  are:

$$\mathbb{E}[X|Z, \theta] = Z\theta \quad (3.10)$$

$$\text{Var}(X|Z, \theta) = Z\theta(1 - \theta). \quad (3.11)$$

Evaluating the conditional variance of (3.10), we have

$$\text{Var} \left( \mathbb{E}[X|Z, \theta] \middle| Z \right) = \text{Var} \left( Z\theta \middle| Z \right) = Z^2 \sigma_{\theta|Z}^2. \quad (3.12)$$

Similarly, we assess the conditional mean of (3.11):

$$\begin{aligned} \mathbb{E} \left[ \text{Var}(X|Z, \theta) \middle| Z \right] &= Z \mathbb{E} \left[ \theta(1 - \theta) \middle| Z \right] \\ &= Z(\mathbb{E}[\theta|Z] - \mathbb{E}[\theta^2|Z]). \end{aligned} \quad (3.13)$$

Rewriting  $\mathbb{E}[\theta^2|Z]$  as  $\sigma_{\theta|Z}^2 + \mathbb{E}[\theta|Z]^2$  and substituting (3.7),

$$\mathbb{E} \left[ \text{Var}(X|Z, \theta) \middle| Z \right] = \frac{Z}{4} - Z\sigma_{\theta|Z}^2. \quad (3.14)$$

Combining (3.12) and (3.14) proves the Theorem.  $\square$

**Corollary 1.** Define new random variables  $\tilde{X}$ ,  $\tilde{Y}$ , and  $\tilde{Z}$  by the relation

$$\tilde{X} = \alpha X + \beta, \quad \tilde{Y} = \alpha Y + \beta \quad (3.15)$$

$$\tilde{Z} = \tilde{X} + \tilde{Y} = \alpha Z + 2\beta. \quad (3.16)$$

Then the variance of  $\tilde{X}|\tilde{Z}$  is

$$\sigma_{\tilde{X}|\tilde{Z}}^2 = \frac{\alpha}{4}\tilde{Z} - \frac{\alpha\beta}{2} + \sigma_{\theta|Z}^2((\tilde{Z} - 2\beta)^2 - \alpha(\tilde{Z} - 2\beta)). \quad (3.17)$$

*Proof.* By (3.15) and (3.16),

$$\begin{aligned} P[\tilde{X} = \tilde{k}|\tilde{Z} = \tilde{z}] &= P[\alpha X + \beta = \tilde{k}|\tilde{Z} = \tilde{z}] \\ &= P[X = k|Z = z], \end{aligned} \quad (3.18)$$

where

$$k = \alpha^{-1}(\tilde{k} - \beta), \quad z = \alpha^{-1}(\tilde{z} - 2\beta). \quad (3.19)$$

Thus

$$\text{Var}(\tilde{X}|\tilde{Z}) = \text{Var}(\alpha X + \beta|Z) = \alpha^2 \sigma_{X|Z}^2. \quad (3.20)$$

Substituting (3.8) and (3.16) into (3.20) proves the Corollary.  $\square$

**Corollary 2.** Define  $\tilde{X}$ ,  $\tilde{Y}$ , and  $\tilde{Z}$  as before. Then

$$\mathbb{E}[\tilde{X}|\tilde{Z}] = \tilde{Z}/2 \quad (3.21)$$

*Proof.* Recall (3.7). Then

$$\mathbb{E}[\tilde{X}|\tilde{Z}] = \mathbb{E}[\alpha X + \beta|\tilde{Z}] = \mathbb{E}[\alpha\theta Z + \beta|\tilde{Z}]. \quad (3.22)$$

By the assumption (3.7) and the invertibility of (3.16), we have

$$\mathbb{E}[\tilde{X}|\tilde{Z}] = \mathbb{E}[\tilde{X}|Z] = \alpha Z/2 + \beta. \quad (3.23)$$

Substituting (3.16) proves (3.21).  $\square$

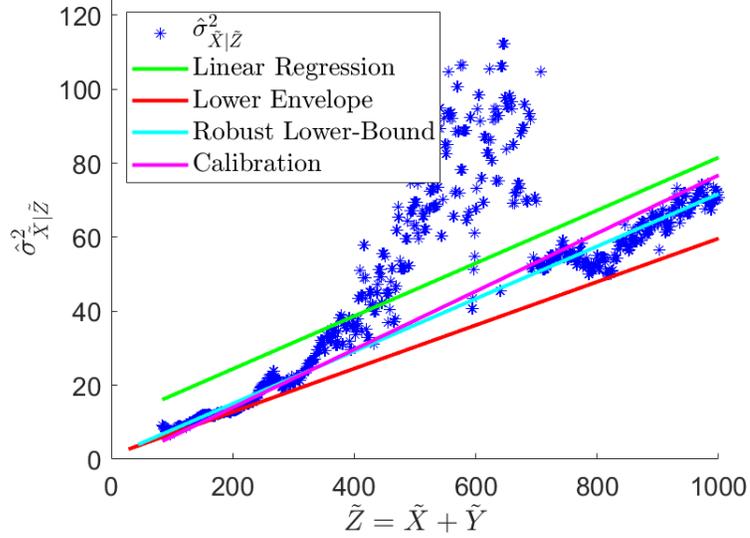


Figure 3.2: Demonstration of the proposed noise parameter estimation. The blue points denote the empirical binomial statistics  $\hat{\sigma}_{\tilde{X}|\tilde{Z}_n=\tilde{z}}^2$  in (3.27). The linear regression in (3.28) and lower envelope fit in (3.29) are shown by green and red lines, respectively. The cyan line denotes the proposed robust lower-bound function  $f(\tilde{z}) = s\tilde{z} + t$  computed using (3.31), based on which we recover the noise parameters using (3.26). For reference, we included the magenta line corresponding to the noise parameters from calibration. Qualitatively, we may assess the accuracy of the parameter estimation by observing the closeness of green/red/cyan lines to the magenta line.

### 3.2 Application: Noise Parameter Estimation

Let  $\tilde{X}_n|\lambda_n$  and  $\tilde{Y}_n|\mu_n$  correspond to the noisy video frames of the latent pixel intensities  $\lambda_n$  and  $\mu_n$  at pixel  $n$ , respectively. Furthermore, we continue to assume that the underlying model for  $\tilde{X}_n|\lambda_n$  and  $\tilde{Y}_n|\mu_n$  is an affine transformation as described in (3.15). Our goal in Poisson noise parameter estimation is to determine the values of  $\alpha$  and  $\beta$  from the properties of  $X|\lambda, \mu, Z$  where  $Z = X + Y$ . The schematic for the proposed algorithm is shown in Figure 3.1.

### 3.2.1 Motivation

Define an affine line  $f : \mathbb{R} \rightarrow \mathbb{R}$  of the form:

$$f(\tilde{z}) := \frac{\alpha}{4}\tilde{z} - \frac{\alpha\beta}{2}. \quad (3.24)$$

Owing to the fact that  $\sigma_{\theta|Z}^2 \geq 0$  and  $Z^2 - Z \geq 0$  (recall  $Z$  is a positive integer), it follows from Corollary (1) that  $\sigma_{\tilde{X}|\tilde{Z}=\tilde{z}}^2$  is lower-bounded by  $f(\tilde{z})$ :

$$\sigma_{\tilde{X}|\tilde{Z}=\tilde{z}}^2 \geq f(\tilde{z}). \quad (3.25)$$

Figure 3.2 confirms that the empirical scatter plot of  $\sigma_{\tilde{X}|\tilde{Z}=\tilde{z}}^2$  against  $\tilde{Z} = \tilde{z}$  is lower-bounded by the function  $f : \mathbb{R} \rightarrow \mathbb{R}$  in (3.25). The key observation is that the noise parameters  $\alpha$  and  $\beta$  can be determined unambiguously from the slope ( $s$ ) and intercept ( $t$ ) of this lower-bound function  $f(\tilde{z}) = s\tilde{z} + t$ :

$$\alpha = 4s, \quad \beta = -\frac{t}{2s}. \quad (3.26)$$

Hence in the remainder of this paper, we focus on developing a novel technique for recovering this lower-bound function  $f : \mathbb{R} \rightarrow \mathbb{R}$  in (3.25). As explained in Section 2.1, the idea to exploit lower-bound function has been developed for patch-based method in [2]. However, the binomial statistics in Corollary 1 and the lower-bound in (3.25) are global (i.e. not patch-based), making the problem fundamentally different.

### 3.2.2 Lower-Bound Function

Given sensor observations  $\tilde{X}_n$  and  $\tilde{Y}_n$ , we may compute  $\tilde{Z}_n = \tilde{X}_n + \tilde{Y}_n$ , and obtain the *empirical* conditional variance  $\hat{\sigma}_{\tilde{X}|\tilde{Z}}^2$  for each value of  $\tilde{Z} = \tilde{z}$  by the following formula:

$$\hat{\sigma}_{\tilde{X}|\tilde{Z}=\tilde{z}}^2 = \frac{1}{\#\{n \in \mathbb{Z}^2 : \tilde{Z}_n = \tilde{z}\}} \sum_{n \in \mathbb{Z}^2 : \tilde{Z}_n = \tilde{z}} \left( \tilde{X}_n - \frac{\tilde{z}}{2} \right)^2. \quad (3.27)$$

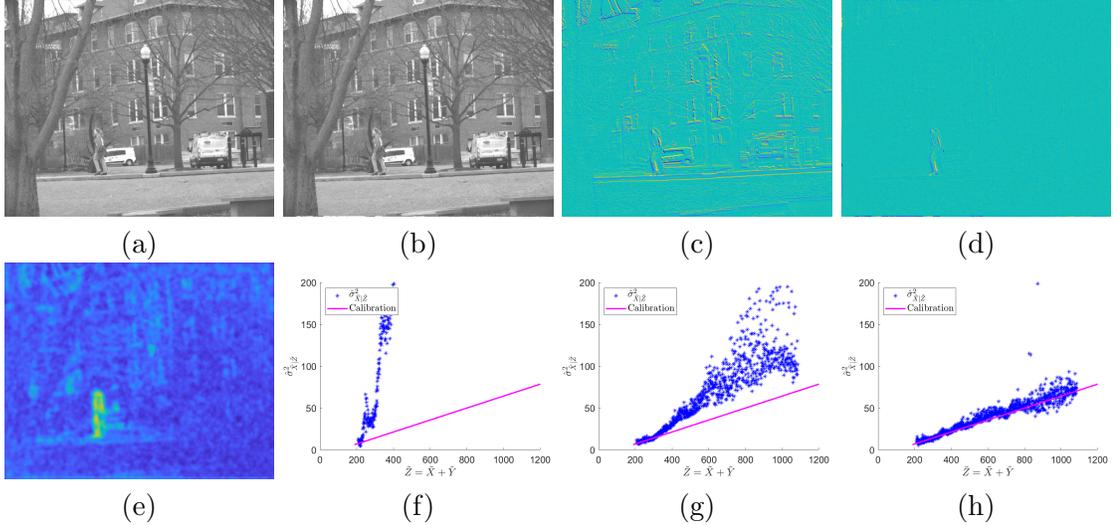


Figure 3.3: Illustration of how proposed method gains robustness to foreground and background motions. (a,b) Two adjacent frames of a video sequence (which we refer to as “frame 1” and “frame 2”, respectively). (c,d) Difference of frames 1 and 2 ( $\tilde{D}_n = \tilde{X}_n - \tilde{Y}_n$ ) before and after frame registration, respectively. Frame registration largely eliminates the background motion caused by camera motion, but the foreground motion remains. (e) Pearson product-moment correlation coefficient of  $sgn(\tilde{D})$  computed on (d). It determines the regions of image affected by foreground motion without relying on the magnitude of  $\tilde{D}$ . (f) Empirical binomial statistics with no motion compensation (as shown in (c)). (g) Empirical binomial statistics after frame registration (as shown in (d)). (h) Empirical binomial statistics after frame registration and thresholding (as shown in (e)). The improvement of addressing foreground and background motion is evidenced by comparing the tightening of clusters in (f), (g), and (h). See text for explanation.

Note that we do not use the “corrected” sample variance (whose denominator in (3.27) would become  $\#\{n \in \mathbb{Z}^2 : \tilde{Z}_n = \tilde{z}\} - 1$ ) because of the fact that the mean value  $\mathbb{E}[\tilde{X}|\tilde{Z}]$  is computed from Corollary 2 and not from the sample mean. In high resolution sensors,  $\hat{\sigma}_{\tilde{X}|\tilde{Z}=\tilde{z}}^2$  is a stable function of  $\tilde{z}$  because of the ensemble averaging in (3.27). Note also that (3.27) is a global statistics, not a local statistics like the patch-based methods in [1–3, 22, 37, 38].

Let  $\Lambda \subset \mathbb{R}$  be the range of valid  $\tilde{z}$  values, based on observation data. We compute the lower-bound function  $f(\tilde{z}) = s\tilde{z} + t$  by fitting a line to  $\hat{\sigma}_{\tilde{X}|\tilde{Z}=\tilde{z}}^2$ . Consider a conventional linear regression of the form

$$(\hat{s}, \hat{t}) = \arg \min_{s,t} \int_{\Lambda} \left( \hat{\sigma}_{\tilde{X}|\tilde{Z}=\tilde{z}}^2 - (s\tilde{z} + t) \right)^2 d\tilde{z}. \quad (3.28)$$

It does not yield the desirable result because the fitted line goes through the center of point cluster (i.e. not lower-bound). On the other hand, imposing a lower-envelope constraint

$$\begin{aligned} (\hat{s}, \hat{t}) = \arg \min_{s,t} \int_{\Lambda} \left( \hat{\sigma}_{\tilde{X}|\tilde{Z}=\tilde{z}}^2 - (s\tilde{z} + t) \right)^2 d\tilde{z} \\ \text{subject to } \hat{\sigma}_{\tilde{X}|\tilde{Z}=\tilde{z}}^2 \geq s\tilde{z} + t \end{aligned} \quad (3.29)$$

is very sensitive to noise and outliers.

We instead propose a novel “robust lower-bound linear regression” method. Based on the intuition that the error term  $\Delta = \hat{\sigma}_{\tilde{X}|\tilde{Z}=\tilde{z}}^2 - s\tilde{z} + t$  should be non-negative most of the time, we design a piece-wise *non-symmetrical* squared error function  $\phi : \mathbb{R} \rightarrow \mathbb{R}$ :

$$\phi(\Delta, \tilde{z}) = \begin{cases} -\Delta & \text{if } \Delta < 0 \\ \gamma_1 \Delta & \text{if } \eta_{\tilde{z}_{min}} \geq \Delta > 0, \\ \gamma_2 \Delta + \eta_{\tilde{z}_{min}} (\gamma_1 - \gamma_2) & \text{if } \Delta > \eta_{\tilde{z}_{min}} \end{cases}, \quad (3.30)$$

where  $\tilde{z}_{min}$  is the smallest  $\tilde{Z}_n = \tilde{X}_n + \tilde{Y}_n$  value present in a frame. The slope parameter values  $\gamma_1$  and  $\gamma_2$  determine the severity of the penalty for the positive error (relative to the negative error). When  $0 < \gamma_2 < \gamma_1 < 1$ , the non-symmetric error function in (3.30) encourages a lower-bound line fitting (because  $\Delta \geq \gamma_1 \Delta \geq \gamma_2 \Delta$ ), while remaining robust

to noise by allowing the possibility of negative error. The parameter  $\eta > 0$  determines the sensitivity of robustness—smaller value of  $\eta$  results in more aggressive lower-bound fitting. In our actual implementation, we solve the following weighted lower-bound regression problem:

$$(\hat{s}, \hat{t}) = \arg \min_{s, t} \int_{\Lambda} w(\tilde{z}) \phi \left( \hat{\sigma}_{\tilde{X}|\tilde{Z}=\tilde{z}}^2 - (s\tilde{z} + t) \right) d\tilde{z}. \quad (3.31)$$

We used heuristical weights of the following form:

$$w(\tilde{z}) = \frac{1}{\kappa + \tilde{z} / \#\{n \in \mathbb{Z}^2 : \tilde{Z}_n = \tilde{z}\}}. \quad (3.32)$$

The rationale for the weights is that the empirical conditional variance  $\hat{\sigma}_{\tilde{X}|\tilde{Z}=\tilde{z}}^2$  in (3.27) is more stable when  $\#\{n \in \mathbb{Z}^2 : \tilde{Z}_n = \tilde{z}\}$  is large; and Theorem 3 suggests that the influence of  $\sigma_{\theta|Z}^2$  grows quadratically with  $\tilde{z}$ . The constant  $\kappa > 0$  prevents division by a small number. Note that (3.31) approaches the strict (and non-robust) lower-envelope estimation in (3.29) as  $\gamma_1 \rightarrow 0$ . The affine line parameters  $(\hat{s}, \hat{t})$  was computed using Matlab's `fminsearch`( $\cdot$ ) function; and the noise parameters  $(\hat{\alpha}, \hat{\beta})$  can be recovered from  $(\hat{s}, \hat{t})$  using (3.26). In practice, we found that median filtering on  $\hat{\sigma}_{\tilde{X}|\tilde{Z}=\tilde{z}}^2$  improved the stability when the spatial resolution is limited.

### 3.2.3 Improvements: Pixel Registration and Selection

Estimating the lower-bound function directly from an empirical scatter plot of  $\sigma_{\tilde{X}|\tilde{Z}}^2$  in Figure 3.2 is sensitive to unknown positive value  $\sigma_{\theta|Z}^2$ . We focus on the fact that an *equality* between  $\sigma_{\tilde{X}|\tilde{Z}=\tilde{z}}^2$  and  $f(\tilde{z})$  in (3.25) is attained when  $\theta_n = \frac{1}{2}$ . (Proof:  $\theta_n := \frac{\lambda_n}{\lambda_n + \mu_n} = 1/2$  is a constant, and so  $\sigma_{\theta|Z}^2 = 0$ .) In video, let  $\lambda_n$  and  $\mu_n$  refer to two consecutive frames. The scenario  $\lambda_n \neq \mu_n$  occurs when either the foreground object movement or the camera movement cause the intensity values of pixel  $n \in \mathbb{Z}^2$  to change between frames pair.

In this work, we improve the lower-bound regression by taking a number of safeguards against the pixel movements. Specifically, we evaluate  $(\hat{s}, \hat{t})$  and  $w(\tilde{z})$  in (3.31) and (3.32) only over the subset  $\Gamma \subset \mathbb{Z}^2$  of pixels satisfying the condition:

$$\Gamma = \{n \in \mathbb{Z}^2 : \lambda_n = \mu_n\}. \quad (3.33)$$

First, modern cameras and smartphones have inertial measurement units (IMU) that measure the camera motion. The inertial measurements can be leveraged to determine the frames affected by camera movements, which should be excluded from the subset  $\Gamma$  because pixels are likely to have shifted. Only the frames under the noise floor of the IMU are usable. (Obviously, one can use a tripod to ensure camera does not move, but this is highly restrictive. We do not assume this.)

Second, (for the remaining frames that were not discarded,) we further correct for frame-to-frame movements by global image registration. Specifically, we use the well-known SURF feature extraction to find the corresponding points across the two frames [39], then we use RANSAC to recover the homography transformation between the two frames [40]. By registering  $\tilde{X}$  to the coordinates of  $\tilde{Y}$  to compensate for the camera motion across frames, background pixels become stationary. See Figures 3.3(c) and 3.3(f) for the improvements.

