Improved compression, demosaicking and denoising for digital still cameras

Lian, Naixiang

2007

Lian, N. (2007). Improved compression, demosaicking and denoising for digital still cameras. Doctoral thesis, Nanyang Technological University, Singapore.

<https://hdl.handle.net/10356/4607>

<https://doi.org/10.32657/10356/4607>

Nanyang Technological University

Downloaded on 26 Apr 2023 20:24:24 SGT

Improved Compression, Demosaicking and Denoising for Digital Still Cameras

Lian Naixiang

School of Electrical & Electronic Engineering

A thesis submitted to the Nanyang Technological University
in fulfillment of the requirement for the degree of
Doctor of Philosophy

2007

Statement of Originality

I hereby certify that the work embodied in this thesis is the result of original research done by me and has not been submitted for a higher degree to any other University or Institute.

.....

Date

.....

Lian Naixiang

*To my beloved wife,
To my parents,
for their love and support.*

Acknowledgements

This thesis could not have been completed without the help of a number of people. Here, I would like to express my sincere, heartfelt gratitude to those who have figured most prominently over the last four years of my postgraduate study.

I am greatly indebted to my supervisor, Associate Professor Tan Yap Peng, who gave me strong support and sincere encouragement throughout the tough times of the research. Within these four years, I learned from him not only specialized knowledge, but also the spirit to explore science. It is his excellent guidance and persistent encouragement that push me to complete the work of this thesis. Although sometimes appearing to be too tough towards me, his driving is the impetus for me to carry out such a challenging project and to face many unexpected difficulties during these years.

My sincere gratitude also goes to Assistant Professor Vitali Zagorodnov for his invaluable suggestion and discussion during the past two years. It is his friendliness and encouragement that give me confidence to improve my writing skill and explore related research fields. Without his assistance, I would have taken more time to complete the thesis.

I gratefully acknowledge the financial support and the excellent research

facilities and resources from School of EEE, NTU, Singapore. I would like to thank all of my friends, graduate students, the faculties, and technicians in School of EEE, NTU. All of my experiences and interaction with these people have helped me and my work, and have made my study in NTU worthwhile and memorable.

I should thank my parents for their large doses of encouragement and considerations. They never understand what I have been doing, “image processing”, which they might regard as using a brush to process pictures. Nevertheless they choose to support me without reservations, even after they haven’t seen me for over two years’ time. My thanks also go to my sister and her husband (my brother-in-law), who take care of our parents for these years, which allows me to concentrate on my ideal.

I am especially grateful to my beloved Xiaoying, whose love, encouragement and support are my power to pursue my dream. She is the angel, making my hard-working more meaningful, and my life joyful and hopeful.

Contents

Acknowledgements	1
Contents	3
Summary	9
List of Figures	11
List of Tables	18
List of Abbreviations	24
1 Introduction	26
1.1 Motivation	26
1.2 Background on Compression Error Modeling	30
1.3 Background on Demosaicking and Denoising	33
1.3.1 Demosaicking	33
1.3.2 Denoising	35
1.3.3 Using high-frequency image content correlation for better performance	38

1.4	Main Contributions and Organization of the Thesis	44
I	Compression	48
2	Reversing Demosaicking and Compression in Digital Still Cam-	
	eras	49
2.1	Introduction	49
2.2	Previous Work and Motivation	53
2.2.1	Alternative processing methods	53
2.2.2	Motivation and notation	55
2.3	Error Models of Processing Chains	57
2.3.1	Conventional: Demosaicking-Compression	59
2.3.2	Alternative: Compression-Demosaicking	64
2.3.3	Quick comparison of processing chains	68
2.4	Compression Error Model	69
2.4.1	Full-color image compression	70
2.4.2	CFA image compression	72
2.4.3	Evaluation of the compression error models	74
2.5	Performance Comparison of Two Processing Chains	75
2.6	Suitable Reference Images for Performance Evaluation	83
2.7	Conclusions	86
3	Error Inhomogeneity of Wavelet Based Compression	88
3.1	Introduction	88

3.2	Supporting Experiments and Motivation	90
3.3	Error Modeling for Even and Odd Signal Samples	94
3.4	Error Inhomogeneity Prediction	98
3.4.1	Case 1: Uniform uncorrelated quantization errors	99
3.4.2	Case 2: Nonuniform uncorrelated quantization errors	101
3.4.3	Case 3: Nonuniform and correlated quantization error	103
3.5	Balanced Wavelet Design	106
3.5.1	Balancing 9-7 wavelet	108
3.5.2	Balancing 9-3 wavelet	112
3.6	Trade-off between Overall Performance and Error Inhomogeneity	113
3.7	Applications	115
3.8	Conclusions	118
II	Demosaicking	120
4	Adaptive Filtering for Color Filter Array Demosaicking	121
4.1	Introduction	121
4.2	Key Observations about CFA Demosaicking	123
4.2.1	Similarity of inter-color high-frequency content	123
4.2.2	Importance of different high-frequency components	124
4.3	Frequency Analysis of Bayer CFA Pattern	126
4.4	Better Luminance Filtering at Green Samples	131
4.5	Luminance Filtering at Red/Blue Samples	133