Third, we develop another technique to handle foreground object movements. Conventional hypothesis tests such as Student's-t and Welch are designed to exclude pixels  $n \in \mathbb{Z}^2$  with large differences  $D_n := X_n - Y_n$ . However, these methods also discard pixels where  $D_n$  is large due to noise (i.e. not because  $\lambda_n \neq \mu_n$ ), degrading the overall quality of  $(\hat{s}, \hat{t})$  estimates because the magnitude of  $\hat{\sigma}_{\tilde{X}|\tilde{Z}}^2$  is reduced. To overcome this problem, we develop a hypothesis test based on the spatial correlation of  $D_n$ , and *not* based on the magnitude of  $D_n$ .

Consider Table 3.1 showing that Pearson product-moment correlation coefficient of the noise of two neighboring pixels ( $X_n$  and  $X_{n+1}$ )

$$\begin{aligned}\rho_n^{X|\lambda} &= \frac{\text{Cov}(X_n, X_{n+1}|\lambda_n, \lambda_{n+1})}{\sqrt{\text{Var}(X_n|\lambda_n) \text{Var}(X_{n+1}|\lambda_{n+1})}} \\ &= \frac{\mathbb{E}[(X_n - \lambda_n)(X_{n+1} - \lambda_{n+1})|\lambda_n, \lambda_{n+1}]}{\sqrt{\mathbb{E}[(X_n - \lambda_n)^2|\lambda_n]\mathbb{E}[(X_{n+1} - \lambda_{n+1})^2|\lambda_{n+1}]}}\end{aligned}\quad (3.34)$$

is negligibly small. (Its experimental methodology is described in Section 5.1.1.) That is, noise  $X_n - \lambda_n$  of raw image sensor data are spatially uncorrelated.<sup>1</sup> By extension,  $\rho^{Y|\mu}$  is also negligible. Hence, the spatial correlation of the difference  $D_n = X_n - Y_n$  is *entirely* due to the difference  $\lambda_n - \mu_n$ . More concretely, we have the following Theorem.

**Theorem 3.** *Suppose that the following approximations hold:*

$$\begin{aligned}\text{Cov}(X_n, X_{n+1}|\lambda_n, \lambda_{n+1}) &\approx 0 \\ \text{Cov}(Y_n, Y_{n+1}|\mu_n, \mu_{n+1}) &\approx 0.\end{aligned}\quad (3.35)$$

Then  $\text{Cov}(D_n, D_{n+1}) \approx \text{Cov}(\lambda_n - \mu_n, \lambda_{n+1} - \mu_{n+1})$ .

*Proof.* By the law of total covariance,

$$\begin{aligned}\text{Cov}(D_n, D_{n+1}) &= \mathbb{E}\left[\text{Cov}(D_n, D_{n+1}|\lambda, \mu)\right] \\ &\quad + \text{Cov}\left(\mathbb{E}[D_n|\lambda, \mu], \mathbb{E}[D_{n+1}|\lambda, \mu]\right).\end{aligned}\quad (3.36)$$

Because  $X_n|\lambda_n$  and  $Y_n|\mu_n$  are independent, this reduces to

$$\begin{aligned}\text{Cov}(D_n, D_{n+1}) &= \text{Cov}(\lambda_n - \mu_n, \lambda_{n+1} - \mu_{n+1}) \\ &\quad + \mathbb{E}\left[\text{Cov}(X_n, X_{n+1}|\lambda) + \text{Cov}(Y_n, Y_{n+1}|\mu)\right] \\ &\approx \text{Cov}(\lambda_n - \mu_n, \lambda_{n+1} - \mu_{n+1}).\end{aligned}\quad (3.37)$$

where the approximation stems from (3.35). □

<sup>1</sup>Recall that the covariance and variance *conditioned* on  $\lambda$  are statistics *only* on noise. These quantities *do not* represent the covariance/variance of the signal.

We conclude from Theorem 3 that we can draw inference on  $\lambda_n - \mu_n$  indirectly by observing the Pearson product-moment correlation coefficient of  $D_n = X_n - Y_n$ . Since Pearson product-moment correlation coefficient is invariant to affine transformation, we propose to threshold the Pearson product-moment correlation coefficient of  $\tilde{D}_n := \tilde{X}_n - \tilde{Y}_n$  instead of  $D_n = X_n - Y_n$  to determine the set  $\Gamma \subset \mathbb{Z}^2$ :

$$\hat{\Gamma} = \left\{ n \in \mathbb{Z}^2 : \left| \frac{\text{Cov}(\tilde{D}_n, \tilde{D}_{n+1})}{\sqrt{\text{Var}(\tilde{D}_n) \text{Var}(\tilde{D}_{n+1})}} \right| < \epsilon \right\}. \quad (3.38)$$

The purpose of (3.38) is to discard the  $D_n$  that is spatially correlated with  $D_{n+1}$ , i.e.  $\lambda_n \neq \mu_n$  that represents pixels of foreground object movements.

The correlation coefficient in (3.38) is sensitive to noise. Inspired in part by work by [41], we overcome this shortcoming by proposing a robust alternative to correlation coefficient in (3.38). Specifically, we compute the correlation coefficient using the “sign” of  $\tilde{D}_n$  instead of  $\tilde{D}_n$  itself:

$$\begin{aligned} \hat{\Gamma} &= \left\{ n \in \mathbb{Z}^2 : \left| \frac{\text{Cov}(\text{sgn}(\tilde{D}_n), \text{sgn}(\tilde{D}_{n+1}))}{\sqrt{\text{Var}(\text{sgn}(\tilde{D}_n)) \text{Var}(\text{sgn}(\tilde{D}_{n+1}))}} \right| < \epsilon \right\} \\ &= \left\{ n \in \mathbb{Z}^2 : \left| \text{Cov}(\text{sgn}(\tilde{D}_n), \text{sgn}(\tilde{D}_{n+1})) \right| < \epsilon \right\}. \end{aligned} \quad (3.39)$$

Heuristically, we found that the threshold  $\epsilon$  can be chosen to be *arbitrarily* small, as long as it is sufficient to keep enough pixels for the noise parameter estimation. See Figures 3.3(d) and 3.3(g) as well as Table 5.1 for evidence of robustness to motion that these technique achieve.

Table 3.1: Spatial correlation of noise in various commercial cameras and smartphones are **uncorrelated**. We collected 100 raw images (Bayer color filter array patterned sensor) of maximally blurred blank walls taken for each camera setup placed on a tripod. We computed Pearson product-moment correlation coefficient of spatially adjacent pixels (which have two different color filters, denoted “cross color”) and two pixels apart (with same color filters). Maximum correlation coefficient magnitude is only 0.0969, most are smaller than 0.05.

Device	ISO – ShutterSpeed	cross color	red	green1	green2	blue
Nikon D5100	100 – 1/15	0.0711	0.0814	0.0974	0.0969	0.0254
	100 – 1/60	0.0127	0.0081	0.0147	0.0149	0.0071
	1250 – 1/15	0.0055	0.0031	0.0045	0.0046	0.0023
	5000 – 1/60	0.0043	0.0025	0.0030	0.0029	0.0022
Nikon D90	200 – 1/30	0.0417	0.0539	0.0533	0.0555	0.0516
	1000 – 1/15	0.0332	0.0426	0.0351	0.0442	0.0431
	1000 – 1/60	0.0199	0.0215	0.0242	0.0294	0.0216
Canon 5D III	200 – 1/60	0.0404	0.0421	0.0635	0.0638	0.0288
	1000 – 1/15	0.0352	0.0606	0.0439	0.0456	0.0111
	1000 – 1/60	0.0499	0.0567	0.0661	0.0672	0.0231
Sony $\alpha$ 7R3	100 – 1/15	0.0201	0.0267	0.0205	0.0159	0.0267
	100 – 1/60	0.0197	0.0262	0.0184	0.0177	0.0258
	2000 – 1/15	0.0051	0.0053	0.0048	0.0116	0.0053
	5000 – 1/60	0.0028	0.0020	0.0017	0.0116	0.0020
Samsung Galaxy S7	50 – 1/15	0.0017	0.0031	0.0030	0.0013	0.0050
	200 – 1/60	-0.006	-0.004	0.0004	-0.007	0.0032
	800 – 1/15	0.0023	0.0064	0.0014	0.0020	0.0067
	5000 – 1/60	0.0028	0.0020	0.0017	0.0116	0.0020
iPhone X	100 – 1/100	0.0439	0.0225	-0.0207	0.0055	0.0136
OnePlus 3	125 – 1/15	0.0168	0.0544	-0.072	-0.071	0.0492
	1000 – 1/15	0.0000	0.0455	-0.086	-0.085	0.0471
	2500 – 1/60	-0.002	0.0570	-0.069	-0.069	0.0692

## CHAPTER IV

### B:PROPOSED THEOREMS OF COLOR IMAGE DENOISING

#### 4.1 Background: Aitchison Geometry

##### 4.1.1 Elementary Operations

Define  $\mathbb{R}_+^m = [0, \infty)^m$  to be a vector space of positive-valued real numbers, where  $\mathbf{x} = [x_1, \dots, x_m]^T \in \mathbb{R}_+^m$  denotes an  $m$ -dimensional non-negative vector. Let

$$\|\mathbf{x}\| := x_1 + \dots + x_m \quad (4.1)$$

denote an  $\ell^1$  norm. (Since  $x_1, \dots, x_m \geq 0$ , absolute value operator is not needed.) In Aitchison Geometry, we define compositional data as  $m$ -dimensional unit simplex  $\Delta^m$ .

**Definition 1.** *Unit simplex  $\Delta^m$  is defined as subset of  $\mathbb{R}_+^m$ , as follows:*

$$\Delta^m := \{\mathbf{x} \in \mathbb{R}_+^m \mid \|\mathbf{x}\| = 1\} \subset \mathbb{R}_+^m. \quad (4.2)$$

Let  $\mathbf{y}, \mathbf{z} \in \Delta^m$  and  $a \in \mathbb{R}$ . Known as *perturbation* and *powering* operators, we respectively define simplex addition  $\oplus : \Delta^m \times \Delta^m \rightarrow \Delta^m$  and simplex multiplication  $\otimes : \mathbb{R} \times \Delta^m \rightarrow \Delta^m$  as follows:

$$\mathbf{y} \oplus \mathbf{z} := \frac{(y_1 z_1, \dots, y_m z_m)^T}{y_1 z_1 + \dots + y_m z_m} \quad (4.3)$$

$$a \otimes \mathbf{y} := \frac{(y_1^a, \dots, y_m^a)^T}{y_1^a + \dots + y_m^a}, \quad (4.4)$$

where the denominators normalizes to ensure that the range spaces of  $\oplus$  and  $\otimes$  are also simplex.

**Proposition 2** (Aitchison vector space). *The perturbation and powering operators in (4.3) satisfy the axioms of a vector space, where the simplex vector  $\mathbf{c} = [\frac{1}{m}, \dots, \frac{1}{m}]^T \in \Delta^m$  is the*

zero vector *with the property*:

$$\mathbf{y} \oplus \mathbf{c} = \mathbf{y} \quad (4.5)$$

$$0 \otimes \mathbf{y} = \mathbf{c}. \quad (4.6)$$

*Proof.* The commutativity and associativity of  $\oplus$  follow directly from the same properties of multiplication. The neutral element in (4.5) and inverse conditions can be verified in a straightforward way:

$$\mathbf{y} \oplus \mathbf{c} = \frac{[\frac{y_1}{m}, \dots, \frac{y_m}{m}]^T}{\frac{y_1}{m} + \dots + \frac{y_m}{m}} = \mathbf{y} \quad (4.7)$$

$$\mathbf{y} \oplus (-1 \otimes \mathbf{y}) = \frac{[y_1 y_1^{-1}, \dots, y_m y_m^{-1}]^T}{y_1 y_1 + \dots + y_m y_m} \quad (4.8)$$

$$\mathbf{y} \oplus \mathbf{c} = \frac{[\frac{y_1}{m}, \dots, \frac{y_m}{m}]^T}{\frac{y_1}{m} + \dots + \frac{y_m}{m}} = \mathbf{y} \quad (4.9)$$

$$0 \otimes \mathbf{y} = \frac{[y_1^0, \dots, y_m^0]^T}{y_1^0 + \dots + y_m^0} = \frac{[1, \dots, 1]^T}{m} = \mathbf{c}. \quad (4.10)$$

□

#### 4.1.2 Centered Log-Ratio Transform

The Aitchison vector space has a few canonical representations. We describe one such representation system below.

**Definition 2.** *Centered log-ratio transform (CLRT)  $\phi : \Delta^n \rightarrow \mathbb{R}^m$  is defined by the relation*

*[?]:*

$$\phi(\mathbf{y}) := \underbrace{\begin{bmatrix} 1 - \frac{1}{m} & -\frac{1}{m} & -\frac{1}{m} & \dots & -\frac{1}{m} \\ -\frac{1}{m} & 1 - \frac{1}{m} & -\frac{1}{m} & \dots & -\frac{1}{m} \\ \vdots & & \ddots & & \vdots \\ -\frac{1}{m} & \dots & -\frac{1}{m} & 1 - \frac{1}{m} \end{bmatrix}}_{\mathbf{M}_{m \times m}} \begin{bmatrix} \log y_1 \\ \vdots \\ \log y_m \end{bmatrix} \quad (4.11)$$

Let  $\mathbf{Y} = \phi(\mathbf{y}) \in \mathbb{R}^m$ . Define  $\mathbf{u}_k \in \Delta^m$  as a set of basis vectors, where

$$\mathbf{u}_k = \frac{[1, \dots, 1, e, 1, \dots, 1]^\top}{m - 1 + e}, \quad (4.12)$$

where  $e$  appears on the  $k$ -th entry of  $\mathbf{u}_k$ . Then the inverse CLRT (ICLRT)  $\phi^{-1} : \mathbb{R}^m \rightarrow \Delta^m$  takes a canonical vector space representation:

$$\mathbf{y} = \phi^{-1}(\mathbf{Y}) := (Y_1 \otimes \mathbf{u}_1) \oplus \dots \oplus (Y_m \otimes \mathbf{u}_m). \quad (4.13)$$

**Corollary 3.** Inverse CLRT in (4.13) can be rewritten as

$$\phi^{-1}(\mathbf{Y}) := \frac{[e^{Y_1}, \dots, e^{Y_m}]^\top}{e^{Y_1} + \dots + e^{Y_m}}. \quad (4.14)$$

**Proposition 3** (Homomorphism). Centered log-ratio transform  $\phi$  has the following sense:

$$\phi((a \otimes \mathbf{y}) \oplus (b \otimes \mathbf{z})) = a\phi(\mathbf{y}) + b\phi(\mathbf{z}). \quad (4.15)$$

*Proof.* To see why this is the case, consider

$$(a \otimes \mathbf{y}) \oplus (b \otimes \mathbf{z}) = \frac{[y_1^a z_1^b, \dots, y_m^a z_m^b]}{y_1^a z_1^b + \dots + y_m^a z_m^b} \quad (4.16)$$

Rewriting  $\phi(\mathbf{y})$  as  $\phi(\mathbf{y}) = (\phi_1(\mathbf{y}), \dots, \phi_m(\mathbf{y}))$

$$\begin{aligned} & \phi_k((a \otimes \mathbf{y}) \oplus (b \otimes \mathbf{z})) \\ &= a \log(y_k) + b \log(z_k) \\ & \quad - a \frac{\log y_1 + \dots + \log y_m}{m} + b \frac{\log z_1 + \dots + \log z_m}{m} \\ &= a\phi_k(\mathbf{y}) + b\phi_k(\mathbf{z}), \end{aligned} \quad (4.17)$$

where the denominator in (4.16) term has conveniently canceled out in (4.17).  $\square$

**Proposition 4.** The unit simplex  $\Delta^m$  is isomorphic to  $\mathbb{R}^{m-1}$ .

*Proof.* The range space of CLRT  $\phi(\mathbf{y})$  is isomorphic to  $\mathbb{R}^{m-1}$  due to the fact that  $\mathbf{M} \in \mathbb{R}^{m \times m}$  is a rank  $m-1$  matrix. The nullspace of  $\mathbf{M} \in \mathbb{R}^{m \times m}$  is  $\{\mathbf{x} \in \mathbb{R}^m | x_1 = x_2 = \dots = x_m\}$ ,

which can only be reached by  $\mathbf{c}$  (i.e.  $\log y_1 = \dots = \log y_m$  if and only if  $\mathbf{y} = \mathbf{c}$ ). Thus by the first group isomorphism theorem and by Proposition 3, the Proposition is true [?].<sup>1</sup>  $\square$

**Proposition 5** (CLR coefficients). *If  $\mathbf{Y} = \phi(\mathbf{y})$ , then*

$$\sum_{k=1}^m \mathbf{Y}_k = 0. \quad (4.18)$$

*Proof.* This is a simple consequence of the fact every column of  $\mathbf{M}$  sums to zero.  $\square$

**Corollary 4** (Log-unit-simplex-Sum). *If  $\mathbf{y} = \phi^{-1}(\mathbf{Y}) \in \Delta^m$ , then*

$$\sum_{k=1}^m \log y_k \approx -m \cdot \max(Y_1, \dots, Y_m). \quad (4.19)$$

*Proof.* Recall (4.13) and (4.19), the sum of log-unit-simplex can be rewritten as:

$$\begin{aligned} \sum_{k=1}^m \log y_k &= \sum_{k=1}^m \log e^{Y_k} - m \cdot \log(e^{Y_1} + \dots + e^{Y_m}) \\ &= -m \cdot \log(e^{Y_1} + \dots + e^{Y_m}). \end{aligned} \quad (4.20)$$

Consider a set of real number  $v_1, \dots, v_m \in \mathbb{R}$ . The maximum function  $\max\{-v_1, \dots, -v_m\}$  can be rewritten as log-sum-exponential form [?]:

$$\max\{-v_1, \dots, -v_m\} = \lim_{\lambda \rightarrow +\infty} \frac{1}{\lambda} \log \left( \sum_{k=1}^m e^{(-v_k \lambda)} \right). \quad (4.21)$$

Assume  $v_k \lambda \approx Y_k$ ,

$$\log(e^{Y_1} + \dots + e^{Y_m}) \approx \max(Y_1, \dots, Y_m). \quad (4.22)$$

$\square$

---

<sup>1</sup>Alternatively,  $\Delta^m$  is isomorphic to  $\mathbb{R}^{m-1}$  by transivity since CRLT is invertible.

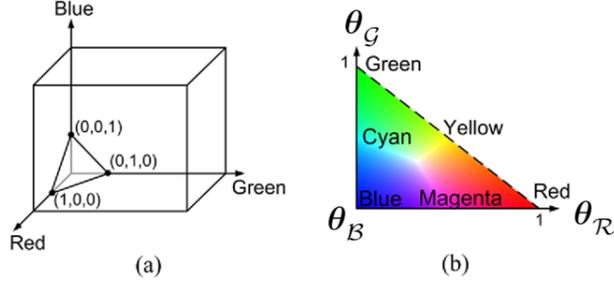


Figure 4.1: Demonstration of chromaticity coordinate. Equation (4.23) maps color channels  $\mathcal{R} \mathcal{G} \mathcal{B}$  onto the triangle plane in (a)—the chromaticity coordinate. By coloring the chromaticity coordinate triangle, (b) demonstrates the relation between color and color ratios .

### 4.1.3 Log-chromaticity Coordinate Representation of Color

Recall the definitions of clean pixel intensities and clean luminance in Section 2.2.2 and 2.2.3, denote by  $\theta = [\theta_{\mathcal{R}}^T, \theta_{\mathcal{G}}^T, \theta_{\mathcal{B}}^T]^T \in \mathbb{R}^{3N}$  the chromaticities of color intensities can be calculated by the Grassmann's law:  $\forall n \in [1, N]$ ,

$$\theta_{\mathcal{R}}(n) = \frac{\lambda_{\mathcal{R}}(n)}{\lambda_{\mathcal{L}}(n)}, \theta_{\mathcal{G}}(n) = \frac{\lambda_{\mathcal{G}}(n)}{\lambda_{\mathcal{L}}(n)}, \theta_{\mathcal{B}}(n) = \frac{\lambda_{\mathcal{B}}(n)}{\lambda_{\mathcal{L}}(n)}. \quad (4.23)$$

Figure 4.1 demonstrates that Equation (4.23) maps  $\lambda$  to the chromaticity coordinates. Obviously, for every pixel,  $\theta_{\mathcal{R}}(n) + \theta_{\mathcal{G}}(n) + \theta_{\mathcal{B}}(n) = 1$  and  $\theta_k(n) \geq 0$ . Thus the chromaticity vector  $\theta(n) = [\theta_{\mathcal{R}}(n), \theta_{\mathcal{G}}(n), \theta_{\mathcal{B}}(n)]^T \in \Delta^3$ . Define an unit simplex vector  $\mathbf{c} = [\frac{1}{3}, \frac{1}{3}, \frac{1}{3}]^T$ . And let  $\log \theta(n) = [\log \theta_{\mathcal{R}}(n), \log \theta_{\mathcal{G}}(n), \log \theta_{\mathcal{B}}(n)]^T$  denotes the log chromaticity vector. By (4.11), let  $[\mathbf{r}(n), \mathbf{g}(n), \mathbf{b}(n)]^T$  denotes the CLRT coefficients of  $\log \theta(n)$ , such that:

$$[\mathbf{r}(n), \mathbf{g}(n), \mathbf{b}(n)]^T = [\mathbf{M}_{3 \times 3}] \log \theta(n) \quad (4.24)$$

According to (4.19), we further expand  $\log \theta(n)$  as follow:

$$\begin{aligned} \log \theta(n) &= [\mathbf{M}_{3 \times 3}^T \quad 3\mathbf{c}^T] \begin{bmatrix} \mathbf{M}_{3 \times 3} \\ \mathbf{c} \end{bmatrix} \begin{bmatrix} \log \theta_{\mathcal{R}}(n) \\ \log \theta_{\mathcal{G}}(n) \\ \log \theta_{\mathcal{B}}(n) \end{bmatrix} \\ &\approx \begin{bmatrix} \frac{2}{3} & -\frac{1}{3} & -\frac{1}{3} & 1 \\ -\frac{1}{3} & \frac{2}{3} & -\frac{1}{3} & 1 \\ -\frac{1}{3} & -\frac{1}{3} & \frac{2}{3} & 1 \end{bmatrix} \begin{bmatrix} \mathbf{I}_{3 \times 3} \\ -\boldsymbol{\delta}(n) \end{bmatrix} \begin{bmatrix} \mathbf{r}(n) \\ \mathbf{g}(n) \\ \mathbf{b}(n) \end{bmatrix}, \end{aligned} \quad (4.25)$$

where  $\mathbf{I}_{3 \times 3} \in \mathbb{R}^{3 \times 3}$  is the identity matrix. And the row vector  $\boldsymbol{\delta}(n) = [\delta_{\mathbf{r}}(n), \delta_{\mathbf{g}}(n), \delta_{\mathbf{b}}(n)]$  is a selector to choose which color channel has the highest value. Specifically, for instance,

$$\boldsymbol{\delta}(n) = [1, 0, 0] \text{ if } \lambda_{\mathcal{R}}(n) \geq \lambda_{\mathcal{G}}(n) \ \& \ \lambda_{\mathcal{R}}(n) \geq \lambda_{\mathcal{B}}(n) \quad (4.26)$$

*Proof.*

$$\begin{bmatrix} \frac{2}{3} & -\frac{1}{3} & -\frac{1}{3} & 1 \\ -\frac{1}{3} & \frac{2}{3} & -\frac{1}{3} & 1 \\ -\frac{1}{3} & -\frac{1}{3} & \frac{2}{3} & 1 \end{bmatrix} \begin{bmatrix} \frac{2}{3} & -\frac{1}{3} & -\frac{1}{3} \\ -\frac{1}{3} & \frac{2}{3} & -\frac{1}{3} \\ -\frac{1}{3} & -\frac{1}{3} & \frac{2}{3} \\ \frac{1}{3} & \frac{1}{3} & \frac{1}{3} \end{bmatrix} = [\mathbf{I}_{3 \times 3}]. \quad (4.27)$$

□

Recall (4.11), (4.18), and (4.19) and ,

$$\begin{bmatrix} \frac{2}{3} & -\frac{1}{3} & -\frac{1}{3} \\ -\frac{1}{3} & \frac{2}{3} & -\frac{1}{3} \\ -\frac{1}{3} & -\frac{1}{3} & \frac{2}{3} \\ \frac{1}{3} & \frac{1}{3} & \frac{1}{3} \end{bmatrix} \begin{bmatrix} \log \theta_{\mathcal{R}}(n) \\ \log \theta_{\mathcal{G}}(n) \\ \log \theta_{\mathcal{B}}(n) \end{bmatrix} \approx \begin{bmatrix} \mathbf{I}_{3 \times 3} \\ \boldsymbol{\delta}(n) \end{bmatrix} \begin{bmatrix} \mathbf{r}(n) \\ \mathbf{g}(n) \\ \mathbf{b}(n) \end{bmatrix}. \quad (4.28)$$

#### 4.1.4 Time-Frequency Analysis Of CLRT Simplex coefficients

The analysis in Section 4.1.3 explicitly described the log-chromaticity coordinate representation of single color pixel. Image denoising techniques normally gain strength from multiple pixels. In addition, in signal processing, many time-frequency analysis techniques

have been developed to address non-stationary random process—short-time Fourier transform, filter banks, frame transform, and discrete wavelet transforms are examples of popular linear transformations developed for such purpose. We would like to extend these techniques to *CLR coefficients*. Below, we provide a toy example first to develop the basic idea, before presenting the full Haar-wavelet analysis in Section 4.2.

Define discrete Haar wavelet transform (DWT) as

$$\mathbf{T}_W = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}. \quad (4.29)$$

And let  $\mathbb{S} = [\mathbf{S}_r^\top, \mathbf{S}_g^\top, \mathbf{S}_b^\top]^\top$  and  $\mathbb{W} = [\mathbf{W}_r^\top, \mathbf{W}_g^\top, \mathbf{W}_b^\top]^\top$  denote the scaling and wavelet CLR coefficients, respectively. The Haar wavelet transform of CLR coefficients can be described as:

$$\begin{bmatrix} S_r(1) \\ W_r(1) \\ S_g(1) \\ W_g(1) \\ S_b(1) \\ W_b(1) \end{bmatrix} := \underbrace{\begin{bmatrix} \mathbf{T}_W & & \\ & \mathbf{T}_W & \\ & & \mathbf{T}_W \end{bmatrix}}_{6 \times 6} \begin{bmatrix} \mathbf{r}(1) \\ \mathbf{r}(2) \\ \mathbf{g}(1) \\ \mathbf{g}(2) \\ \mathbf{b}(1) \\ \mathbf{b}(2) \end{bmatrix}, \quad (4.30)$$

where the elements in blank areas are zeros. Recall Proposition 3, the CLR coefficients of chromaticities are isomorphic to  $\mathbb{R}^2$ . In particular,  $S_r(1) + S_g(1) + S_b(1) = 0$ ,  $W_r(1) + W_g(1) + W_b(1) = 0$ .

## 4.2 Multivariate Inhomogeneous Poisson Sequence and Bayes Estimate in Chromaticity Coordinates

### 4.2.1 Poisson-Multinomial Likelihood Function and Minimum Mean Square Error Estimator in Chromaticity Coordinates

Recall that  $k \in \{\mathcal{R}, \mathcal{G}, \mathcal{B}\}$  be the color *channel* index. Let  $\mathbb{F}|\mathbb{X}$  as defined in Section 2.1, in multivariate in-homogeneous Poisson sequence problems, every noisy measurement of  $\mathbb{X}$  is

an independent Poisson corrupted pixel as described in (2.19). We now turn our attention to a multivariate random sequence.

For any single color pixel  $\mathbb{F}(n)|\lambda(n)$ ,  $\forall n \in [1, N]$ , the multivariate Poisson probability is described as

$$Pr[\mathbb{F}(n)|\lambda(n)] = \prod_{k \in [\mathcal{R}, \mathcal{G}, \mathcal{B}]} \frac{e^{-\lambda_k(n)} \lambda_k(n)^{F_k(n)}}{F_k(n)!}. \quad (4.31)$$

Recall (2.21) and consider rewriting the Poisson probability as a product of conditional probabilities:

$$\begin{aligned} Pr[\mathbb{F}(n)|\lambda(n)] &= Pr[F_{\mathcal{R}}(n), F_{\mathcal{G}}(n), F_{\mathcal{L}}(n)|\lambda(n)] \\ &= Pr[F_{\mathcal{R}}(n), F_{\mathcal{G}}(n)|F_{\mathcal{L}}(n), \lambda(n)] Pr[F_{\mathcal{L}}(n)|\lambda(n)]. \end{aligned} \quad (4.32)$$

Therefore, given that chromaticity is defined as in (4.23), the conditional probability  $\mathbb{F}(n)|F_{\mathcal{L}}(n), \theta(n)$  has the distribution of the form:

$$\begin{aligned} &Pr[\mathbb{F}(n)|F_{\mathcal{L}}(n), \theta(n)] \\ &= \frac{F_{\mathcal{L}}(n)!}{F_{\mathcal{R}}(n)! F_{\mathcal{G}}(n)! F_{\mathcal{B}}(n)!} \prod_{k \in [\mathcal{R}, \mathcal{G}, \mathcal{B}]} \theta_k(n)^{F_k(n)} \end{aligned} \quad (4.33)$$

*Proof.* Rewrite (4.32),

$$\begin{aligned} &Pr[F_{\mathcal{R}}(n), F_{\mathcal{G}}(n)|F_{\mathcal{L}}(n), \lambda(n)] Pr[F_{\mathcal{L}}(n)|\lambda(n)], \\ &= Pr[F_{\mathcal{R}}(n), F_{\mathcal{G}}(n), F_{\mathcal{L}}(n) - F_{\mathcal{R}}(n) - F_{\mathcal{G}}(n)|F_{\mathcal{L}}(n), \lambda(n)] \\ &\quad \cdot Pr[F_{\mathcal{L}}(n)|\lambda(n)], \\ &= Pr[\mathbb{F}(n)|F_{\mathcal{L}}(n), \lambda(n)] Pr[F_{\mathcal{L}}(n)|\lambda(n)]. \end{aligned} \quad (4.34)$$

Therefore,

$$Pr[\mathbb{F}(n)|F_{\mathcal{L}}(n), \lambda(n)] = \frac{Pr[\mathbb{F}(n)|\lambda(n)]}{Pr[F_{\mathcal{L}}(n)|\lambda(n)]} \quad (4.35)$$

Substituting (2.23), (4.31), and (4.23) we proof (4.33).  $\square$

$$\begin{aligned}
& \sum_{k \in \{\mathcal{R}, \mathcal{G}, \mathcal{B}\}} F_k(n) \log \theta_k(n) \\
&= [F_{\mathcal{R}}(n) \quad F_{\mathcal{G}}(n) \quad F_{\mathcal{B}}(n)] \begin{bmatrix} \frac{2}{3} & -\frac{1}{3} & -\frac{1}{3} & 1 \\ -\frac{1}{3} & \frac{2}{3} & -\frac{1}{3} & 1 \\ -\frac{1}{3} & -\frac{1}{3} & \frac{2}{3} & 1 \end{bmatrix} \begin{bmatrix} \frac{2}{3} & -\frac{1}{3} & -\frac{1}{3} \\ -\frac{1}{3} & \frac{2}{3} & -\frac{1}{3} \\ -\frac{1}{3} & -\frac{1}{3} & \frac{2}{3} \\ \frac{1}{3} & \frac{1}{3} & \frac{1}{3} \end{bmatrix} \begin{bmatrix} \log \theta_{\mathcal{R}}(n) \\ \log \theta_{\mathcal{G}}(n) \\ \log \theta_{\mathcal{B}}(n) \end{bmatrix}, \\
&\approx [F_{\mathcal{R}}(n) \quad F_{\mathcal{G}}(n) \quad F_{\mathcal{B}}(n)] \begin{bmatrix} \frac{2}{3} & -\frac{1}{3} & -\frac{1}{3} & 1 \\ -\frac{1}{3} & \frac{2}{3} & -\frac{1}{3} & 1 \\ -\frac{1}{3} & -\frac{1}{3} & \frac{2}{3} & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ -\delta_{\mathbf{r}}(n) & -\delta_{\mathbf{g}}(n) & -\delta_{\mathbf{b}}(n) \end{bmatrix} \begin{bmatrix} \mathbf{r}(n) \\ \mathbf{g}(n) \\ \mathbf{b}(n) \end{bmatrix}, \\
&= [F_{\mathcal{R}}(n) - (\frac{1}{3} + \delta_{\mathbf{r}}(n))F_{\mathcal{L}}(n) \quad F_{\mathcal{G}}(n) - (\frac{1}{3} + \delta_{\mathbf{g}}(n))F_{\mathcal{L}}(n) \quad F_{\mathcal{B}}(n) - (\frac{1}{3} + \delta_{\mathbf{b}}(n))F_{\mathcal{L}}(n)] \begin{bmatrix} \mathbf{r}(n) \\ \mathbf{g}(n) \\ \mathbf{b}(n) \end{bmatrix}. \tag{4.38}
\end{aligned}$$

$$\begin{aligned}
& \prod_{n=1}^N \prod_k \theta_k(n)^{F_k(n)} = \exp \left\{ \begin{bmatrix} \mathbf{F}_{\mathcal{R}}^{\top} & \mathbf{F}_{\mathcal{G}}^{\top} & \mathbf{F}_{\mathcal{B}}^{\top} \\ \log \theta_{\mathcal{R}} \\ \log \theta_{\mathcal{G}} \\ \log \theta_{\mathcal{B}} \end{bmatrix} \right\} \\
&\approx \exp \left\{ \begin{bmatrix} \mathbf{F}_{\mathcal{R}}^{\top} - (\mathcal{M}_{\mathcal{R}}^{\delta} \circ \mathbf{F}_{\mathcal{L}})^{\top} & \mathbf{F}_{\mathcal{R}}^{\top} - (\mathcal{M}_{\mathcal{G}}^{\delta} \circ \mathbf{F}_{\mathcal{L}})^{\top} & \mathbf{F}_{\mathcal{R}}^{\top} - (\mathcal{M}_{\mathcal{B}}^{\delta} \circ \mathbf{F}_{\mathcal{L}})^{\top} \\ \mathbf{r} \\ \mathbf{g} \\ \mathbf{b} \end{bmatrix} \right\}, \tag{4.40} \\
&= \exp \left\{ \underbrace{\begin{bmatrix} \mathbf{F}_{\mathcal{R}}^{\top} - (\mathcal{M}_{\mathcal{R}}^{\delta} \circ \mathbf{F}_{\mathcal{L}})^{\top} & \mathbf{F}_{\mathcal{R}}^{\top} - (\mathcal{M}_{\mathcal{G}}^{\delta} \circ \mathbf{F}_{\mathcal{L}})^{\top} & \mathbf{F}_{\mathcal{R}}^{\top} - (\mathcal{M}_{\mathcal{B}}^{\delta} \circ \mathbf{F}_{\mathcal{L}})^{\top} \end{bmatrix}}_{\text{DWT of noisy image}} \underbrace{\begin{bmatrix} T_W^{\top} & & \\ & T_W^{\top} & \\ & & \ddots \\ & & & T_W^{\top} \end{bmatrix}}_{3N \times 3N} \underbrace{\begin{bmatrix} T_W & & \\ & T_W & \\ & & \ddots \\ & & & T_W \end{bmatrix}}_{3N \times 3N} \begin{bmatrix} \mathbf{r} \\ \mathbf{g} \\ \mathbf{b} \end{bmatrix}}_{\text{DWT of clean CLR coef.}} \right\}, \tag{4.41}
\end{aligned}$$

we focus the influence of  $\mathcal{M}_k^{\delta}$  on the wavelet coefficients for denoising purpose,

$$\approx \exp \left\{ \begin{bmatrix} t'_{\mathcal{R}}{}^{\top} & t'_{\mathcal{G}}{}^{\top} & t'_{\mathcal{B}}{}^{\top} & (\mathbf{x}_{\mathcal{R}} - \mathcal{M}_{\mathcal{R}}^{\delta} \circ \mathbf{x}_{\mathcal{L}})^{\top} & (\mathbf{x}_{\mathcal{R}} - \mathcal{M}_{\mathcal{G}}^{\delta} \circ \mathbf{x}_{\mathcal{L}})^{\top} & (\mathbf{x}_{\mathcal{R}} - \mathcal{M}_{\mathcal{B}}^{\delta} \circ \mathbf{x}_{\mathcal{L}})^{\top} \\ S_{\mathbf{r}} \\ S_{\mathbf{g}} \\ W_{\mathbf{r}} \\ W_{\mathbf{g}} \\ W_{\mathbf{b}} \end{bmatrix} \right\}. \tag{4.42}$$

The traditional way of using minimum mean square error (MMSE) to reconstruct clean chromaticities  $\widehat{\Theta}$  is a posterior mean  $\mathbb{E}[\Theta|\mathbf{F}_{\mathcal{L}}, \mathbb{F}]$  of the form:

$$\begin{aligned}\widehat{\Theta} &= \int \Theta Pr[\Theta|\mathbf{F}_{\mathcal{L}}, \mathbb{F}] d\Theta \\ &= \frac{\int \Theta \prod_{n=1}^N Pr[\mathbb{F}(n)|F_{\mathcal{L}}(n), \Theta(n)] Pr[\Theta(n)] d\Theta}{\int \prod_{n=1}^N Pr[\mathbb{F}(n)|F_{\mathcal{L}}(n), \Theta(n)] Pr[\Theta(n)] d\Theta},\end{aligned}\tag{4.36}$$

where the probability of the clean chromaticity  $Pr[\Theta(n)]$  indicates the prior distribution.

There are two majority challenges in using (4.36) directly. First, if we substitute (4.33) in (4.36), it yields a multidimensional product<sup>2</sup>:

$$\prod_{n=1}^N \prod_k \theta_k(n)^{F_k(n)},\tag{4.37}$$

which is hard and impractical to deal with. Second,  $Pr[\Theta(n)]$  is a density in three mutually dependent dimensions.