4.6	Summary of Proposed Method	139
4.7	Experimental Results	140
4.8	Discussion	145
4.8.1	Computational complexity	145
4.8.2	Modified CFA patterns	148
4.9	Conclusions	149
III	Denoising	151
5	Wavelet and Min-Cut Analysis for Image Denoising	152
5.1	Introduction	152
5.2	Min-Cut in Wavelet Denoising	153
5.3	Incorporating Intra-Scale Correlations	158
5.4	Experimental Results	159
5.5	Conclusions	160
6	Optimal Color Space Projection for Image Denoising	165
6.1	Introduction	165
6.2	Estimate Error of Wavelet-Based Denoising	167
6.3	The Optimal Color Space Projection	169
6.4	Estimation of Noise Covariances	174
6.4.1	Estimation of uncorrelated noise variances	175
6.4.2	Estimation of correlated noise covariances	177
6.5	Denoising of Color-Difference Images	179

6.6	Summary of the Proposed Denoising Method	182
6.7	Experimental Results and Discussion	183
6.7.1	Performance contributions of proposed algorithm compo- nents	183
6.7.2	OCP with other grayscale wavelet-based denoising ap- proaches	185
6.7.3	Comparison to existing state-of-the-art methods	186
6.8	Conclusions	196
7	Vector Estimation of Wavelet Coefficients for Video Denoising	198
7.1	Introduction	198
7.2	Motivation and Previous Work	200
7.3	Vector Estimator	203
7.4	Vector Hidden Markov Tree	206
7.5	Motion Alignment and Change Compensation	210
7.5.1	Motion alignment	210
7.5.2	Change compensation	211
7.6	Experimental Results	212
7.6.1	Contribution of inter-color and inter-frame correlations .	213
7.6.2	Contribution of signal covariance matrix estimation . . .	216
7.6.3	Contribution of motion alignment and change compensation	217
7.6.4	Comparison with existing methods	217
7.6.5	Comparison of VEWC and OCP denoising	219
7.7	Conclusions	224

8	Conclusions and Future Work	225
A	Error Inhomogeneity in Multi-Level Wavelet Decompositions	228
B	Optimal Color Projection for Arbitrary Signal Covariance Matrix	231
C	Estimation Error of VEWC and OCP Denoising Approaches	234
	Author's Publications	237
	Bibliography	239

Summary

Digital Still Cameras (DSC) processing consists of a sequence of several basic operations. Since most DSCs acquire color imagery with a color filter array (CFA), capturing only one color sample for each pixel, the other two colors need to be interpolated. This interpolation process is referred to as demosaicking. The reconstructed full-color image then undergoes a set of correction operations that may include color and gamma correction, white balancing, and denoising. Finally, the image is compressed and stored in the memory. Potential improvements of DSC processing chains are generally restricted by camera's limited computational resources. Also, once built, the camera's processing algorithms are difficult to upgrade. Hence, a recent trend is to move most DSC operations (demosaicking, denoising) out of the camera to a computer (alternative processing chain).

At this point it is not clear whether this trend will continue to develop as several questions remain unanswered. For example, since the demosaicking is to be performed outside the camera, the compression will have to be done directly on the CFA data. However, traditional JPEG compression standard and a new wavelet-based standard JPEG2000 were developed for full color images, and it is necessary to know whether there will be any performance drawbacks if these algorithms are applied to CFA data. In this thesis we develop analytical models to analyze the performance of conventional and alternative processing chains, and show that the alternative chain can outperform the conventional

one at a large range of compression ratios. We also show that compression error of JPEG2000 compression has inhomogeneous structure, and suggest a way to reduce this inhomogeneity.

For demosaicking and denoising, computer-based processing opens up possibilities to explore advanced and computationally intensive algorithms capable of achieving best-possible quality. Generally, demosaicking and denoising can be considered as low pass filtering processes, and hence might lead to image oversmoothing due to loss of high frequency content. In this thesis we show that performance of demosaicking and denoising can be substantially improved by incorporating high frequency content correlations, which exist between color channels, video frames, and even scales of wavelet transformation. For demosaicking, we propose an adaptive filtering approach based on the strong inter-color correlations. For denoising, we propose three wavelet-based approaches that show how incorporation of inter- and intra-scale, inter-color, and inter-frame correlations can improve image quality.