#### 4.2.2 Linearized Poisson-Multinomial Likelihood Function in Wavelet Domain

We firstly focus on linearizing and simplifying the likelihood function. Rewrite the multiplication product as  $\prod_k \theta_k(n)^{F_k(n)} = \exp\{\sum_k F_k(n) \log \theta_k(n)\}$ , by substituting (4.24) — (4.27), we expand the above exponent as (4.38). Now we define a variable vector  $\mathcal{M}_k^\delta = [\dots, \mathcal{M}_k^\delta(n), \dots]^\top$  such that:

$$\mathcal{M}_k^\delta(n) = \begin{cases} \frac{4}{3}, & \text{if } \lambda_k(n) = \max\{\lambda_{\mathcal{R}}(n), \lambda_{\mathcal{G}}(n), \lambda_{\mathcal{B}}(n)\} \\ \frac{1}{3}, & \text{otherwise} \end{cases}\tag{4.39}$$

As a result, we can rewrite the product (4.37) in terms of exponential summation and log-chromaticities or CLR coefficients as illustrated in (4.40), where the “ $\circ$ ” denotes the Hadamard product. Note that (4.29) is equivalent to (2.11). And  $\mathbf{T}_W$  is an orthornormal matrix. Using (4.29) — (4.48), we implement Haar-wavelet analysis of (4.37) in CLRT space as illustrated in (4.41) and (4.42). For convenience, we denote  $\mathbf{X}'_k = \mathbf{X}_k - \mathcal{M}_k^\delta \circ \mathbf{X}_{\mathcal{L}}$ .

<sup>2</sup>For convenience,  $\prod_{k \in [\mathcal{R}, \mathcal{G}, \mathcal{B}]}$ ,  $\sum_{k \in [\mathcal{R}, \mathcal{G}, \mathcal{B}]}$  will be denoted as  $\prod_k$ ,  $\sum_k$ .

Equation (4.41) and (4.42) are absolutely equal if the dominated color channel of underlying pixels has no change. By contrast, this relation becomes weaker at each color edge. We admit that there exists trade-off between using wavelet analysis and maintaining the intact of (4.40). However, in our real application, The mismatching of  $\mathbf{X}'_k$  and  $\boldsymbol{\lambda}_k$  does not cause color invariance at the color edges. We will verify that the color edges will not be obviously influenced using real sensor data in Section 5.2.3. The significance of (4.42) is that it illustrates the connection of clean log-chromaticities and noisy pixels in Haar wavelet space, enabling a way to use wavelet based denoising in chromaticity coordinates. For wavelet based denoising, clean scaling coefficients are technically unavailable. In practice, they are replaced by corresponding noisy scaling coefficient. Specifically, denote  $\widehat{\mathbb{S}} = [\widehat{\mathbf{S}}_r^T, \widehat{\mathbf{S}}_g^T, \widehat{\mathbf{S}}_b^T]^T$  and  $\widetilde{\mathbb{W}} = [\widetilde{\mathbf{W}}_r^T, \widetilde{\mathbf{W}}_g^T, \widetilde{\mathbf{W}}_b^T]^T$  the corresponding noisy scaling and wavelet coefficients which are calculated using (4.24) and (4.30), and meanwhile, replacing  $\theta_k(n)$  by  $\frac{F_k(n)}{F_{\mathcal{L}}(n)}, \forall n, k$ . We set  $\mathbb{S} = \widehat{\mathbb{S}}$ . Then the following parts of this work will focus on the estimation of wavelet CLR coefficients  $\widehat{\mathbb{W}} = [\widehat{\mathbf{W}}_r^T, \widehat{\mathbf{W}}_g^T, \widehat{\mathbf{W}}_b^T]^T$ .

**Proposition 6** (Isomorphic Transferred likelihood function of wavelet CLR Coefficients).  
Let  $\mathbb{F}|\boldsymbol{\lambda}, \mathbf{F}_{\mathcal{L}}, \boldsymbol{\lambda}_{\mathcal{L}}$  being defined as in Section 2.1, and  $\Theta$  being defined as in 4.1.3, define an image dependent constant:

$$\mathcal{C}(\mathbb{F}) = \prod_{n=1}^N \frac{F_{\mathcal{L}}(n)!}{F_{\mathcal{R}}(n)!F_{\mathcal{G}}(n)!F_{\mathcal{B}}(n)!}, \quad (4.43)$$

the original likelihood function of independent Poisson color pixels has the form:

$$\begin{aligned} Pr[\mathbb{F}|\mathbf{F}_{\mathcal{L}}, \Theta] &= \prod_{n=1}^N Pr[\mathbb{F}(n)|F_{\mathcal{L}}, \Theta(n)] \\ &= \mathcal{C}(\mathbb{F}) \times \prod_{n=1}^N \prod_k \theta_k(n)^{F_k(n)}. \end{aligned} \quad (4.44)$$

Let  $\mathbb{F} \mapsto (\mathfrak{t}, \mathbb{X})$  being defined as in Section 2.1. Let  $\mathbb{S}$  and  $\mathbb{W}$  being defined as in 4.1.4, and  $\mathbb{X}'$  being denoted as in 4.2.2. Assume that  $\widehat{\mathbb{S}} \approx \mathbb{S}$ , we define a new likelihood function:

$$\begin{aligned} Pr[\mathbb{X}|\mathbf{F}_{\mathcal{L}}, \mathbb{W}] &= \prod_{n=1}^{N/2} Pr[\mathbb{X}(n)|\mathbf{F}_{\mathcal{L}}, \mathbb{W}(n)] \\ &= e^{f(\mathfrak{t}, \widehat{\mathbb{S}})} \cdot \mathcal{C}(\mathbb{F}) \times \exp \left\{ \begin{bmatrix} \mathbf{X}'_{\mathcal{R}}{}^\top & \mathbf{X}'_{\mathcal{G}}{}^\top & \mathbf{X}'_{\mathcal{B}}{}^\top \end{bmatrix} \begin{bmatrix} \mathbf{W}_{\mathfrak{r}} \\ \mathbf{W}_{\mathfrak{g}} \\ \mathbf{W}_{\mathfrak{b}} \end{bmatrix} \right\}, \end{aligned} \quad (4.45)$$

where the  $f(\mathfrak{t}, \widehat{\mathbb{S}})$  is another constant that depending on  $\mathfrak{t}$  and  $\widehat{\mathbb{S}}$ . The form of  $f(\mathfrak{t}, \widehat{\mathbb{S}})$  follows the exponential summation of scaling coefficients that partially described in (4.42). We omit such analysis in this paper because we assume it can be approximately accessed. According to the described in (4.42),

$$Pr[\mathbb{X}|\mathbf{F}_{\mathcal{L}}, \mathbb{W}] \approx Pr[\mathbb{F}|\mathbf{F}_{\mathcal{L}}, \Theta] \quad (4.46)$$

Consider the orthonormal rotation matrix:

$$\mathbf{T}_{\mathcal{C}} = \begin{bmatrix} \frac{1}{\sqrt{3}} & \frac{1}{\sqrt{3}} & \frac{1}{\sqrt{3}} \\ \frac{1}{\sqrt{6}} & \frac{-2}{\sqrt{6}} & \frac{1}{\sqrt{6}} \\ \frac{1}{\sqrt{2}} & 0 & \frac{-1}{\sqrt{2}} \end{bmatrix}, \quad (4.47)$$

we denote the isomorphic color transform of wavelet coefficients as follow:

$$\begin{bmatrix} \mathbf{A}_0^\top \\ \mathbf{A}_1^\top \\ \mathbf{A}_2^\top \end{bmatrix} = \mathbf{T}_{\mathcal{C}} \begin{bmatrix} \mathbf{X}'_{\mathcal{R}}{}^\top \\ \mathbf{X}'_{\mathcal{G}}{}^\top \\ \mathbf{X}'_{\mathcal{B}}{}^\top \end{bmatrix}, \quad \begin{bmatrix} \mathbf{B}_0^\top \\ \mathbf{B}_1^\top \\ \mathbf{B}_2^\top \end{bmatrix} = \mathbf{T}_{\mathcal{C}} \begin{bmatrix} \mathbf{W}_{\mathfrak{r}}^\top \\ \mathbf{W}_{\mathfrak{g}}^\top \\ \mathbf{W}_{\mathfrak{b}}^\top \end{bmatrix}, \quad (4.48)$$

where  $B_0(n) = 0 \ \forall n \in [1, \frac{N}{2}]$ . For future usage, we let  $\widetilde{\mathbb{B}}$  denotes the isomorphic color transformed result of  $\widetilde{\mathbb{W}}$ . Finally, we transfer (4.45), and define another likelihood function to describe the wavelet CLR coefficients such that:

$$\begin{aligned} Pr[\mathbb{A}|\mathbf{F}_{\mathcal{L}}, \mathbb{B}] &= \prod_{n=1}^{N/2} Pr[\mathbb{A}(n)|\mathbf{F}_{\mathcal{L}}, \mathbb{B}(n)] \\ &= e^{f(\mathfrak{t}, \widehat{\mathbb{S}})} \cdot \mathcal{C}(\mathbb{F}) \times \exp \left\{ \begin{bmatrix} \mathbf{A}_1^\top & \mathbf{A}_2^\top \end{bmatrix} \begin{bmatrix} \mathbf{B}_1 \\ \mathbf{B}_2 \end{bmatrix} \right\}. \end{aligned} \quad (4.49)$$

Note that  $\forall n_1 \neq n_2$ ,  $Pr[\mathbb{A}(n_1)|\mathbf{F}_{\mathcal{L}}, \mathbb{B}(n_1)]$  is independent to  $Pr[\mathbb{A}(n_2)|\mathbf{F}_{\mathcal{L}}, \mathbb{B}(n_2)]$ . With the fact that  $Pr[\mathbb{A}|\mathbf{F}_{\mathcal{L}}, \mathbb{B}] = Pr[\mathbb{X}|\mathbf{F}_{\mathcal{L}}, \mathbb{W}]$ , we say  $Pr[\mathbb{A}|\mathbf{F}_{\mathcal{L}}, \mathbb{B}]$  is isomorphic to  $Pr[\mathbb{F}|\mathbf{F}_{\mathcal{L}}, \Theta]$

*Proof.*

$$\begin{aligned} & \begin{bmatrix} \mathbf{X}'_{\mathcal{R}} & \mathbf{X}'_{\mathcal{G}} & \mathbf{X}'_{\mathcal{B}} \end{bmatrix} \mathbf{T}_c^{\top} \mathbf{T}_c \begin{bmatrix} \mathbf{W}_{\mathbf{r}}^{\top} \\ \mathbf{W}_{\mathbf{g}}^{\top} \\ \mathbf{W}_{\mathbf{b}}^{\top} \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{A}_0 & \mathbf{A}_1 & \mathbf{A}_2 \end{bmatrix} \begin{bmatrix} \mathbf{B}_0^{\top} \\ \mathbf{B}_1^{\top} \\ \mathbf{B}_2^{\top} \end{bmatrix} \end{aligned} \quad (4.50)$$

□

### 4.2.3 Practical MMSE Denoising in Chromaticity Coordinates

**Proposition 7** (Framework of color image denoising in log-chromaticity coordinates). *With the definition of (4.49), we propose a practical color denoising strategy in chromaticity coordinates as follow: 1), estimate the clean coefficients  $\widehat{\mathbb{B}}(\cdot)$  using MMSE estimator method from  $\mathbb{A}$ . Here we emphasize again that  $\mathbb{B}$  is the transformed coefficient of  $\log \theta$ ; 2), calculate  $\widehat{\mathbb{S}}$  as described in 4.2.2. And inverse the orthornormal color transform and wavelet transform which are defined in (4.47) and (4.29), respectively, to obtain  $[\widehat{\mathbf{r}} \ \widehat{\mathbf{g}} \ \widehat{\mathbf{b}}]^{\top}$ ; 3), apply ICRLT using (4.13) to obtain  $\widehat{\Theta}$ . Now we propose the reconstructed clean coefficients  $\widehat{\mathbb{B}}(\mathbb{A})$  using posterior mean. 4), reconstruct  $\widehat{\lambda}_{\mathcal{L}}$  using Poisson image denoising technique, then reconstruct  $[\widehat{\lambda}_{\mathcal{R}}^{\top}, \widehat{\lambda}_{\mathcal{G}}^{\top}, \widehat{\lambda}_{\mathcal{B}}^{\top}] \in \mathbb{R}^{3N}$  by:*

$$\begin{aligned} \widehat{\lambda}_{\mathcal{R}}(n) &= \widehat{\lambda}_{\mathcal{L}}(n) \cdot \widehat{\theta}_{\mathcal{R}}(n) \\ \widehat{\lambda}_{\mathcal{G}}(n) &= \widehat{\lambda}_{\mathcal{L}}(n) \cdot \widehat{\theta}_{\mathcal{G}}(n) \quad \forall n \in \mathbb{R}^N \\ \widehat{\lambda}_{\mathcal{B}}(n) &= \widehat{\lambda}_{\mathcal{L}}(n) \cdot \widehat{\theta}_{\mathcal{B}}(n). \end{aligned} \quad (4.51)$$

*Details of step 1) will be explained in the following sections.*

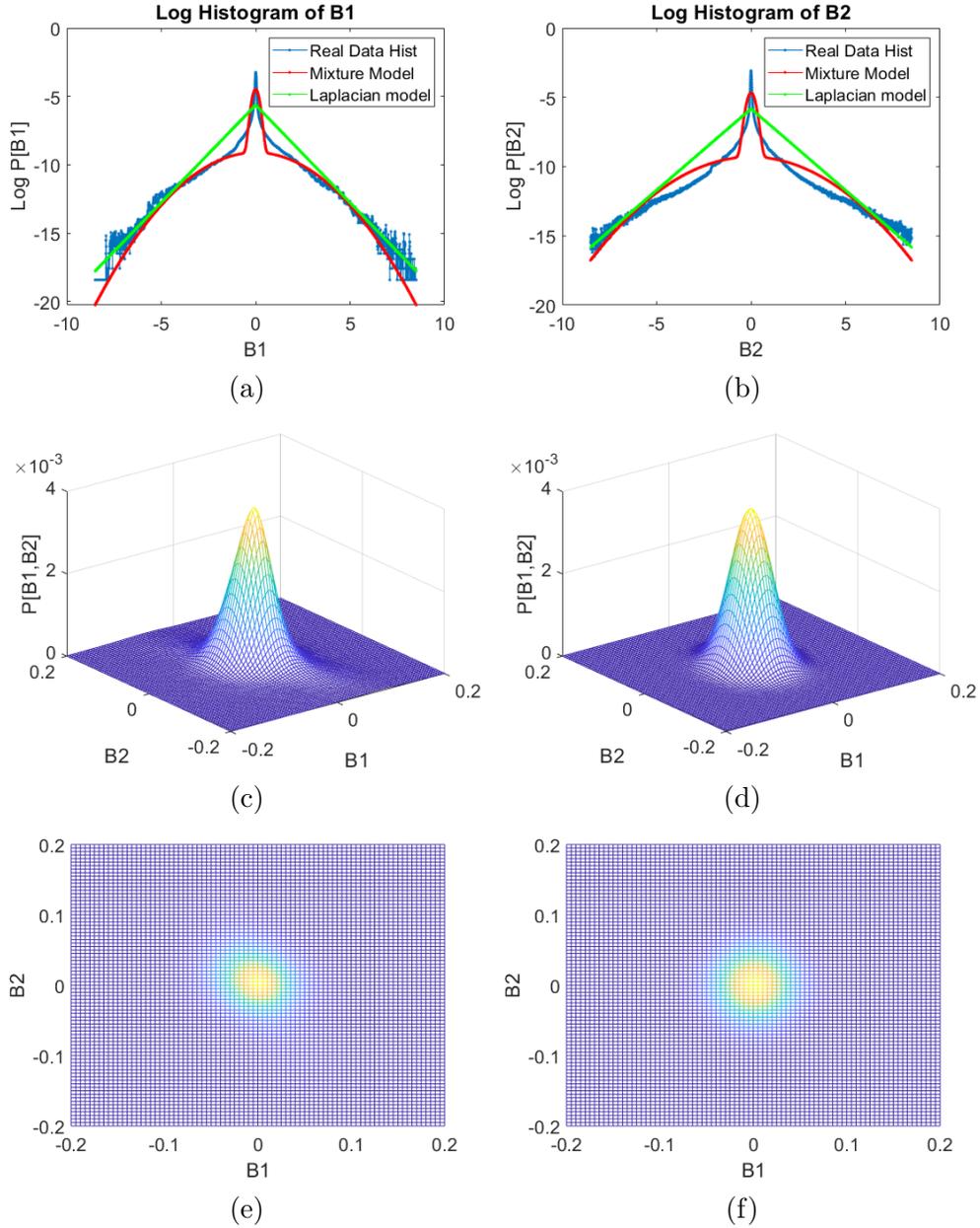


Figure 4.2: Analysis of prior density  $Pr[\mathbb{B}]$  using long exposure image as the proxy of clean reference. (a,b), Log-histogram (blue plots) of the marginal prior density  $Pr[\mathbf{B}_1]$  and  $Pr[\mathbf{B}_2]$ , respectively. The red-line plots illustrate the approximation of the marginal prior with mixture-Gaussian model. The green-line plots illustrate the approximation with Laplacian model. (c), 2D-histogram of  $Pr[\mathbb{B}]$ . (d), Approximation of  $Pr[\mathbb{B}]$  using 2-D mixture-Gaussian model as described in (4.55). (e), the top-down view of (c). We can see  $\mathbf{B}_1$  and  $\mathbf{B}_2$  are less correlated. (f), the approximation of (e) using 2-D mixture-Gaussian model. In this model,  $\mathbf{B}_1$  and  $\mathbf{B}_2$  are dependent but uncorrelated.

Recalling (4.36), due to the independency of (4.49), the clean coefficient  $\widehat{\mathbb{B}}(\mathbb{A}(n))$  can be reconstructed using posterior mean  $\mathbb{E}[\mathbb{B}(n)|\mathbf{F}_{\mathcal{L}}, \mathbb{A}(n)]$  of the form:

$$\begin{aligned} \widehat{\mathbb{B}}(\mathbb{A}(n)) &= \int \mathbb{B}(n) Pr[\mathbb{B}(n)|\mathbf{F}_{\mathcal{L}}, \mathbb{A}(n)] \mathfrak{d}\mathbb{B}(n) \\ &= \frac{\int \mathbb{B}(n) \exp\{A_1(n)B_1(n) + A_2(n)B_2(n)\} Pr[\mathbb{B}(n)] \mathfrak{d}\mathbb{B}(n)}{\int \exp\{A_1(n)B_1(n) + A_2(n)B_2(n)\} Pr[\mathbb{B}(n)] \mathfrak{d}\mathbb{B}(n)}. \end{aligned} \quad (4.52)$$

Note that all the constant terms in (4.49) are canceled eventually. Compare with the MMSE in original chromaticity coordinates, three dimensional prior density  $Pr[\mathbb{B}(n)]$  is mapped to a restrict two dimensional form  $Pr[\mathbb{B}(n)]$ . Thereby, we can visualize the 2D prior density with color image data. As illustrated in Figure 4.2, we study the density  $Pr[\mathbb{B}]$  using real sensor data with longer exposure as a proxy of clean image. First of all, from (a,b), we notice that the marginal prior densities  $\mathbf{B}_1$  and  $\mathbf{B}_2$ , especially the heavy tails, can be approximated by (but not limited to) mixture-Gaussian model of the form  $Pr[B_k(n)]$ :

$$\begin{aligned} &= \rho Pr[B_k(n)|Z = z_0] + (1 - \rho) Pr[B_k(n)|Z = z_1] \\ &= \rho \frac{1}{\sqrt{2\pi}\sigma_{0k}} e^{-\left(\frac{B_k^2(n)}{2\sigma_{0k}^2}\right)} + (1 - \rho) \frac{1}{\sqrt{2\pi}\sigma_{1k}} e^{-\left(\frac{B_k^2(n)}{2\sigma_{1k}^2}\right)}, \end{aligned} \quad (4.53)$$

where  $k = 1$  or  $2$ , and  $Z$  is a latent hidden variable specifying the identity of the mixture components. ( $\{B_k(n)|Z = z_0\} \sim \mathcal{N}(0, \sigma_{0k}^2)$  represents the edge/texture information and  $\{B_k(n)|Z = z_1\} \sim \mathcal{N}(0, \sigma_{1k}^2)$  represents the smoother/flatter areas, in particular.) The prior probability  $\rho = Pr[Z = z_0]$  indicates the proportion of  $\{Z = z_0\}$  vs  $\{Z = z_1\}$ .

The mixture Gaussian model is good, but not enough to describe the heavy tail phenomenon in log-scale. So, as demonstrated in (4.54), we tried to use Laplacian model to overcome the heavier-tail issue remained in mixture-Gaussian model. The Laplacian model can be described as:

$$Pr[B_k(n)] = \frac{1}{2b_k} e^{-\frac{|B_k(n)|}{b_k}}, \quad (4.54)$$

where the positive constant  $b_k$  is the Laplacian distribution parameter. The Figure 4.2 (a,b) show the potential of the accurate approximation using complicate mixture model.

However, due to the computational complexity, we keep using mixture of two Gaussian's in this paper.

Furthermore, in the original chromaticiy coordinates,  $\boldsymbol{\theta}_{\mathcal{R}}$ ,  $\boldsymbol{\theta}_{\mathcal{G}}$ , and  $\boldsymbol{\theta}_{\mathcal{B}}$  are highly correlated. From Figure 4.2 (e), we find out a fact that  $\mathbf{B}_1$  and  $\mathbf{B}_2$  have limited correlation. We verified above findings with several different scene contents, as well as images captured under illuminations in different color temperatures. Overall, we use the 2D mixture-Gaussian function to approximated the prior density, which is described as  $Pr[\mathbb{B}(n)] :=$

$$\begin{aligned} & \rho Pr[\mathbb{B}(n)|Z = z_0] + (1 - \rho) Pr[\mathbb{B}(n)|Z = z_1] \\ & = \rho \frac{\exp\left\{-\frac{\mathbb{B}^\top(n)\Sigma_0^{-1}\mathbb{B}(n)}{2}\right\}}{2\pi\sqrt{|\Sigma_0|}} + (1 - \rho) \frac{\exp\left\{-\frac{\mathbb{B}^\top(n)\Sigma_1^{-1}\mathbb{B}(n)}{2}\right\}}{2\pi\sqrt{|\Sigma_1|}}, \end{aligned} \quad (4.55)$$

where  $\Sigma_m = \begin{bmatrix} \sigma_{m1}^2 & \\ & \sigma_{m2}^2 \end{bmatrix} \forall m = 0 \text{ or } 1$ . Such model has the marginal density function that is exactly same as (4.53), has the  $B_1(n)$  and  $B_2(n)$  are dependent but uncorrelated random variables. The latent variable  $Z$  in (4.53) and (4.55) plays the role of “edge detector”. It is correct that both  $\mathbf{B}_1$  and  $\mathbf{B}_2$  share the similar edge map, but their color variations are spatially different. The approximation of prior density is demonstrated as the comparison between (c,e) and (d,f).

To implement MMSE in terms of mixture prior, we further extend the posterior density in terms of the hidden variable  $Z$ . By total probability,  $Pr[\mathbb{B}(n)|\mathbf{F}_{\mathcal{L}}, \mathbb{A}(n)] :=$

$$\begin{aligned} & \int Pr[\mathbb{B}(n)|\mathbf{F}_{\mathcal{L}}, \mathbb{A}(n), Z] Pr[Z|\mathbf{F}_{\mathcal{L}}, \mathbb{A}(n)] dZ \\ & = \int \frac{Pr[\mathbb{A}(n)|\mathbf{F}_{\mathcal{L}}, \mathbb{B}(n), Z] Pr[\mathbb{B}(n)|Z]}{Pr[\mathbb{A}(n)|\mathbf{F}_{\mathcal{L}}, Z]} Pr[Z|\mathbf{F}_{\mathcal{L}}, \mathbb{A}(n)] dZ. \end{aligned} \quad (4.56)$$

And assume  $m = 0$  or  $1$ , by total expectation,

$$\begin{aligned} & \mathbb{E}[\mathbb{B}(n)|\mathbf{F}_{\mathcal{L}}, \mathbb{A}(n)] := \\ & \sum_{m=0}^1 \mathbb{E}[\mathbb{B}(n)|\mathbf{F}_{\mathcal{L}}, \mathbb{A}(n), Z = z_m] Pr[Z = z_m|\mathbf{F}_{\mathcal{L}}, \mathbb{A}(n)]. \end{aligned} \quad (4.57)$$

Assume that the prior density  $Pr[\mathbb{B}(n)]$  is defined as in (4.55). By substituting (4.52), (4.56) into (4.57), we explicitly solved the MMSE estimator  $\widehat{\mathbb{B}}(\mathbb{A})$  as follow. For each  $n \in [1 : \frac{N}{2}]$

$$\begin{bmatrix} \widehat{B}_1(A_1(n)) \\ \widehat{B}_2(A_2(n)) \end{bmatrix} = \begin{bmatrix} A_1(n) \cdot (\sigma_{01}^2 \widehat{\tau} + \sigma_{11}^2 (1 - \widehat{\tau})) \\ A_2(n) \cdot (\sigma_{02}^2 \widehat{\tau} + \sigma_{12}^2 (1 - \widehat{\tau})) \end{bmatrix}, \quad (4.58)$$

where  $\widehat{\tau} = Pr[Z = z_0 | \mathbf{F}_{\mathcal{L}}, \mathbb{A}(n)]$ , such that:

$$\widehat{\tau} = \frac{\rho \cdot e^{\frac{\mathbb{A}^T(n) \Sigma_0 \mathbb{A}(n)}{2}}}{\rho \cdot e^{\frac{\mathbb{A}^T(n) \Sigma_0 \mathbb{A}(n)}{2}} + (1 - \rho) \cdot e^{\frac{\mathbb{A}^T(n) \Sigma_1 \mathbb{A}(n)}{2}}}. \quad (4.59)$$

The derivations of (4.58) and (4.59) are explained in detail in Appendix A.

## CHAPTER V

### EXPERIMENTAL RESULTS AND DISCUSSION

#### 5.1 A: Experimental Results and Discussion of Parameter Estimation

##### 5.1.1 Spatial Correlation of Noise

We first verify that the noise  $X_n - \lambda_n$  is spatially uncorrelated with  $X_{n+1} - \lambda_{n+1}$ . The experiment was carried out with various commercial cameras (Sony  $\alpha 7Riii$ , Nikon D5100, Nikon D90, Canon 5D) as well as smartphones (iPhoneX, Samsung Galaxy S7, OnePlus 3). We collected 100 raw sensor images from cameras placed on a tripod, taken under exactly same condition and parameters. The cameras were pointed to a blank scene content (a wall) and the focus was manually set to be maximally blurred promote maximal spatial uniformity. The light source used DC power to prevent illumination from flickering (e.g. 60Hz) during image acquisition.

We compute the Pearson product-moment correlation coefficient in (3.34) of neighboring pixels over these images. The average pixel values  $\lambda_n$  and  $\lambda_{n+1}$  were computed by averaging over the 100 images. Recalling that commercial cameras have Bayer color filter array pattern, the correlation coefficients were computed both between spatially adjacent pixels (i.e.  $\mathbf{X}_n - \lambda_n$  and  $\mathbf{X}_{n+1} - \lambda_{n+1}$ ) which have two different color filters, and between pixels that are two pixels apart (i.e.  $\mathbf{X}_n - \lambda_n$  and  $\mathbf{X}_{n+2} - \lambda_{n+2}$ ) which have the same color filters. The experiment is repeated for various combination of low/high ISO settings and shutter speeds to cover a wide range of operating conditions for the camera.

The results are reported in Table 3.1. The Pearson product-moment correlation coefficients of noise are far less than 0.1—most are smaller than 0.05. This trend is in force



Figure 5.1: Frame examples of video sequence. Each scene has two recordings—first was shot on tripod, second one shot with camera held by hand.

regardless of whether we used spatially adjacent pixels (of different color filters) or two pixels apart (of the same color filters), and the observation holds for a wide variety of cameras, including DSLRs and smartphones. Thus we conclude that the noise of raw image sensor data are largely spatially uncorrelated. Thus Theorem 3 and the pixel selection technique in (3.39) are indeed valid.

### 5.1.2 Dataset and Experiment Setup

We now describe a dataset we prepared to quantitatively evaluate the performance of Poisson parameter estimation using real-world data.<sup>1</sup> We acquired raw sensor frame data from FLIR (Point Grey) video camera (Model#: CM3-U3-13Y3M) recording at 60Hz. Because this camera does not have a built-in IMU, we did not exclude frames based on presumed camera motion magnitude. Our dataset is comprised of 9 diverse scenes of indoor and outdoor contents, including natural objects, such as trees, flowers, etc; man-made objects like stuffed animals, books, metal/plastic products, buildings; and human faces. See Figure 5.1. To cover a wide range of camera settings, every scene was captured using

<sup>1</sup>Upon acceptance of this paper, we will make this dataset and code public.

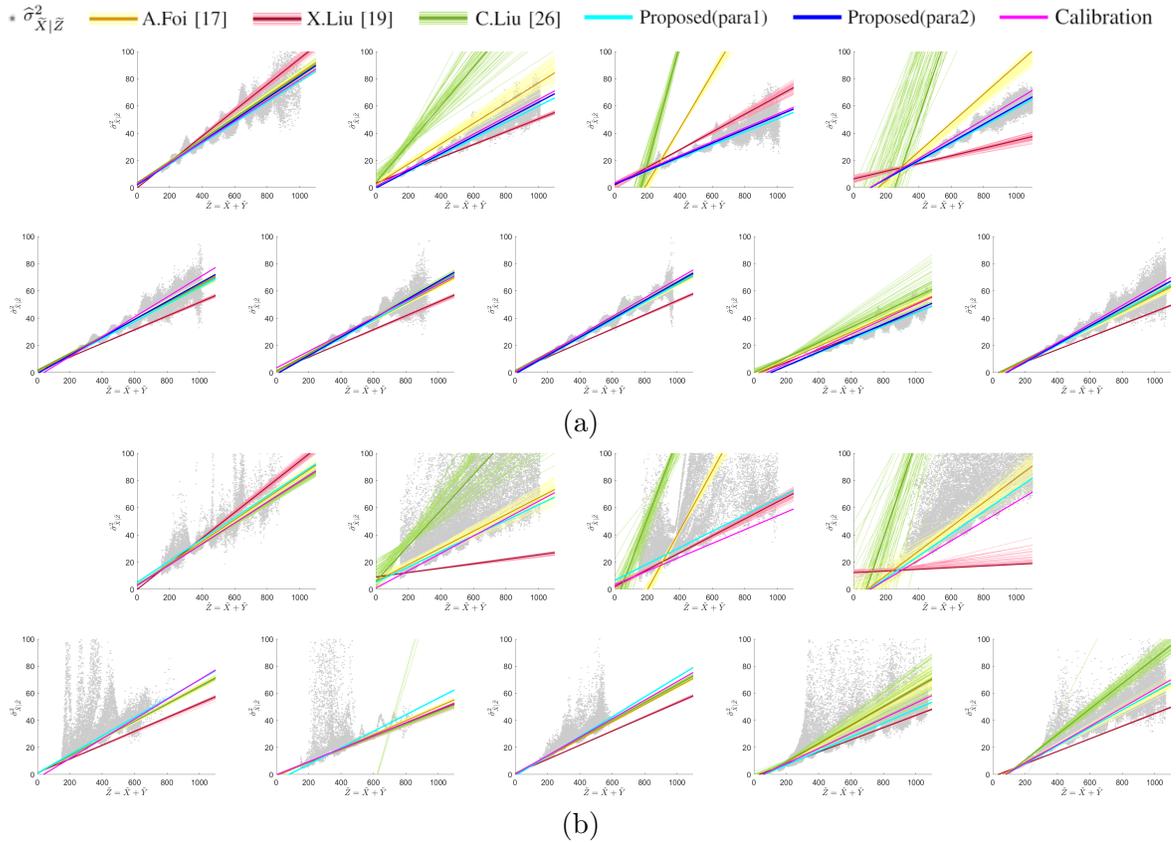


Figure 5.2: Performance evaluation of the noise parameter estimation algorithms. Each plot represents one experiment with 69 consecutive frame pairs of scenes in Figure 5.1, respectively. The plots in (a) correspond to the test video recorded on tripod, while (b) correspond to the tests with a camera that is held by hand. The results of the proposed robust lower-bound regression are shown by cyan and blue lines, while the “ground truth” line (according to the calibration) is shown by the magenta line. For comparison, we map the noise parameter estimates of methods in [1–3] to the binomial statistics plot as well. Each thin yellow/red/green line corresponds to a parameter estimate from one of the 70 frames in a video sequence. The thick line corresponds to their average. Qualitatively, we may assess the accuracy of the parameter estimation by observing the closeness of these lines to the magenta line.

Table 5.1: Quality of estimated Poisson noise parameters  $(\hat{\alpha}, \hat{\beta})$  assessed by S-MSE and N-MSE scores. The proposed algorithm yields one estimate for the entire video sequence (so we only report one MSE score). For methods in [1–3] that yield noise parameters for each frame, we report the minimum/maximum/mean/median MSE scores of all frames, which are subsequently averaged over all video sequences. We partitioned our dataset into low, medium, and high gain settings; and assessed the performance for stationary and hand-held cameras separately.

Averaged S-MSE Scores				
MethodGain #	2–5	6–10	11–13	Overall Average
Camera on tripod				
A.Foi [1]	0.1367/ 0.4744/ 0.2456/ 0.2377	0.0730/ 0.2475/ 0.1642/ 0.1630	0.0317/ 0.1247/ 0.0740/ 0.0725	0.0810/ 0.2847/ 0.1611/ 0.1574
X.Liu [3]	220.37/ 361.31/ 291.21/ 290.45	0.2782/ 0.8480/ 0.4564/ 0.4300	0.4234/ 1.8610/ 0.8073/ 0.7366	78.93/ 129.95/ 104.42/ 104.12
C.Liu [2]	0.1726/ 0.8278/ 0.3744/ 0.3644	0.1525/ $5 \times 10^7$ / $7 \times 10^9$ / 0.3842	0.1486/ $2 \times 10^8$ / $3 \times 10^9$ / 0.2424	0.1583/ $2 \times 10^7$ / $30 \times 10^9$ / 0.3265
Proposed (para1)	0.0058	0.0086	0.0169	0.0106
Proposed (para2)	0.0070	0.0161	0.0265	0.0166
Camera hold by hand				
A.Foi [1]	0.0592/ 47595/ 7512.4/ 0.2145	0.1120/ 0.2666/ 0.1767/ 0.1768	0.0380/ 11.851/ 0.2496/ 0.0785	0.0667/ 169931/ 2683.1/ 0.1552
X.Liu [3]	366.54/ 699.04/ 548.80/ 550.25	$1 \times 10^{29}$ / $5 \times 10^{45}$ / $5 \times 10^{43}$ / $2 \times 10^{34}$	0.6170/ 743.98/ 157.15/ 127.06	$3 \times 10^{28}$ / $9 \times 10^{44}$ / $1 \times 10^{43}$ / $5 \times 10^{33}$
C.Liu [2]	0.1714/ 1.7620/ 0.3978/ 0.3777	0.2404/ 0.8542/ 0.3941/ 0.4067	0.1409/ 1.4450/ 0.4039/ 0.2636	0.1802/ 1.3894/ 0.3989/ 0.3452
Proposed (para2)	0.0388	0.0395	0.0359	0.0379
Averaged N-MSE Scores				
MethodGain #	2–5	6–10	11–13	Overall Average
Camera on tripod				
A.Foi [1]	0.1626/ 0.7246/ 0.2967/ 0.2782	0.1271/ 0.2315/ 0.1740/ 0.1729	0.0530/ 0.1696/ 0.1035/ 0.1018	0.1133/ 0.3855/ 0.1927/ 0.1851
X.Liu [3]	494.13/ 852.91/ 672.79/ 671.80	0.3038/ 0.9225/ 0.5028/ 0.4766	0.6007/ 4.8496/ 1.6222/ 1.4063	176.78/ 306.61/ 241.01/ 240.57
C.Liu [2]	0.1795/ $4 \times 10^{10}$ / $7 \times 10^8$ / 0.3017	0.1785/ $1 \times 10^8$ / $2 \times 10^9$ / 0.2841	0.1377/ $3 \times 10^7$ / $5 \times 10^5$ / 0.2186	0.1643/ $1.6 \times 10^{11}$ / $2 \times 10^8$ / 0.2670
Proposed (para1)	0.0138	0.0141	0.0264	0.0184
Proposed (para2)	0.0106	0.0155	0.0375	0.0216
Camera hold by hand				
A.Foi [1]	0.0983/ $1.5 \times 10^9$ / 25051/ 0.2152	0.1362/ 0.2388/ 0.1802/ 0.1793	0.0508/ 55.008/ 0.8948/ 0.0941	0.0921/ 572401/ 8947.2/ 0.1617
X.Liu [3]	811.01/ 1651.9/ 1267.4/ 1277.5	$4 \times 10^{29}$ / $1 \times 10^{46}$ / $2 \times 10^{44}$ / $7 \times 10^{34}$	1.1646/ 3673.6/ 728.00/ 561.43	$1 \times 10^{29}$ / $4 \times 10^{45}$ / $5 \times 10^{43}$ / $2 \times 10^{34}$
C.Liu [2]	0.1717/ $7.1 \times 10^7$ / $1 \times 10^6$ / 0.3001	0.1588/ 0.6822/ 0.2948/ 0.2986	0.1213/ 1.0201/ 0.3086/ 0.2255	0.1501/ $2.5 \times 10^7$ / 370865/ 0.2730
Proposed (para2)	0.0498	0.0342	0.0412	0.0426

Averaged S-MSE Scores				
MethodGain #	2–5	6–10	11–13	Overall Average
A.Foi	7512.4	0.1767	0.2496	2683.1
X.Liu	548.80	$5 \times 10^{43}$	157.15	$1 \times 10^{43}$
C.Liu	0.3978	0.3941	0.4039	0.3989
Proposed	0.0388	0.0395	0.0359	0.0379
Averaged N-MSE Scores				
MethodGain #	2–5	6–10	11–13	Overall Average
A.Foi	25051	0.1802	0.8948	8947.2
X.Liu	1267.4	$2 \times 10^{44}$	728.00	$5 \times 10^{43}$
C.Liu	$1 \times 10^6$	0.2948	0.3086	370865
Proposed	0.0498	0.0342	0.0412	0.0426

two or more gain values (ranging between 2–13) and at least two exposure time settings. Each scene was recorded by holding the camera by hand (i.e. with both scene and camera motions), and by camera mounted on a tripod (i.e. only foreground motion). In total, we prepared 42 video sequences. From this video sequence, we used 70 frames each (69 pairs of adjacent frames for our algorithm).