List of Figures

1.1	(a) Bayer color filter array pattern and sub-sampled (b) red, (c) green, (d) blue planes.	27
1.2	Block diagrams of DSC image processing chains: (a) conventional, demosaicking followed by compression in DSC and decompression in end device; (b) alternative, compression in DSC and decompression followed by demosaicking in end device. . . .	28
1.3	(a) Original color image and the demosaicked one by (b) bilinear, (c) Hamilton [1], and (d) AP [2].	34
1.4	Wavelet decomposition of “Lena”.	36
1.5	(a) Noisy image “Lena” and the denoised one by (b) wiener filtering, (c) thresholding of wavelet coefficients [3], and (d) uHMT [4].	37
1.6	Scatter plots of detail wavelet coefficients. The coordinates of each point are the values of the R, G, and B components.	41
1.7	Basis vectors of a typical YCbCr space projection, $\mathbf{v} = (0.299, 0.587, 0.114)^T$, $\mathbf{d}_1 = (-0.147, -0.289, 0.436)^T$ and $\mathbf{d}_2 = (0.615, -0.515, -0.100)^T$ [5].	42
1.8	(a) Original and (b) noisy color image, and denoised results in (c) RGB and (d) YCbCr color space using uHMT algorithm. . .	43

2.1	Simplified block diagrams of DSC image processing chains: (a) conventional, and (b) alternative.	52
2.2	Treating a CFA image as grayscale.	54
2.3	Color transformation of a 2×2 Bayer CFA unit.	54
2.4	Y plane and its 45° rotation.	55
2.5	Cropped region of Bike image: (a) After bilinear demosaicking; (b) After bilinear demosaicking and JPEG (10:1) compression.	57
2.6	Block diagrams of error models.	58
2.7	Relationship between Ω^2 and ξ_s^2 for 15 test images. Each curve corresponds to one test image.	60
2.8	Relationship between Ω^2 and ζ^2 for 15 test images. Each curve corresponds to one test image.	61
2.9	Predicted total error Ω^2 of conventional chain using simplified model (2.4).	63
2.10	Relationship between Ψ_o^2 and ζ^2 for 15 test images. Each curve corresponds to one test image.	65
2.11	Relationship between Ψ_o^2 and ξ_o^2 for 15 test images. Each curve corresponds to one test image.	66
2.12	Predicted the error of alternative chain using simple model (2.7).	67
2.13	Average logarithm of compression error variance <i>versus</i> bitrate.	70
2.14	Ratio between the compression error variances of the full-color and CFA images.	75
2.15	Predicted <i>versus</i> actual performance comparison between conventional and alternative processing chains. The actual results are computed from the average of the last 15 images in Figure 2.16.	78

2.16	Test images (referred to as Image 1 to Image 30, enumerated from left-to-right and top-to-bottom).	79
2.17	Performance comparison between the conventional processing chain using bilinear and the alternative one using iECI.	80
2.18	Comparison of the images reconstructed from a real CFA image by using <i>bilinear</i> demosaicking and JPEG2000 compression (10:1).	81
2.19	Comparison of the reconstructed images after demosaicking and JPEG2000 compression (10:1) of CFA data downsampled from a test image.	82
2.20	Relative performance of alternative and conventional processing chains for different reference images: left column, demosaicked image; right column, original full color images.	85
3.1	Image division into four sub-images.	89
3.2	Test images used in the experiments	91
3.3	Comparison of sub-images 0 and 3 for different wavelets and bitrates. (a-b) 9-7 wavelet, 0.50bpp, (c-d) 9-7 wavelet, 0.20bpp, (e-f) 9-3 wavelet, 0.20bpp.	93
3.4	Block scheme of wavelet analysis and synthesis operations	95
3.5	Filtering process (*) in wavelet synthesis.	97
3.6	Ratio r between quantization error variances	102
3.7	Error homogeneity as a function of r for several wavelets: (a) predicted using (3.11) (b) evaluated by averaging over 19 test images	103
3.8	Correlation between quantization errors of neighboring coefficients	104