In each video sequence, X-Rite Macbeth Colorchecker (October 2015 edition) was recorded in the first few frames. The Colorchecker is used to determine the ground truth noise parameters  $\alpha$  and  $\beta$  by a calibration (by regressing mean and variance of Colorchecker patches), but is excluded from the performance evaluation of Poisson parameter estimation technique. For each scene, we reconstructed the Poisson image  $X = (\tilde{X} - \beta)/\alpha$  as a reference image. We then estimated  $(\hat{\alpha}, \hat{\beta})$  using the proposed technique, reconstruct the Poisson image  $\hat{X} = (\tilde{X} - \hat{\beta})/\hat{\alpha}$ , which was compared to the reference image  $X$  in the mean squared sense for performance evaluation in Table 5.1. Comparing  $\hat{X}$  to  $X$  is preferred over directly computing error between  $(\hat{\alpha}, \hat{\beta})$  and  $(\alpha, \beta)$  because of the fact that smaller  $\hat{\alpha}$  value can be compensated for by a larger  $\hat{\beta}$  value, etc.

The proposed Poisson noise parameter estimation technique has several control variables:  $\kappa$  in (3.32); and  $\gamma_1, \gamma_2$ , and  $\eta$  in (3.30); and  $\epsilon$  in (3.39). We chose two sets of values empirically, one for stationary camera on tripod (“param1” with  $(\kappa, \gamma_1, \gamma_2, \eta) = (10, 0.5, 0.5, 33)$ ) and another for hand-held camera (“param2” with  $(\kappa, \gamma_1, \gamma_2, \eta) = (10, 0.5, 0.01, 33)$ ) tuned more aggressively towards the lower-bound in (3.25) to deal with motion. The value of  $\epsilon$  is set to admit 45% of the pixels in  $\hat{\Gamma}$  in (3.39).

We compared our results to state-of-the-art Poisson noise parameter estimation methods in [1–3]. Method in [2] is a patch-based non-parametric noise parameter estimation

technique (i.e. not specific to Poisson). To make the comparison more fair, we improved its performance by enforcing the output space to be restricted to Poisson noise parameter space. Since methods in [1–3] are designed for a single frame, we apply these techniques on all 70 images yielding 70 different noise parameter estimates  $(\alpha, \beta)$ . These parameters were averaged (per scene) to yield a final estimate of  $(\alpha, \beta)$ , which was subsequently used in Table 5.1 to compute the performance scores.

### 5.1.3 Noise Parameter Estimation Performance

Recall Corollary 1 describing the relationship between the binomial statistics in (3.17) and the affine line  $f(\tilde{z})$  in (3.24). In Figure 5.2, we show the scatter plot of the binomial statistics  $\hat{\sigma}_{\tilde{X}|\tilde{Z}}^2$  as a function of  $\tilde{Z}$  over the 9 video sequences (grey dots). The proposed method uses a robust lower-bound regression as shown by cyan (tuned for hand-held camera) and blue (tuned for tripod) lines. For reference, we plot the “ground truth line” (magenta) corresponding to the calibration noise parameters  $(\alpha, \beta)$  obtained from Colorchecker—qualitatively, we may assess the performance of the proposed method by observing how close cyan and blue lines are to the magenta line.

In tripod sequences, the binomial statistics (grey dots) are clustered closely, thanks to the absence of background motion, with variation only due to foreground/scene motion and noise. As expected, the blue line (tuned for tripod sequence) is closer to the magenta line in stationary camera case, though the cyan line also behaves well. The hand-held camera sequences are more difficult, as evidenced by the wider range of binomial statistics. Yet, the proposed robust lower-bound regression is able to handle the scene variations—the cyan line never deviates far from the magenta line, indicating robustness of the algorithm.

In Figure 5.2, we compare this to the state-of-the-art Poisson parameter estimation methods in [1] (yellow), [3] (red), and [2] (green). The thin lines correspond to estimation of  $(\alpha, \beta)$  on each of the 70 frames; the thick lines show the estimated  $(\alpha, \beta)$  averaged over the 70 frames. As evidenced by Figure 5.2, these methods are influenced by scene content, with no prior method performing reliably across all scenes. Even within the same scene, the variation among frames were high (especially for method in [2]). This is understandable—the patch-based algorithms rely on finding homogeneous regions within an image. Thus when using a wide-angle lens (where image details are smaller) or if a scene is dominated by textures, these algorithms fail. In presence of strong noise, the ability to identify smooth regions is difficult. By comparison, the proposed method relies on global statistics, insensitive to textures and small image features. We gain robustness to motion using more aggressive robust lower-bound regression, the improvements we described in Section 3.2.3, and by the use of multiple video frames instead of relying on a single frame. Overall, the reliability of the proposed method is highest among the Poisson parameter estimation techniques tested.

Table 5.1 summarizes our a quantitative performance evaluation of the proposed Poisson parameter estimation method. For every single frame, we compute the signal-normalized MSE (S-MSE) score, defined as:

$$\text{S-MSE} = \frac{1}{N} \sum_{n=1}^N \left\| \frac{X_n - \hat{X}_n}{\nu} \right\|^2, \quad (5.1)$$

where  $\nu$  denotes the mean value of “Neutral 5” patch on the Colorchecker (to normalize to the illumination radiance). We also adopt another performance metric convention commonly used in the Poisson rate estimation literature. Specifically, let  $\psi_j$  and  $\hat{\psi}_j$  refer to the average pixel values of  $X_n$  and  $\hat{X}_n$  in the  $j$ -th color patch of the Colorchecker. Then we

compute the noise-normalized MSE (N-MSE), defined as:

$$\text{N-MSE} = \frac{1}{J} \sum_{j=1}^J \left\| \frac{\psi_j - \hat{\psi}_j}{\psi_j} \right\|^2. \quad (5.2)$$

The intuition here is that the noise variance in  $X$  and  $\hat{X}$  scales linearly with the mean value  $\psi_j$ . Hence (5.2) normalizes by the noise variance.

Table 5.1 reports the S-MSE and N-MSE scores of various noise parameter estimation techniques. The proposed method estimates unique  $(\hat{\alpha}, \hat{\beta})$  for an entire sequence, and therefore have one S-MSE and N-MSE score each. On the other hand, the methods in [1–3] yield the Poisson noise parameter estimates  $(\hat{\alpha}, \hat{\beta})$  for each of the 70 frames. Thus determine the minimum, maximum, mean, and median S-MSE/N-MSE scores for each video sequence. We report in Table 5.1 the average the above min/max/mean/median scores over all the video sequences.

We draw the following conclusions from Table 5.1. The S-MSE and N-MSE scores indicate that the proposed method performs most reliably among all methods considered. In fact, the performance of our method is better than even the minimum S-MSE/N-MSE scores of the alternatives. The maximum MSE scores of the state-of-the-art methods are much higher than that of minimum and median scores, suggesting a large disparity of estimation performance among the video sequences. As predicted, the performance of the proposed method is better when camera is placed on tripod, and by and large it is insensitive to the gain setting when the camera is hand held.

#### 5.1.4 Discussion

We experimentally support the claim of the invariance to scene content by providing a result estimating the noise parameters from a textured patch, which is arguably the most

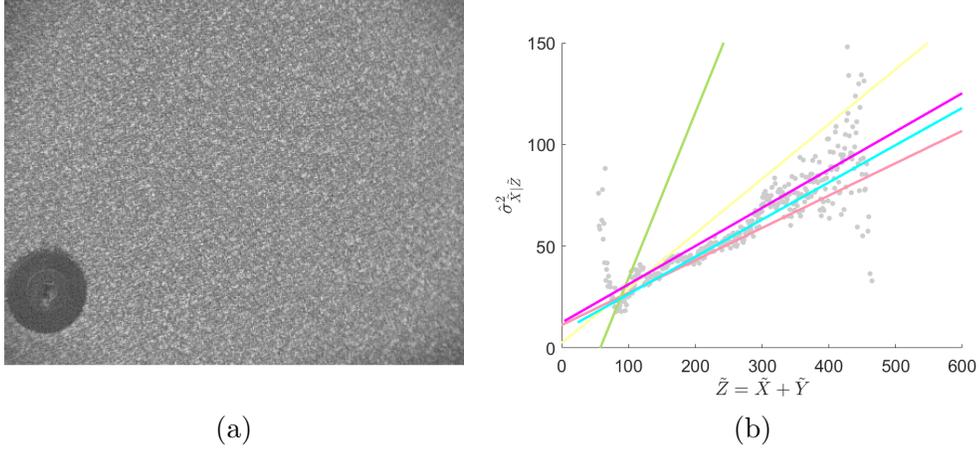


Figure 5.3: (a) An example frame from a noisy texture patch video sequence. (b) The empirical binomial statistics  $\hat{\sigma}_{\tilde{X}|\tilde{Z}}^2$  in (3.27). The plot follows the legends used in Figure 5.2. The recovered S-MSE/N-MSE scores from the proposed method are 0.0062/0.0020. Compare this to the S-MSE/N-MSE scores of 0.4429/0.1733 by method in [1], 0.0911/0.0383 by method in [3], and 1.7271/0.7209 by method in [2].

difficult scene for any noise parameter estimation algorithms. Figure 5.3 shows an example of a frame capturing a highly textured surface. In absence of any homogeneous patch, the state-of-the-art noise parameter estimation methods fail (see S-MSE and N-MSE scores in the figure caption). Nevertheless, the proposed noise parameter estimation provides reliable result, thanks to the invariance of the joint binomial statistics of the adjacent frames to the scene content.

Lastly, recall that the proposed algorithm performs the best when  $\lambda_n = \mu_n$ . One way to achieve this is to let  $\tilde{X}_n$  and  $\tilde{Y}_n$  correspond to consecutive frames in a video sequence— $\lambda_n$  and  $\mu_n$  correspond to the same scene point after frame registration (Section 3.2.3). One alternative to this approach is to consider a single-shot framework, where  $\tilde{Y}_n$  can be a shifted version of the image  $\tilde{X}_n$  (i.e.  $\tilde{Y}_n = \tilde{X}_{n-1}$ ). In absence of an edge or texture,  $\lambda_n \approx \lambda_{n-1}$ , this method can be thought of as a global-statistics alternative to the single-shot noise parameter

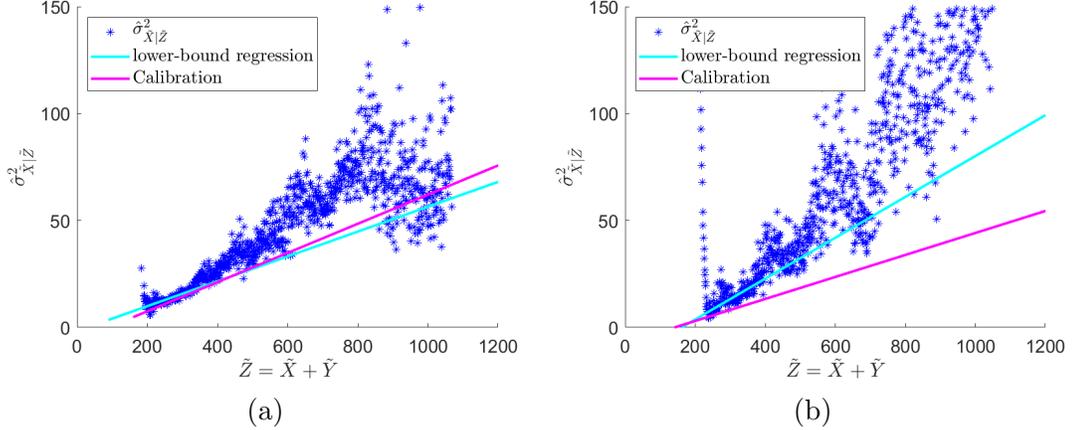


Figure 5.4: Successful and non-successful examples of binomial statistics computed from a single frame with  $(\kappa, \gamma_1, \gamma_2, \eta) = (10, 0.5, 0.002, 15)$ . In the scene in (a), the proposed robust lower-bound function regression successfully estimates the noise parameter. In (b), the noise parameter estimation is more difficult. The topic of single frame noise parameter estimation is left for future work. See text.

techniques in [1–3, 22, 37, 38]. The pixel selection algorithm in (3.39) can be thought of as an edge detector (to exclude unfavorable pixels from noise parameter estimation). The main advantage of this single-shot variant of the proposed noise parameter estimation technique over the state-of-the-art methods is that it does not require a large homogeneous patch (only need two adjacent pixels to be similar). However, it is still sensitive to textures (where no adjacent pixels are alike) and therefore it is not entirely invariant to the scene content. See Figure 5.4. Improving the single-shot variant of the proposed method is left as a future topic of research.

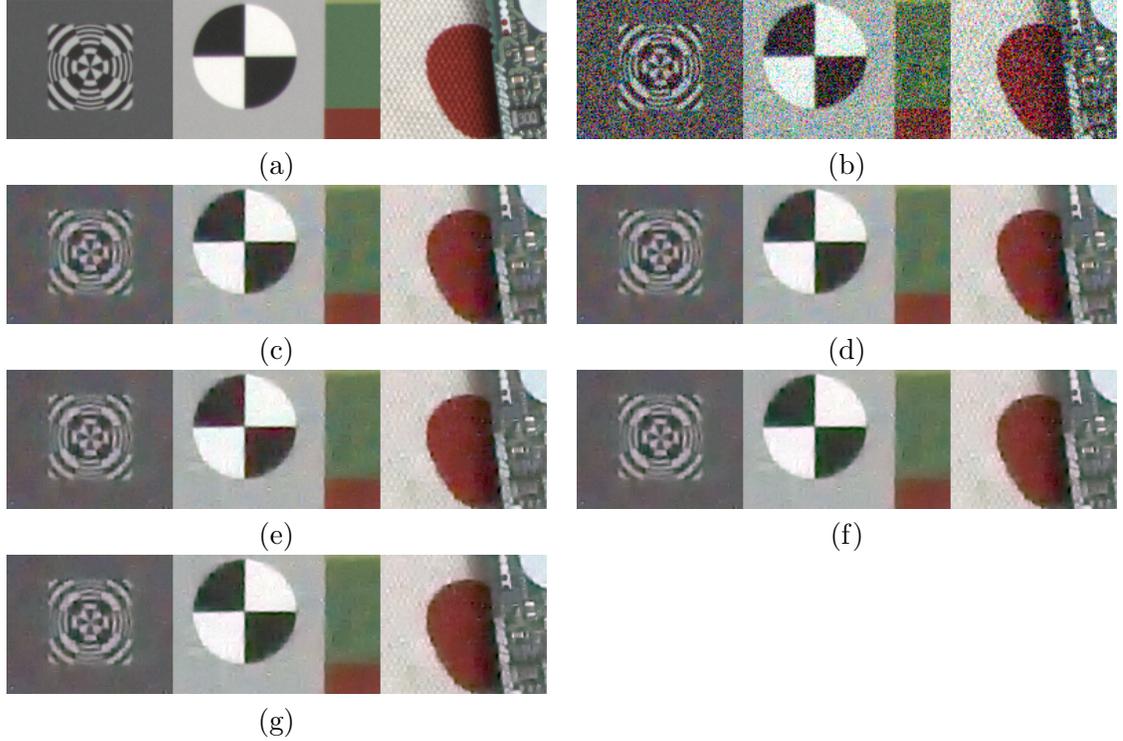


Figure 5.5: Behavior analysis for tunable parameter  $\eta.y_{\bar{B}_1}$  and  $\eta.y_{\bar{B}_2}$ . (a) Relatively clean image taken by increasing the exposure time. (b) Noisy image taken under low illumination. (c) Denoised result with very low value of  $y$  under three levels of Haar wavelet decomposition. (d) Denoised result with very high value of  $y$  under three levels decomposition. (e) Denoised result with very low value of  $y$  under four levels decomposition. (f) Denoised result with very high value of  $y$  under four levels decomposition. (g) Denoised result with fine tuned value of  $y$  under four levels decomposition.

## 5.2 B:Experimental Results and Discussion of Color Image Denoising

### 5.2.1 Parameter Tuning Methodology for Proposed Denoising

Here, we remind the reader that the focus of this paper is color denoising in log-chromaticity coordinate. As for the luminance component denoising, we simply borrow the ideas that described in Section 2.2.3. Next, we will discuss the parametric tuning of the proposed estimator. In summary, the proposed estimator contains tunable variables as follow:  $\rho$ ,  $\sigma_{01}$ ,  $\sigma_{11}$ ,  $\sigma_{02}$ , and  $\sigma_{12}$ . The meanings of each have been explicitly explained above.

In this research, we have investigated at most five levels Haar-wavelet decomposition. In every wavelet scale, there are three subbands corresponding to horizontal, vertical, and diagonal components sharing the same parent coefficients (the finer-scale scaling coefficients). Therefore, we claim to fix a unique  $\rho$  value for each wavelet scale. And due to the compact property of wavelet, the clean coefficients are dominated by the  $\{Z = z_1\}$  elements, and are more sparse in finer scale than the coarser one. In particular, after studying the cleaner images of natural scene, like we described in Figure 4.2, the  $\rho$  values in this research were chosen within the range between 0.05 to 0.1. And their values should be increasing monotonically from the first (finest) to the fifth (coarsest) level.

Unlike conventional denoising schemes, the wavelet-based reconstruction of  $\mathbb{B}$  comes indirectly from non-wavelet noisy input  $\mathbb{A}$  rather than directly from the corresponding  $\tilde{\mathbb{B}}$ . It seems ambiguous to determine the variance values. Therefore, we seek for a more robust environment for parameter tuning. For convenient, we restrict the relationship rules for variance pair such that

$$\sigma_{11}^2 = t1 \times \sigma_{01}^2, \text{ and } \sigma_{12}^2 = t2 \times \sigma_{02}^2, \quad (5.3)$$

where the ratio values  $t1, t2 \ll 1$ . In real experiments, we found that a good choice of  $t1, t2$  could be less than 0.01. Meanwhile, the denoising performance is less sensitive with  $\sigma_{1m}$ , but more relies on the choice of  $\sigma_{0m}$ . It matches the fact that the variance value corresponding to edges should be obviously larger than the one to clean smooth areas.

Finally, we draw our attentions to the tuning of  $\sigma_{0m}$ 's. From Equation (4.58), we can learn two basic features. First, the  $\sigma_{01}^2 \cdot \mathbf{A}_1$  and  $\sigma_{02}^2 \cdot \mathbf{A}_2$  have to be upper bounded by  $\max(\mathbf{B}_1)$  and  $\max(\mathbf{B}_2)$ . In theorem, those hidden parameter can be found by  $\sigma_{01} = \sqrt{\frac{\max(\mathbf{B}_1)}{\max(\mathbf{A}_1)}}$ , and  $\sigma_{02} = \sqrt{\frac{\max(\mathbf{B}_2)}{\max(\mathbf{A}_2)}}$ . Ideally, we need to choose  $\max(\hat{\mathbf{B}}_1) \approx \max(\mathbf{B}_1)$  and

$\max(\widehat{\mathbf{B}}_2) \approx \max(\mathbf{B}_2)$ . Second, as long as  $\sigma_{0m}$ 's are fixed values, the estimator in (4.58) is an one-to-one function. However, the pixel values which are close to the maximum have higher chance to be outliers or suffer from saturation. So, let  $\eta_{.x_{A_m}}$  denotes the  $x$ -th percentile of  $\mathbf{A}_m$ , instead of  $[\max(\mathbf{A}_1), \max(\mathbf{A}_2)]$ , we are mapping the near maximum values  $[\eta_{.x_{A_1}}, \eta_{.x_{A_2}}]$  to  $\max(\mathbb{B})$ .