3.9	Predicted inhomogeneity under various assumptions on the statistics of quantization errors. Here we extend our 1D prediction equations to 2D by applying them to both horizontal and vertical directions and take the reconstruction error in the LL subband (due to multi-scale decomposition) as the quantization error.	105
3.10	Reconstructed sub-images (bitrate 0.20bpp): (a,b) original 9-7 wavelet, (c, d) balanced 9-7 wavelet.	111
3.11	Plots of coding gain (CG) and error inhomogeneity (EI) of modified 9-7 wavelet as functions of γ	115
3.12	Compression image using (a) original and (b) balanced 9-7 wavelets (bitrate = 0.25bpp, SPIHT)	116
3.13	The downsize example of compressed sub-images after low-pass filtering (bitrate = 0.25bpp, JPEG2000, 9-7 wavelet)	118
3.14	Downsampling using “nearest neighbor” method of the Matlab function “imresize”	119
4.1	An image and its spectrum.	125
4.2	(b)-(d) images with removed frequency components outside centered circle, as shown in (a); (f)-(h) images with removed frequency components outside a region elongated along both axes, as shown in (e). Note that the ideal property of the filter leads to the ringing errors near the boundary of these images [6], which is not our concern.	126
4.3	Fourier transforms of (a) full-resolution color component, (b) quincuncial sampled green component, and (c) rectangular sampled red/blue component.	128

4.4	(a) Fourier transform and (b) a typical Fourier spectrum of a CFA image	130
4.5	(a) The passband and (b) Fourier transform of the low-pass filter in [7].	130
4.6	The optimal values of r_2 for 30 test images.	133
4.7	MSE of the estimated green pixel luminance <i>vs.</i> (a) r_2 (b) filter size.	134
4.8	Proposed low-pass filter for green pixel luminance: (a) passband, (b) spectrum representation, and (c) impulse response.	135
4.9	(a) Luminance instead of green in CFA, (b) reference neighboring samples, and (c) the adaptive interpolation filter.	135
4.10	(a) color-difference image $Cr = L - R_I$ and its spectrum of regions with (b) diagonal edge, (c) horizontal edges, and (d) vertical edges.	136
4.11	(a) Bilinear interpolation filter and (b,c) two examples of adaptive filter for horizontal and diagonal edges, respectively. The side or corner passband is due to the replication of spectrum. . .	137
4.12	Differences between original Cr component and the estimated ones by (a) bilinear interpolation, and (b) proposed adaptive filter. The dark pixels are with intensity levels larger than 8. . . .	138
4.13	Original and demosaicked results of a cropped region in Image 3.	143
4.14	Original and demosaicked results of a cropped region in Image 9.	143
4.15	Original and demosaicked results of a cropped region in Image 27.	144
4.16	Original and demosaicked results of a cropped region in Image 27.	145

4.17	Comparison of AHD and proposed demosaicking methods in texture regions	146
4.18	Modified Bayer CFA patterns with (a) red and (b) blue in quincunx lattice.	148
5.1	Graphs exploit (a) inter-scale correlation only or (b) both inter and intra-scale correlations.	157
5.2	Test images (Image 1 to Image 24, enumerated from left-to-right and top-to-bottom).	159
5.3	Denoised images obtained by the <i>wiener2</i> , uHMT and WMCA methods. The standard deviations of the noise are $\{0.1, 0.1, 0.1\}$, added to the R, G, and B components, respectively.	163
5.4	Denoised images obtained by the <i>wiener2</i> , uHMT and WMCA methods. The standard deviations of the noise are $\{0.15, 0.10, 0.05\}$ added to the R, G, and B components, respectively.	164
6.1	Estimation error variance as a function of noise variance.	169
6.2	(a) Noiseless luminance. (b) Denoised luminance using $\mathbf{v} = [0.299, 0.587, 0.114]^T$. (c) Denoised luminance using $\mathbf{v}_{opt}$	174
6.3	(a) Color-difference image. The states of wavelet coefficients (b) after uHMT classification, (c) after uHMT classification with Fine/Coarse-Scale separation, and (d) in coarse scale after uHMT classification with Fine/Coarse-Scale separation; Gray and white pixels correspond to small and large coefficient states, respectively.	181
6.4	Comparison of LSAI denoising in different color spaces on the Parrot image corrupted by uncorrelated, nonuniform noise.	187

6.5	Comparison of different denoising approaches on the Fence image corrupted by uncorrelated, nonuniform noise.	193
6.6	Comparison of different denoising approaches on the Parrot image corrupted by uncorrelated, nonuniform noise.	194
6.7	Comparison of different denoising approaches on the Bike image corrupted by correlated, nonuniform noise.	195
6.8	Real image and the OCP-denoising result.	196
7.1	Vector of coefficients (for $K = 1$) in video sequence.	204
7.2	Two neighboring frames in test video “News”.	212
7.3	Test videos.	214
7.4	CPSNR performance comparison along video frame.	219
7.5	Comparison of denoised Frame 3 of “MOBILE”.	220
7.6	Comparison of denoised Frame 55 of “MOBILE”.	221
7.7	Comparison of denoised Frame 41 of “GARDEN”.	222