More importantly,  $[\max(\mathbf{B}_1), \max(\mathbf{B}_2)]$  is not directly available. So to determine the accurate upper bounds becomes impractical. By comparing the range and distribution of Figure 5.5(a) and (b), we realize that there exist  $y < 100$  such that the  $y$ -th percentile of  $\widetilde{\mathbb{B}}$ :  $[\eta_{.y_{\widetilde{B}_1}}, \eta_{.y_{\widetilde{B}_2}}] = [\max(\mathbf{B}_1), \max(\mathbf{B}_2)]$ . Then the denoising parameters are computed by:

$$\sigma_{01} = \sqrt{\frac{\eta_{.y_{\widetilde{B}_1}}}{\eta_{.x_{A_1}}}}, \text{ and } \sigma_{02} = \sqrt{\frac{\eta_{.y_{\widetilde{B}_2}}}{\eta_{.x_{A_2}}}}. \quad (5.4)$$

As a result, the tunable parameters switches from  $\sigma_{0m}$ 's to the pairs of  $(x, y)$ . Based on our experiments, we found a reasonable range of  $x$  could be  $1 \sim 3$ . Since the coarser level wavelet coefficients naturally suppress the noise, the values of  $y$ 's are increasing as the wavelet level growing up.

Next, as shown in Figure 5.5, we compared several denoising results using over-high values of  $y$  ( $y = 98$ ) for (c,e) and very low values ( $y = 5$ ) for (d,f). The results in (c) and (d) have obvious color artifacts in the flatter areas. By contrast, the results in (e) and (f) are cleaner because they were denoised under one extra level of wavelet decomposition. Note that the parameters we used for (c) and (d) remained the same for (e) and (f) in the three most fine levels. Unlike (c) and (d) whose results are generally similar ((c) is slightly noisier than (d)), the result in (e) has strong over-shooting defects in the edges, but closer sharpness as (c,d). On the contrary, the result in (f) is too over-smoothed and having strong bleeding artifacts in the edge. We repeated this comparisons using 5 different scenes

with several different noise levels, and observed the same phenomenon. Therefore we have conclusions as follow: 1), the over-shooting and bleeding artifacts are more sensitive to the choice of  $y$ 's in coarser level, particularly for the 4-th or higher level coarser subbands. 2), for the first 2 or 3 levels fine subbands, the values of  $y$ 's should be low to have adequate enough noise reduction. In summary, as an example illustrated in Figure 5.5(g), we expect a range of upper bounds that just smaller than the values producing over-shooting and bigger than the values can result in bleeding artifact.

### 5.2.2 Verification Using Synthetic Images

Let  $\log \hat{\theta}(\mathbb{F})$  denote any denoising function estimating the latent clean chromaticity  $\theta$  in log domain. In the previous sections, we established a scheme that in log chromaticity coordinate,  $\log \hat{\theta}_{\mathbb{B}}(\mathbb{A})$  could achieve the minimum  $\ell^2$  error with respect to  $\log \theta$ , such that:

$$\gamma \left[ \log \hat{\theta}_{\mathbb{B}}(\mathbb{A}) \right] \leq \gamma \left[ \log \hat{\theta}(\mathbb{A}) \right] \quad (5.5)$$

To verify this, we implement the proposed method using synthetic images, and compare its performance with some other existing color denoising techniques. We collected 63 different natural scenes of cleaner RGB images using Sony  $\alpha 7R$  camera, and used them as ideal references  $\lambda$ . Then we simulated the Poisson corrupted images with three levels of pseudorandom noise. In this study, the level of noise is represented in terms of the maximum intensity of  $\mathbb{F}$ . Then we denoise all the  $63 \times 3$  samples using CBM3D [5] and Anscombe transformation [9], the color-MRSO [4], the dnCNN [6], and the proposed method, respectively. For better visual performance, the CBM3D, the color-MRSO (CMRSO), and the dnCNN methods are implemented in color-opponent space. In particular, the use of CMRSO is described in Section 2.2.2 since the original work was designed for gray-scale image. For the use of dnCnn, we implemented it using the MATLAB 2018b function from the Deep Learning

Table 5.2: Summation MSE score in log chromaticity coordinates averaged over 63 samples. Each score is computed as  $\sum_m \gamma \left[ \log \hat{\theta}_m(\mathbb{F}) \right]$ . The maximum intensity of  $\mathbb{F}$  indicates the noise level.  $\max(\mathbb{F}) = 20$  corresponds to the moderate noisy scenario.  $\max(\mathbb{F}) = 10$  corresponds to the high noisy scenario.  $\max(\mathbb{F}) = 5$  corresponds to the extremely noisy scenario. Each noisy image was denoised with four different methods. The method with score in bolder face has the minimum  $\ell^2$  error, indicating the best obtained performance in each noise level.

	Denoising Method			
$\max(\mathbb{F})$	CBM3D [5]+ [9]	CMRSO [4]	dnCNN [6]	Proposed
20	0.0057	<b>0.0051</b>	2.0187	0.0062
10	0.0104	0.0098	9.7161	<b>0.0087</b>
5	0.0212	0.0185	33.3189	<b>0.0130</b>

Toolbox<sup>TM</sup>. The MATLAB build-in function was not pre-trained using RAW sensor data. In order to better match the scale of training image, we tried several ways, such as to include white-balance before denoising, or to compare gamma correction before/after denoising. We realize that the build-in training works better with pre-white-balance and pre-gamma corrected image. So we kept those pre-processing for both simulation experiments and for the following real data experiments.

With the discussion above in mind, we report in Table 5.2 the MSE comparison between different denoising methods in log chromaticity coordinate. For each sample we collected summation MSE scores of four denoising methods' result. The summation score was computed by the sum of  $\ell^2$  errors in each color channel, such as  $\sum_m \gamma \left[ \log \hat{\theta}_m(\mathbb{F}) \right]$ . From the experimental results we see that when noise level is low (the maximum pixel value is high), the denoising performances of CBM3D and CMRSO are compatible with the proposed method. However, when the noise level is high, the performances of compared methods drop faster than the proposed one.

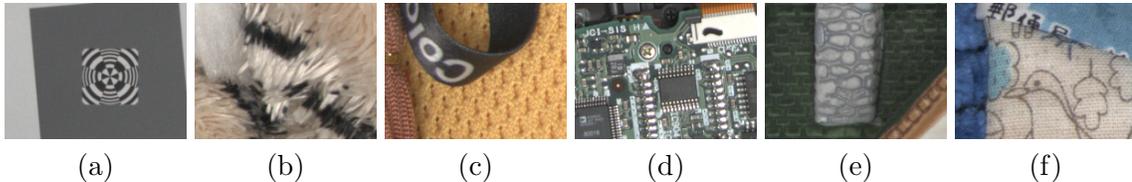


Figure 5.6: The Relatively clean image taken by increasing the illumination. Scenes were captured using Sony  $\alpha 7R III$  with pixel-shifting mode to capture the full resolution R/G/B data. Clean images are used as the proxies of latent color intensities.

### 5.2.3 Real Camera Sensor Data Experiment

Finally, we verify the visual performance of the proposed color denoising. Similar as we mentioned before, several scenes of images were captured using Sony  $\alpha 7R III$  in raw sensor mode with all manual settings under low light condition. For each scenes, camera settings were varied to ensure that both less/moderate/very noisy conditions were being included. We captured a diverse set of scene contents comprised of resolution chart, natural objects, man-made objects, stuffed animals, etc. The Sony pixel-shifting mode enables the access of full resolution RGB data without the requirement of demosaicking technique. Each scene was captured with the Gretag Macbeth Colorchecker [42] being included. Because the raw sensor data value  $\mathbb{H}$  is assumed to be an affine transformation of the Poisson count variable  $\mathbb{F}$ , such that  $\mathbb{H} = \alpha\mathbb{F} + \beta$ . And the affine noise parameters  $\alpha$  and  $\beta$  could be accurately determined by the Colorchecker calibration process [34]. Following the procedures introduced above, the denoising parameters were chosen to avoid over-shooting artifact, and to keep as much sharpness as we could. In order to show the robustness, we kept using same set of parameters to denoise all the scenes. For very noisy condition, we used 4 levels of Haar-wavelet transform. For the rest, we used 3 levels of transform. Same as the test described in Figure 5.5, the parameters for 3 levels denoising remained the same in the first three finest scales

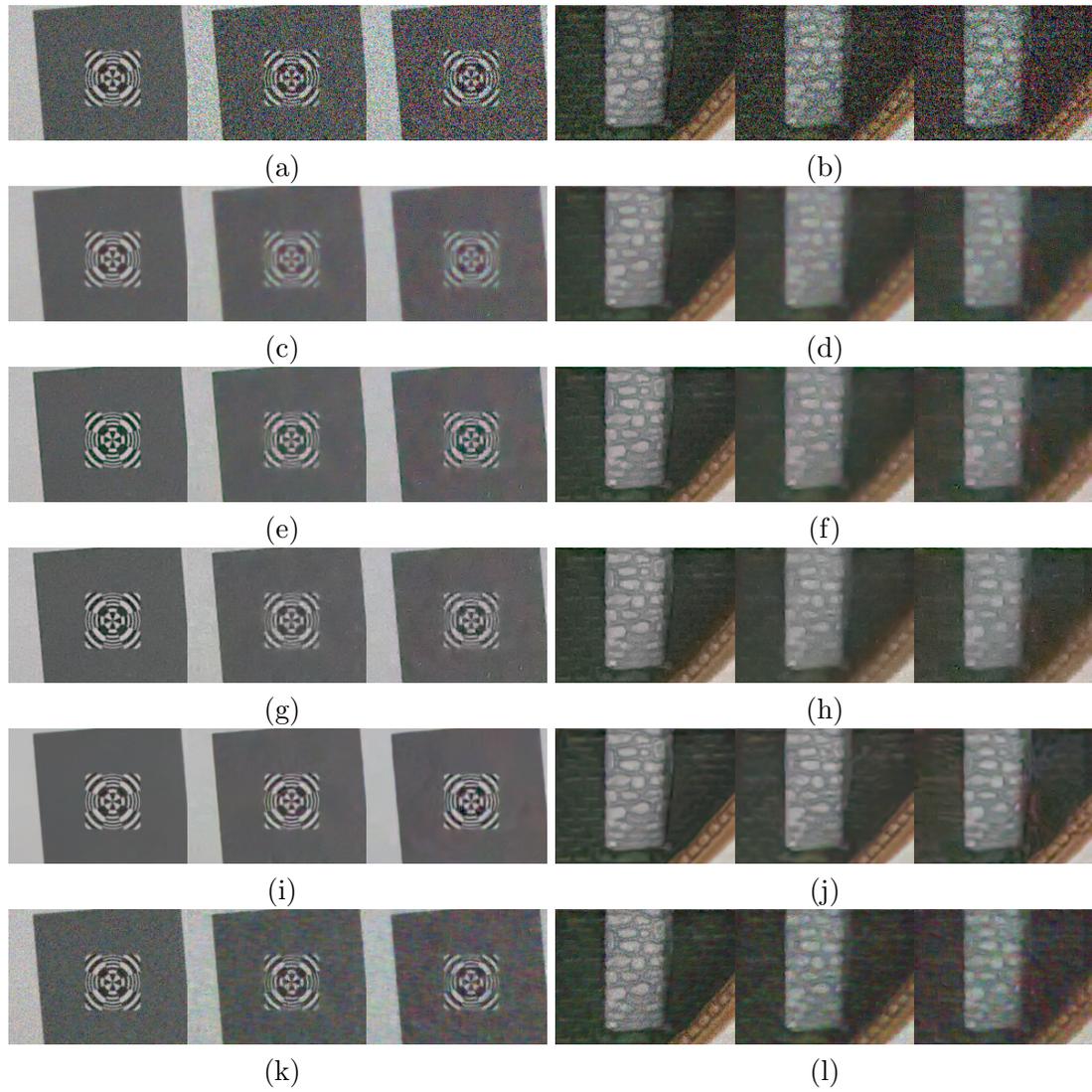


Figure 5.7: Color denoising results comparison. The reference image correspond to Figure5.6(a) and (e). Noisy images were taken under low illumination using same camera as Figure5.6, but three different camera settings. The acquired less/moderate/very noisy images are displaced in a same picture to demonstrate the degradation change according to noise level. (a,b) Noisy images. (c,d) Denoising each color channel independently using MRSO [4]. (e,f) Denoising in color-opponent space using (2.22)–(2.27). (g,h) Proposed color denoising in log-chromaticity coordinate. (i,j) Denoising using CBM3D [5]. (k,l) Denoising in color-opponent using dnCnn [6].

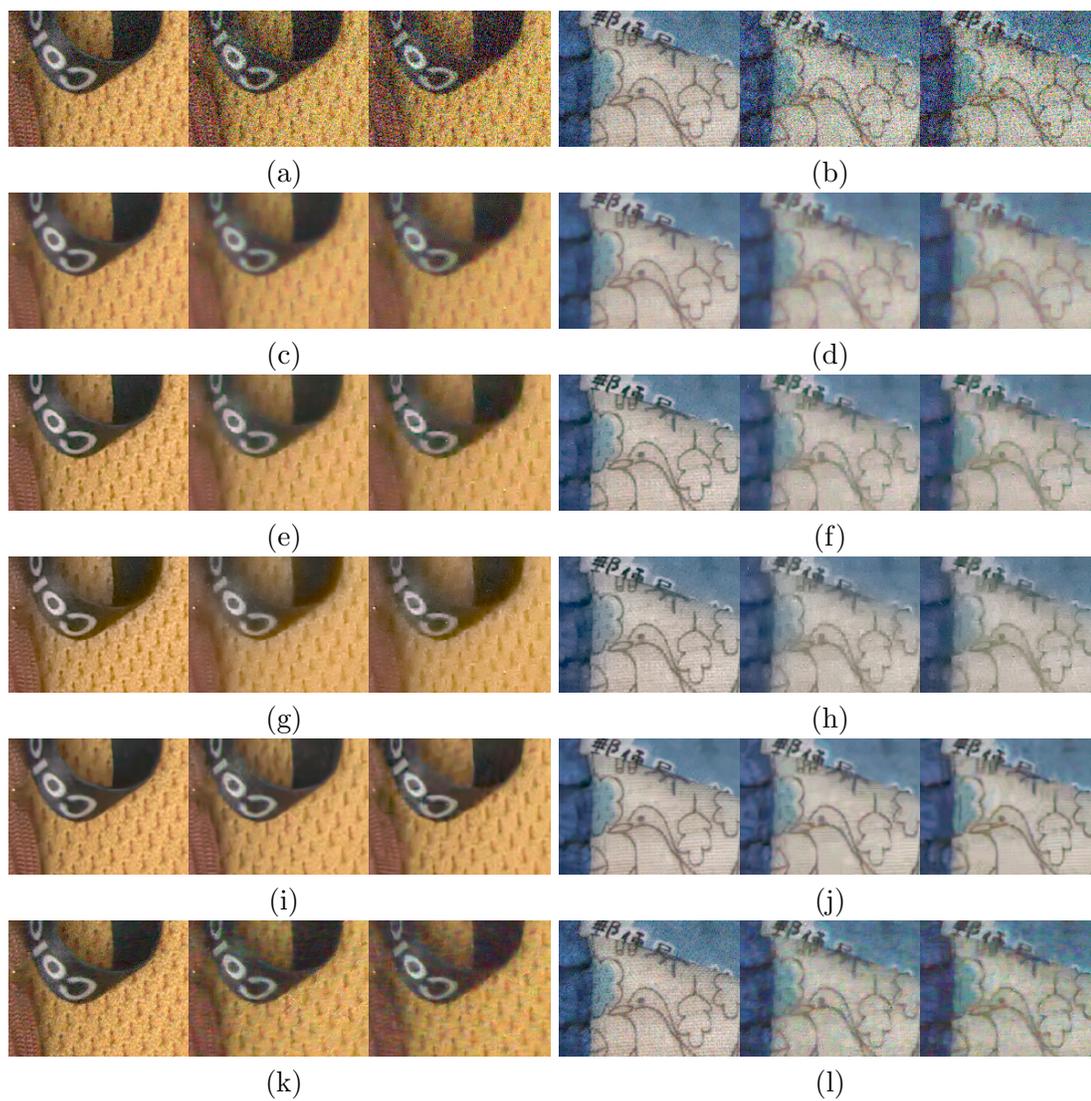


Figure 5.8: Some other examples of color denoising results comparison. The reference image correspond to Figure5.6(c) and (f). The figure arrangement follows Figure 5.7.

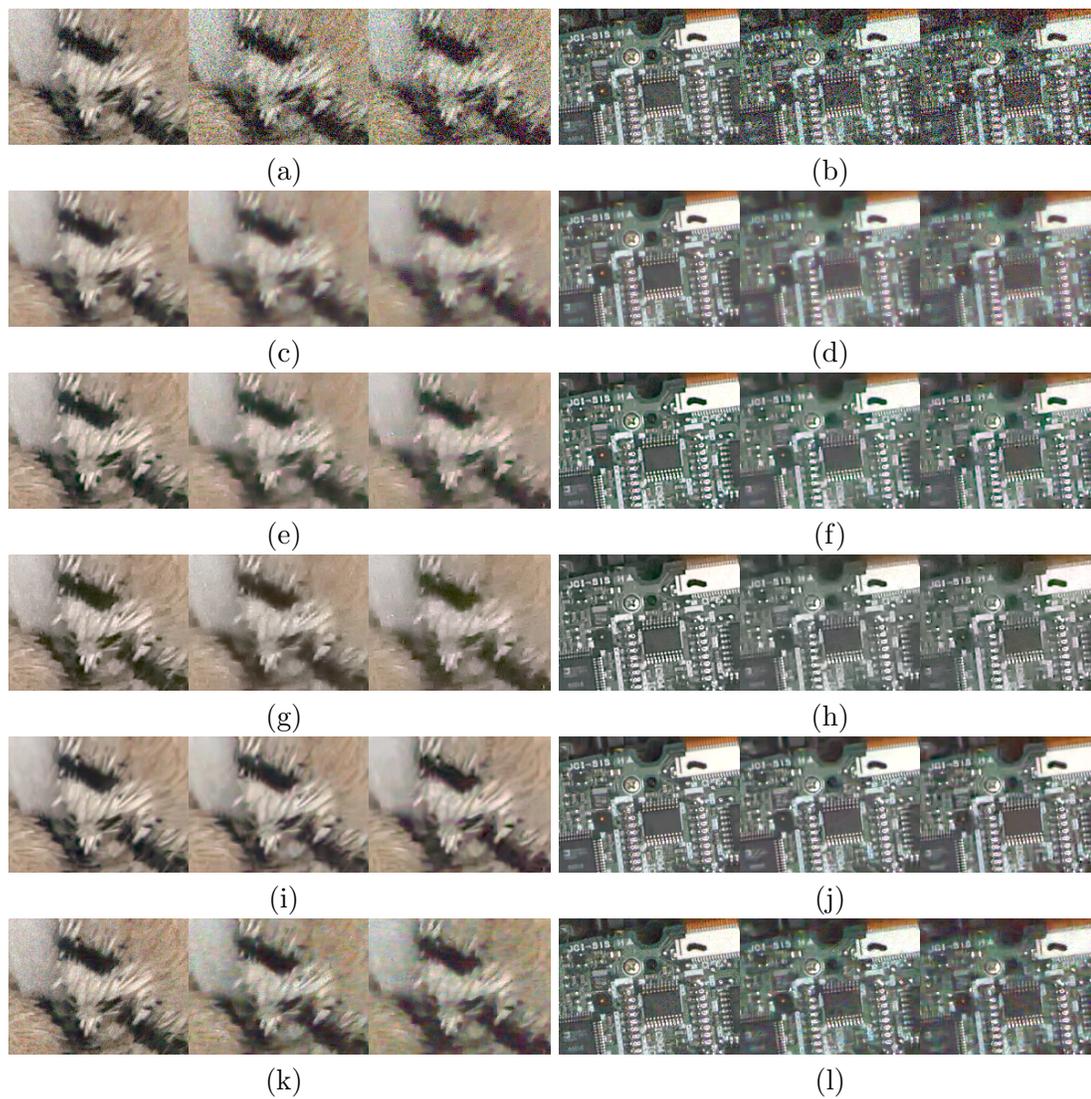


Figure 5.9: Some other examples of color denoising results comparison. The reference image correspond to Figure 5.6(b) and (d). The figure arrangement follows Figure 5.7.

of 4 levels denoising. Similarly, we compared its visual quality with other results denoised by CBM3D [5] and Anscome-transform [9], original MRSO [4] (denoising each color channel independently), CMRSO in color-opponent space, and dnCnn [6] in color-opponent space. Using Figure 5.6 as cleaner references, the comparisons are illustrated in Figure 5.7—5.9. As expected, CBM3D results in (i,j) and dnCnn results in (k,l) work fine during less noisy conditions, but will have strange artifacts in higher noise levels. The dnCnn denoising could be better if we have enough real data to re-train the network. However, it is not practical in this work because the lack of real noisy image dataset. As for the using of original MRSO in each color channel independently, as shown in (c,d), the color artifacts are obvious and spatially inconsistent. If do CMRSO denoising in color-opponent space, as shown in (e,f), the colors are much more spatially invariant. However, the color shifting around the edge are strong. For example, around black-white line patterns, the white areas turn in pink. This is the color artifacts suffered from white-balance. It matches the discussion we introduced in Figure 2.2. By contrast, the proposed denoising in log-chromaticity coordinate, as results shown in (g,h). It can maintain the spatial consistency, meanwhile can achieve the correct color noise reduction in both flatter and edge areas. Remember that the proposed method shares the same luminance denoising as CMRSO in this research. Thus (e,f) and (g,h) have the same spatial sharpness. So the color improvement from (e,f) to (g,h) demonstrates the significance of the proposed CLRT domain strategy. Meanwhile, here we would like to emphasize reader that the color edges were not obviously influenced by the trade-off described in (4.40).