List of Tables

1.1	Pairwise correlation of wavelet coefficients in the three finest scales.	40
2.1	CPSNR performance (in dB) of the two processing chain with bilinear demosaicking and JPEG/JPEG2000 compression (10:1).	56
2.2	Estimated values of a_c^2 and ρ_a in the model of conventional processing chain [Eq. (2.3)].	62
2.3	Estimated values of b_d^2 and ρ_b in the error model of alternative processing chain [Eq. (2.6)].	67
2.4	Similarity between $\rho_0 \cdot \nu_{s_1, s_2}$ and its exact values.	71
2.5	Superiority of processing chains (conventional or alternative). . .	76
3.1	Comparison of sub-image PSNR for 19 grayscale test images (bitrate = 0.50 bpp, SPIHT, 9-7 wavelet)	92
3.2	Comparison of sub-image PSNR for 5 color images (bitrate = 0.50 bpp, JPEG2000, 9-7 wavelet)	94
3.3	Comparison of sub-image PSNR for 19 grayscale test images (bitrate = 0.50 bpp, SPIHT, 9-3 wavelet)	95
3.4	Comparison of sub-image PSNR for 19 grayscale test images (bitrate = 0.50 bpp, JPEG2000, 5-3 wavelet)	96

3.5	Comparison of sub-image PSNR in different bit-rates for “Barbara” (SPIHT, 9-7 wavelet)	97
3.6	Norms of even/odd samples of synthesis wavelet filters	100
3.7	Predicted (3.6) and empirically evaluated (over 19 test images) error inhomogeneity (in dB) for several popular wavelets	100
3.8	Values of $\hat{V}_i(1)$ and $\check{V}_i(1)$ wavelet parameters	104
3.9	Balanced 9-7 wavelet filter coefficients	109
3.10	PSNR performance and inhomogeneity (in dB) for original and new balanced 9-7 wavelets (0.50 bpp, SPIHT)	110
3.11	Balanced 9-3 wavelet filter coefficients	113
3.12	PSNR performance and inhomogeneity (in dB) for original and new balanced 9-3 wavelets (0.50 bpp, SPIHT)	114
3.13	Trade-off between PSNR performance and error inhomogeneity — EI (in dB) by wavelet design (0.50 bpp and SPIHT)	116
3.14	PSNR performance of compressed sub-images after low-pass filtering (0.50 bpp, JPEG2000, 9-7 wavelet)	117
4.1	MSE performance of the estimated green pixel luminance by using different filters.	133
4.2	MSE performance of the estimated red/blue pixel luminance by different filters	139
4.3	PSNR performance (in dB) comparison of bilinear and adaptive filter in Step 3.	140

4.4	PSNR performance (in dB) comparison (images 1-15). For each image the red, green and blue components are listed in the 1st, 2nd and 3rd rows. The best PSNR of each row is shown in bold and underlined.	141
4.5	PSNR performance (in dB) comparison (images 16-30). For each image the red, green and blue components are listed in the 1st, 2nd and 3rd rows. The best PSNR of each row is shown in bold and underlined.	142
4.6	Comparison of demosaicking complexity.	147
4.7	Comparison of PSNR performance (in dB) by applying FS approach [7] and the proposed method on original Bayer CFA pattern and two modified ones. The best PSNR in the comparison is shown in bold and underlined.	149
5.1	Performance comparison of the <i>wiener2</i> , uHMT, and WMCA methods in YCbCr space for $\sigma_n = \{0.1, 0.1, 0.1\}$. The PSNR results (in dB) of the Y, Cb and Cr components are listed in the 1st, 2nd and 3rd rows, respectively, of each image.	161
5.2	Performance comparison of the <i>wiener2</i> , uHMT, and WMCA methods in YCbCr space for $\sigma_n = \{0.15, 0.10, 0.05\}$. The PSNR results (in dB) of the Y, Cb and Cr components are listed in the 1st, 2nd and 3rd rows, respectively, of each image.	162

6.1	PSNR performance (in dB) of Y, Cb and Cr using the uHMT method in the case of uncorrelated noise. For uniform noise, the noise standard deviations added to {R, G, B} are equal to $\sigma_n = \{0.1, 0.1, 0.1\}$; for nonuniform noise, $\sigma_n = \{0.15, 0.10, 0.05\}$. These two types of uncorrelated noise remain the same for all other experiments in the chapter.	170
6.2	Comparison of normalized errors of Donoho's and the proposed noise estimators. The two types of uncorrelated noise are the same as that in Table 6.1. For correlated noise, the noise standard deviations added to {R, G, B} and the correlation coefficients for {R&G, R&B, G&B} are equal to $\sigma_n = \{0.1, 0.1, 0.1\}$ and $\{0.4, 0.2, 0.3\}$, respectively, in the case of uniform noise, and equal to $\sigma_n = \{0.05, 0.15, 0.10\}$ and $\{0.3, 0.3, 0.3\}$ in the case of nonuniform noise. These two types of correlated noise remain the same for all other experiments in the chapter.	176
6.3	PSNR performance (in dB) of uHMT on color-difference images with and without Fine/Coarse-scale separation in the case of uncorrelated noise	182
6.4	Comparison of CPSNR performance (in dB) of uHMT in RGB, YCbCr, and the proposed OCP color spaces	184
6.5	Comparison of CPSNR performance (in dB) of Donoho's and the proposed noise estimators	184
6.6	Comparison of CPSNR performance (in dB) with or without separating Fine/Coarse scales (F/C) in denoising color-difference images.	184
6.7	CPSNR performance (in dB) by applying OCP in different approaches for grayscale images.	186

6.8	CPSNR performance (in dB) of <i>wiener2</i> , LSAI, MML, and the proposed OCP methods for uncorrelated noise.	188
6.9	CPSNR performance (in dB) of <i>wiener2</i> , LSAI, MML, and the proposed OCP methods for correlated noise.	189
6.10	Average PSNR performance (in dB) of <i>wiener2</i> , LSAI, MML, and the proposed OCP methods.	190
6.11	CPSNR performance (in dB) in the edge (E) and smooth (S) image regions for uncorrelated noise.	191
6.12	CPSNR performance (in dB) in the edge (E) and smooth (S) image regions for correlated noise.	192
7.1	Average inter-frame correlation of wavelet coefficients of video “MOBILE”.	201
7.2	Average correlations of wavelet coefficients in two states.	209
7.3	Test videos.	213
7.4	Average CPSNR performance for denoising videos using vectors $\mathbf{x}$ with different dimension M	215
7.5	Influence of conditional signal covariance matrix update on the CPSNR performance.	216
7.6	Influence of motion alignment (MA) and change compensation (CC).	217
7.7	Comparison of average CPSNR performance.	218
7.8	CPSNR performance (in dB) of the proposed OCP and VEWC denoising methods.	223

A.1 PSNR performance (in dB) at pixel locations 0-3 (9-7 wavelet, SPIHT).	230
B.1 CPSNR performance (in dB) using the perfect correlation assumption versus arbitrary signal covariance matrix. The covariance matrix of the noise is assumed known.	233

List of Abbreviations

ABBREVIATIONS	FULL EXPRESSIONS
AFD	Adaptive filtering demosaicking
AHD	Adaptive homogeneity directed demosaicking
AP	Alternating projections
CB	Chromaticity-brightness
CC	Change compensation
CCD	Charge-coupled devices
CFA	Color filter array
CG	Coding gain
CWDT	Combined wavelet domain and temporal video denoising
DSC	Digital still camera
ECI	Effective color interpolation
EI	Error inhomogeneity
EZW	Embedded zerotree wavelet
F/C	Fine/Coarse scale separation
FS	Frequency selection
GSM	Gaussian scale mixtures
HSV	Hue-Saturation-Value
iECI	improved effective color interpolation

LHBF	Lagrange half-band filter
LSAI	Local spatial activity indicator
LSE	Least square estimate
HMT	Hidden Markov tree
MA	Motion alignment
MML	Multispectral maximum likelihood
MRF	Markov random field
OCP	Optimal color projection
PCSD	Primary consistent soft decision
PDE	Partial differential equation
PR	Perfect reconstruction
TAF	Temporal averaging filter
Thr	Thresholding
SA	Successive approximation
SPIHT	Set partitioning of images in hierarchical tree
uHMT	universal hidden Markov tree
VEWC	Vector estimation of wavelet coefficients
WMCA	Wavelet and minimum-cut analysis

Chapter 1

Introduction

1.1 Motivation

Images can convey much more information than sounds or symbols, hence a popular saying — “a picture is worth a thousand words”. Storing images in a digital format allows easy transmission and conversion of this information between various displaying devices, such as computer, TV, PDA, mobile phone, etc. Combined with the decreasing cost of digital cameras and camcorders, improvements in the computer processing power, storage capacity, image display and the speed of data communication, digital imaging technology slowly changes various aspects of our lives.

In developing digital still cameras (DSCs), the main concerns are cost, size, image quality, and operational/power efficiency [8]. To reduce the cost and size, most DSCs capture images using a single sensor overlaid with a color filter

array (CFA), sampling one of the three primary/complementary color values at each pixel. The most popular CFA, called the Bayer pattern [9], comprises two green, one blue and one red samples arranged in a 2×2 square block, as shown in Figure 1.1. Demosaicking, a color interpolation process, is then applied to estimate the two missing color values at each pixel location [1, 2, 10–17] and obtain full-color images. For good image quality, sophisticated demosaicking algorithms are usually applied, followed by image compression to reduce the data size [8].