Based on the experimental results for color denoising using synthetic data and real camera data, the proposed denoising strategy outperformed the other compared methods. However, the proposed method is not without limitation. First of all, the tuning process of

denoising parameter is time consuming. If the noise level is far different from our testing data, it might require different setup to achieve the best performance. Secondly, in this paper, the luminance denoising is borrowing the idea of original MRSO [4]. Sometimes, the spatial sharpness of luminance is not compatible with popular patch-based denoising method, e.g. CBM3D [5]. The overall performance would be better if we could have better luminance reconstruction. Thirdly, the choice of Prior model directly control the performance of chromaticity denoising. The mixture of two Gaussian's is not perfectly accurate to describe the cleaner chromaticity in log scale. Therefore, for future study, 1), we will extend the CLRT domain work to luminance study to improve the spatial sharpness. 2), we will try to extend the current work using other Prior model to further improve the color performance.

## CHAPTER VI

### CONCLUSIONS

#### 6.1 A: Conclusions of Parameter Estimation

We proposed a Poisson noise parameter estimation technique designed specifically for video cameras. We showed that the noisy pixel value  $\tilde{X}$  conditioned on the sum of two noisy pixels across frames  $\tilde{Z} = \tilde{X} + \tilde{Y}$  is a Binomial random variable, whose variance  $\sigma_{\tilde{X}|\tilde{Z}}^2$  scales linearly with  $\tilde{Z}$ . We proved that the Poisson noise parameters can be recovered from the slope and intercept of this line, which we carry out using the proposed robust lower-bound regression method. Though the algorithm is sensitive to foreground and background movements, we largely overcome them using camera motion and scene motion analysis. The algorithm is *not* sensitive to textures and edges, which is the main advantage over the existing Poisson noise parameter methods.

#### 6.2 B: Conclusions of Color Image Denoising

We proposed a new Poisson color image denoising strategy for RGB video camera. The idea is to reconstruct the latent chromaticity information, which is fundamentally different with almost all of the existing denoising techniques. We have shown it again that chromaticity coordinate analysis is the stable representation of color signal. However, since it is defined in terms of ratio, the computational difficulty limited people to work with it directly. We largely overcome this problem by providing CLRT transform and corresponding analysis. Based on that, we proposed one MMSE estimator denoising framework, enabling a spatially invariant way for future color image denoising.

## BIBLIOGRAPHY

- [1] A. Foi, M. Trimeche, V. Katkovnik, and K. Egiazarian, “Practical poissonian-gaussian noise modeling and fitting for single-image raw-data,” *IEEE Transactions on Image Processing*, vol. 17, no. 10, pp. 1737–1754, Oct 2008.
- [2] C. Liu, W. T. Freeman, R. Szeliski, and S. B. Kang, “Noise estimation from a single image,” in *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR’06)*, vol. 1, June 2006, pp. 901–908.
- [3] X. Liu, M. Tanaka, and M. Okutomi, “Practical signal-dependent noise parameter estimation from a single noisy image,” *IEEE Transactions on Image Processing*, vol. 23, no. 10, pp. 4361–4371, 2014.
- [4] W. Cheng and K. Hirakawa, “Minimum risk wavelet shrinkage operator for poisson image denoising,” *IEEE Transactions on Image Processing*, vol. 24, no. 5, pp. 1660–1671, 2015.
- [5] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, “Image denoising by sparse 3-d transform-domain collaborative filtering,” *IEEE Transactions on Image Processing*, vol. 16, no. 8, pp. 2080–2095, Aug 2007.
- [6] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang, “Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising,” *IEEE Transactions on Image Processing*, vol. 26, no. 7, pp. 3142–3155, July 2017.
- [7] J. R. Janesick, “Photon transfer,” *Bellingham: SPIE*, 2007.
- [8] M. Mäkitalo and A. Foi, “Poisson-gaussian denoising using the exact unbiased inverse of the generalized anscombe transformation,” in *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2012, pp. 1081–1084.
- [9] M. Makitalo and A. Foi, “Optimal inversion of the generalized anscombe transformation for poisson-gaussian noise,” *IEEE Transactions on Image Processing*, vol. 22, no. 1, pp. 91–103, Jan 2013.
- [10] M. Raphan and E. P. Simoncelli, “Learning to be bayesian without supervision,” in *Advances in neural information processing systems*, 2007, pp. 1145–1152.
- [11] T. Le, R. Chartrand, and T. J. Asaki, “A variational approach to reconstructing images corrupted by poisson noise,” *Journal of mathematical imaging and vision*, vol. 27, no. 3, pp. 257–263, 2007.
- [12] R. Nowak and E. D. Kolaczyk, “A bayesian multiscale framework for poisson inverse problems,” in *1999 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings. ICASSP99 (Cat. No. 99CH36258)*, vol. 3. IEEE, 1999, pp. 1741–1744.

- [13] K. Timmerman and R. D. Nowak, “Multiscale modeling and estimation of poisson processes with application to photon-limited imaging,” *IEEE Transactions on Information Theory*, vol. 45, no. 3, pp. 846–842, 1999.
- [14] K. Hirakawa and P. J. Wolfe, “Skellam shrinkage: Wavelet-based intensity estimation for inhomogeneous poisson data,” *IEEE Transactions on Information Theory*, vol. 58, no. 2, pp. 1080–1093, 2011.
- [15] F. Luisier, T. Blu, and M. Unser, “Undecimated haar thresholding for poisson intensity estimation,” in *2010 IEEE International Conference on Image Processing*. IEEE, 2010, pp. 1697–1700.
- [16] H. Talbot, H. Phelippeau, M. Akil, and S. Bara, “Efficient poisson denoising for photography,” in *2009 16th IEEE International Conference on Image Processing (ICIP)*. IEEE, 2009, pp. 3881–3884.
- [17] M. A. Figueiredo and J. M. Bioucas-Dias, “Restoration of poissonian images using alternating direction optimization,” *IEEE transactions on Image Processing*, vol. 19, no. 12, pp. 3133–3145, 2010.
- [18] S. Setzer, G. Steidl, and T. Teuber, “Deblurring poissonian images by split bregman techniques,” *Journal of Visual Communication and Image Representation*, vol. 21, no. 3, pp. 193–199, 2010.
- [19] P. G. Kumar and R. Ranjan Sahay, “Low rank poisson denoising (lrpd): A low rank approach using split bregman algorithm for poisson noise removal from images,” in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2019.
- [20] Y. Zhang, Y. Zhu, E. Nichols, Q. Wang, S. Zhang, C. Smith, and S. Howard, “A poisson-gaussian denoising dataset with real fluorescence microscopy images,” in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.
- [21] J. Zhang and K. Hirakawa, “Improved denoising via poisson mixture modeling of image sensor noise,” *IEEE Transactions on Image Processing*, vol. 26, no. 4, pp. 1565–1578, April 2017.
- [22] Y. Zhang, G. Wang, and J. Xu, “Parameter estimation of signal-dependent random noise in cmos/ccd image sensor based on numerical characteristic of mixed poisson noise samples,” *Sensors*, vol. 18, no. 7, p. 2276, 2018.
- [23] K. H. Xiaodan Jin, “Approximations to camera sensor noise,” 2013. [Online]. Available: <https://doi.org/10.1117/12.2019212>
- [24] K. E. Timmermann and R. D. Nowak, “Multiscale modeling and estimation of poisson processes with application to photon-limited imaging,” *IEEE Transactions on Information Theory*, vol. 45, no. 3, pp. 846–862, Apr 1999.

- [25] K. Hirakawa and P. J. Wolfe, “Skellam shrinkage: Wavelet-based intensity estimation for inhomogeneous poisson data,” *IEEE Transactions on Information Theory*, vol. 58, no. 2, pp. 1080–1093, Feb 2012.
- [26] —, “Skellam shrinkage: Wavelet-based intensity estimation for inhomogeneous poisson data,” *IEEE Transactions on Information Theory*, vol. 58, no. 2, pp. 1080–1093, 2012.
- [27] G. Finlayson and G. Schaefer, “Hue that is invariant to brightness and gamma,” in *Proc. British Machine Vision Conference*, 2001, pp. 303–312.
- [28] X. Jin, Z. Xu, and K. Hirakawa, “Noise parameter estimation for poisson corrupted images using variance stabilization transforms,” *IEEE Transactions on Image Processing*, vol. 23, no. 3, pp. 1329–1339, March 2014.
- [29] G. Torricelli, F. Argenti, and L. Alparone, “Modelling and assessment of signal-dependent noise for image de-noising,” in *2002 11th European Signal Processing Conference*, Sep. 2002, pp. 1–4.
- [30] F. Argenti, G. Torricelli, and L. Alparone, “Mmse filtering of generalised signal-dependent noise in spatial and shift-invariant wavelet domains,” *Signal Processing*, vol. 86, no. 8, pp. 2056–2066, 2006.
- [31] I. Amerini, R. Caldelli, V. Cappellini, F. Picchioni, and A. Piva, “Analysis of denoising filters for photo response non uniformity noise extraction in source camera identification,” in *2009 16th International Conference on Digital Signal Processing*, July 2009, pp. 1–7.
- [32] S. Pyatykh and J. Hesser, “Image sensor noise parameter estimation by variance stabilization and normality assessment,” *IEEE Transactions on Image Processing*, vol. 23, no. 9, pp. 3990–3998, Sep. 2014.
- [33] A. Bosco, A. Bruna, D. Giacalone, S. Battiato, and R. Rizzo, “Signal dependent raw image denoising using sensor noise characterization via multiple acquisitions,” vol. 7537, 01 2010, p. 753705.
- [34] K. Hirakawa, F. Baqai, and P. J. Wolfe, “Wavelet-based poisson rate estimation using the skellam distribution,” vol. 7246. International Society for Optics and Photonics, 2009, p. 72460R.
- [35] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, “Image denoising by sparse 3-d transform-domain collaborative filtering. image processing, iee transactions on 16 (8), pp. 2080-2095,” 2007.
- [36] D. Pascale, “Rgb coordinates of the macbeth colorchecker,” *The BabelColor Company*, vol. 6, 2006.

- [37] M. Rakhshanfar and M. A. Amer, “Estimation of gaussian, poissonian–gaussian, and processed visual noise and its level function,” *IEEE Transactions on Image Processing*, vol. 25, no. 9, pp. 4172–4185, 2016.
- [38] Y. Li, Z. Li, K. Wei, W. Xiong, J. Yu, and B. Qi, “Noise estimation for image sensor based on local entropy and median absolute deviation,” *Sensors*, vol. 19, no. 2, p. 339, 2019.
- [39] H. Bay, T. Tuytelaars, and L. Van Gool, “Surf: Speeded up robust features,” in *European conference on computer vision*. Springer, 2006, pp. 404–417.
- [40] M. A. Fischler and R. C. Bolles, “Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography,” *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981.
- [41] C. Sutour, J. Aujol, and C. Deledalle, “Automatic estimation of the noise level function for adaptive blind denoising,” in *2016 24th European Signal Processing Conference (EUSIPCO)*, Aug 2016, pp. 76–80.
- [42] D. Pascale, “Rgb coordinates of the macbeth color checker,” pp. 1–16, Jan. 2006.

## APPENDIX A

### Appendix for Color Image Denoising

#### A.1 Proof of (4.58)

Let  $\{\mathbb{B}(n)|Z = z_m\} \sim \mathcal{N}\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \Sigma_m\right)$ , we have:

$$\begin{aligned}
 & e^{A_1(n)B_1(n)+A_2(n)B_2(n)} \cdot Pr[\mathbb{B}(n)|Z = z_m] \\
 &= \frac{e^{-\frac{(B_1^2(n)-2\sigma_{m1}^2 A_1(n)B_1(n))}{2\sigma_{m1}^2}} e^{-\frac{(B_2^2(n)-2\sigma_{m2}^2 A_2(n)B_2(n))}{2\sigma_{m2}^2}}}{2\pi \sqrt{|\Sigma_m|}} \\
 &= \frac{\exp\left\{-\frac{1}{2}\begin{bmatrix} B_1(n) - A_1(n)\sigma_{m1}^2 \\ B_2(n) - A_2(n)\sigma_{m2}^2 \end{bmatrix}^T \Sigma_0^{-1} \begin{bmatrix} B_1(n) - A_1(n)\sigma_{m1}^2 \\ B_2(n) - A_2(n)\sigma_{m2}^2 \end{bmatrix}\right\}}{2\pi \sqrt{|\Sigma_m|}} \\
 &\quad \times e^{\frac{A_1^2(n)\sigma_{m1}^2}{2}} e^{\frac{A_2^2(n)\sigma_{m2}^2}{2}}.
 \end{aligned} \tag{A.1}$$

Note that the first portion of the above product describe a new 2D Gaussian distribution function with mean vector  $\begin{bmatrix} A_1(n)\sigma_{m1}^2 \\ A_2(n)\sigma_{m2}^2 \end{bmatrix}$  and covariance matrix  $\Sigma_m$ . Use similar method for the derivation of (4.52), we compute the following conditional expectation as:

$$\begin{aligned}
 & \mathbb{E}(\mathbb{B}(n)|\mathbf{F}_{\mathcal{L}}, \mathbb{A}(n), Z = z_0) \\
 &= \int \mathbb{B}(n) Pr[\mathbb{B}(n)|\mathbf{F}_{\mathcal{L}}, \mathbb{A}(n), Z = z_0] d\mathbb{B}(n) \\
 &= \frac{\int \mathbb{B}(n) e^{A_1(n)B_1(n)+A_2(n)B_2(n)} \cdot Pr[\mathbb{B}(n)|Z = z_m] d\mathbb{B}(n)}{\int e^{A_1(n)B_1(n)+A_2(n)B_2(n)} \cdot Pr[\mathbb{B}(n)|Z = z_m] d\mathbb{B}(n)}. \\
 &= \begin{bmatrix} A_1(n)\sigma_{m1}^2 \\ A_2(n)\sigma_{m2}^2 \end{bmatrix}
 \end{aligned} \tag{A.2}$$

Thus, the proof of (4.58) is done by substituting (A.2) into (4.57).

## A.2 Proof of (4.59)

Note that  $\{Z = z_m\}$  does not depend on the value of  $\mathbf{F}_{\mathcal{L}}$ . Use Bayes rule, we rewrite the conditional probability  $Pr[Z = z_0 | \mathbf{F}_{\mathcal{L}}, \mathbb{A}(n)]$  as follow:

$$\begin{aligned} & Pr[Z = z_0 | \mathbf{F}_{\mathcal{L}}, \mathbb{A}(n)] \\ &= \frac{Pr[\mathbb{A}(n) | \mathbf{F}_{\mathcal{L}}, Z = z_0] Pr[Z = z_0]}{\sum_{m=0}^1 Pr[\mathbb{A}(n) | \mathbf{F}_{\mathcal{L}}, Z = z_m] Pr[Z = z_m]} \end{aligned} \quad (\text{A.3})$$

By the definition of marginal density function, we have

$$\begin{aligned} & Pr[\mathbb{A}(n) | \mathbf{F}_{\mathcal{L}}, Z = z_m] \\ &= \int Pr[\mathbb{A}(n) | \mathbb{B}(n), \mathbf{F}_{\mathcal{L}}, Z = z_m] Pr[\mathbb{B} | \mathbf{F}_{\mathcal{L}}, Z = z_m] d\mathbb{B}(n) \\ &= \int Pr[\mathbb{A}(n) | \mathbb{B}(n), \mathbf{F}_{\mathcal{L}}] Pr[\mathbb{B} | Z = z_m] d\mathbb{B}(n) \end{aligned} \quad (\text{A.4})$$

Substituting (A.4) into (A.3), we have  $Pr[Z = z_0 | \mathbf{F}_{\mathcal{L}}, \mathbb{A}(n)] :=$

$$\frac{\int Pr[\mathbb{A}(n) | \mathbb{B}(n), \mathbf{F}_{\mathcal{L}}] Pr[\mathbb{B} | Z = z_0] d\mathbb{B}(n) Pr[Z = z_0]}{\sum_{m=0}^1 \int Pr[\mathbb{A}(n) | \mathbb{B}(n), \mathbf{F}_{\mathcal{L}}] Pr[\mathbb{B} | Z = z_m] d\mathbb{B}(n) Pr[Z = z_m]} \quad (\text{A.5})$$

Recalling (4.49) and (A.1), (A.5) can be re-written as:

$$\begin{aligned} & \frac{\int e^{\mathbb{A}^{\top}(n)\mathbb{B}(n)} Pr[\mathbb{B} | Z = z_0] d\mathbb{B}(n) Pr[Z = z_0]}{\sum_{m=0}^1 \int e^{\mathbb{A}^{\top}(n)\mathbb{B}(n)} Pr[\mathbb{B} | Z = z_m] d\mathbb{B}(n) Pr[Z = z_m]} \\ &= \frac{\rho \cdot e^{\frac{\mathbb{A}^{\top}(n)\Sigma_0\mathbb{A}(n)}{2}}}{\rho \cdot e^{\frac{\mathbb{A}^{\top}(n)\Sigma_0\mathbb{A}(n)}{2}} + (1 - \rho) \cdot e^{\frac{\mathbb{A}^{\top}(n)\Sigma_1\mathbb{A}(n)}{2}}}. \end{aligned} \quad (\text{A.6})$$

Pay attention that the constant terms in (4.49) are cancelled in (A.6).