Figure 1.1: (a) Bayer color filter array pattern and sub-sampled (b) red, (c) green, (d) blue planes.

The conventional DSC image processing chain is shown in Figure 1.2(a). The CFA data from the sensor is first demosaicked to estimate the missing color values. The resultant image is then modified to make sure that its colors

match the colors of the original scene when displayed on a computer monitor. These modifications generally include white balance, color correction, and gamma correction [18,19]. White balance removes color tint of an image to make sure white objects appear white and other colors appear right. Color correction transforms the CFA sensor color space to a suitable display color space (e.g., sRGB [20]). Gamma correction adjusts the image intensity to compensate the non-linearity of cathode-ray-tube (CRT) displays. After noise reduction and image sharpening, the image is compressed for storage or transmission [18,19].

Figure 1.2: Block diagrams of DSC image processing chains: (a) conventional, demosaicking followed by compression in DSC and decompression in end device; (b) alternative, compression in DSC and decompression followed by demosaicking in end device.

A downside of conventional DSC processing is the limited DSC computational resources, which prevents application of advanced demosaicking, compression, and noise reduction algorithms. One way to overcome this is to move these operations to computers, for example, by exporting the raw CFA data. This approach is gaining popularity in professional photography, but has not been extended to general consumer market due to high storage space requirements.

As such, one can consider an alternative processing chain by reversing the demosaicking and compression process [21–26], as shown in Figure 1.2(b). Here the output of the CFA sensor is compressed before converting it to a full color image. This allows at least some operations, such as demosaicking, color adjustments (white balance, color correction, and gamma correction), and denoising to be done directly on the computer, rather than inside the camera.

However, at this point it is not clear whether this kind of processing chain will continue to develop, as several questions remain unanswered. For example, since the demosaicking is to be performed outside the camera, the compression will have to be done directly on the CFA data. However, traditional JPEG compression standard and a new wavelet-based standard JPEG2000 were developed for full color images, and might suffer performance drop if applied to CFA data. To answer this question a detailed modeling of the whole processing chain error (both compression and demosaicking) is necessary.

For demosaicking and denoising, computer-based processing opens up possibilities to explore advanced and computationally intensive algorithms capable of achieving best-possible quality. Generally, demosaicking and denoising can be considered as low pass filtering processes, and hence might lead to image oversmoothing due to loss of high-frequency content. So the main question is how to better preserve high-frequency content, for example, by incorporating various image correlations.

While applications of demosaicking work developed in the thesis are limited to DSC processing chains, the denoising approaches developed can be extended to a variety of other applications, such as denoising of medical images of various modalities [27]. Medical images are usually acquired using specialized tech-

niques, such magnetic resonance (MR) imaging or X-Ray, which may lead to different noise statistics from that of the DSCs' images. Even though our denoising approaches rely heavily on the assumptions of specific noise statistics, they can be easily extended to more general cases.

1.2 Background on Compression Error Modeling

Exchangeable image file format (EXIF) [18], a standard for most current DSCs, uses JPEG compression, which is based on discrete cosine transform (DCT) [28]. JPEG belongs to the family of so-called transform-based coders, where the image is decomposed into a set of frequencies (or subbands), and then each frequency is quantized using variable bitrate allocation. The number of bits allocated to each subband depends on the portion of the total signal energy in this subband. For example, JPEG allocates more bits to lower frequencies since they contain larger portion of signal energy.

Wavelet based compression, a newer transform-based coding technique, has recently emerged as a powerful alternative to the traditional DCT-based compression, due to the excellent energy compaction capabilities of wavelet transform on piecewise smooth signals, such as images. Shapiro's pioneering Embedded Zerotree Wavelet (EZW) compression algorithm [29] and later Said and Pearlman's Set Partitioning of Images in Hierarchical Trees (SPIHT) algorithm [30] have both demonstrated that wavelet based compression can achieve much better PSNR performance and visual quality compared to JPEG compression. Many improvements that followed [31–37] finally led to a new still

image compression standard JPEG2000 [38], likely to be incorporated into the future DSCs [18].

While superiority of JPEG2000 has been confirmed experimentally, the actual difference in compression performance depends on the compression ratio used. For example, empirical results show that JPEG2000 substantially outperforms JPEG at high compression ratios, while the advantage at smaller compression ratios is less obvious. However, such empirical observations are difficult to use when one needs a detailed prediction of the performance as a function of bitrate, image content, etc. Such cases may require detailed mathematical modeling of compression performance, as exemplified by Part I of this thesis.

To predict the compression performance we need to consider how energy is distributed among the subbands depending on the transform used and image statistics, such as inter-pixel correlations. In their pioneering work, Katto and Yasuda [39] derived an expression for the error variance ξ_I^2 of wavelet based compression¹ corresponding to the optimal bitrate allocation:

$$\xi_I^2 = \epsilon^2 \sigma_I^2 2^{-2R_I} \cdot \prod_{k=0}^{K-1} (A_k B_k)^{\alpha_k} \quad (1.1)$$

Here R_I is the total compression bitrate, σ_I^2 is the signal variance, $\alpha_k = \frac{N_k}{N}$ (N_k and N are the number of wavelet coefficients in the k th subband and the total number of coefficients respectively), and ϵ is a constant depending on the signal. The parameter A_k represents the relative proportion of signal energy in the k th subband, i.e., $\sigma_k^2 = A_k \sigma_I^2$, and B_k governs the contribution of the k th subband quantization error variance σ_{qk}^2 to the total quantization error, i.e.,

¹The theory can be straightforwardly extended to JPEG compression by grouping DCT coefficients and treating them as subbands. For example, in JPEG compression all zero frequency coefficients are assigned the same bitrate and can be thought of as one subband.

$\xi_I^2 = \sum_{k=0}^{K-1} B_k \sigma_{qk}^2$. The product $A_k B_k$ in (1.1) is closely related to the coding gain [40, 41] defined as

$$\mathcal{G}_I = \frac{1}{\prod_{k=0}^{K-1} (A_k B_k)^{\alpha_k}} \quad (1.2)$$

For 1D signals modeled using Markov chains, A_k and B_k are related to the analysis (transform) h_k and synthesis (inverse) g_k filters and inter-pixel correlation ρ [40, 41] as:

$$A_k = \sum_i \sum_j h_k(i) h_k(j) \rho_{|i-j|} \quad (1.3)$$

$$B_k = \sum_i g_k^2(i) \quad (1.4)$$

The derivation of (1.1) is based on modeling the k th subband quantization error as $\sigma_{qk}^2 = \epsilon^2 \sigma_k^2 2^{-2R_k}$ [42], where R_k is the subband bitrate. This quantization model requires an assumption of high bitrate R_k [43] and may perform poorly at low bitrates. The model (1.1) has also several other important limitations. It is derived for 1D scalar signals and hence is not directly applicable to color images. Extension from 1D to 2D can be done straightforwardly by modeling grayscale images as 2D Markov random processes [40, 41, 44–47]. But the extension to vector signals (colors) is not so obvious, as the existing color image compression techniques are typically based on color transformation [28, 38]. Modeling the CFA image compression performance is even more difficult due to many ways to incorporate inter-pixel correlations [21–26].

1.3 Background on Demosaicking and Denoising

Despite the apparent dissimilarity between demosaicking and denoising, they share a similar goal of predicting intensity of image samples from missing/noisy data (image restoration). As such, both can benefit from inherent correlations present inside the images and videos; see Parts II and III of the thesis.

1.3.1 Demosaicking

In the demosaicking process, the missing color components (values) are interpolated from the neighboring samples. Bilinear interpolation (or bilinear demosaicking) uses a linear function of two or four same color samples located at the spatial proximity of the missing color value [48]. Due to its simplicity, bilinear demosaicking has often been adopted as a reference method in the comparison between the conventional and alternative processing chains [21, 23, 24, 26]. However, its performance can be rather poor, especially in regions rich of image details and edges. Bicubic interpolation can improve the performance moderately but at a substantial computational cost.

Estimating the missing color values is possible because of substantial correlation between neighboring pixel intensities (inter-pixel correlation). Bilinear and bicubic demosaicking use isotropic neighborhoods, which may lead to oversmoothing of edges. In real images the inter-pixel correlations are anisotropic — the correlation is larger along the edges and smaller across the edges. Several researchers suggested using this fact to improve demosaicking quality [1, 10, 12–14, 49, 50]. For example, Hamilton [1] interpolated the missing

Figure 1.3: (a) Original color image and the demosaicked one by (b) bilinear, (c) Hamilton [1], and (d) AP [2].

color values (components) according to the edge direction. This leads to sharper edges and fewer color artifacts [compare Figures 1.3(b) and (c)].

Much insight about demosaicking can also be obtained from Fourier domain analysis. Sub-sampling the original full color image to obtain CFA data results in periodic replication of Fourier spectrum of the original image. If the sampling frequency is less than twice the bandwidth of the original image, the successive periods of the spectrum will overlap, resulting in aliasing artifacts [51, 52]. The bilinear and bicubic interpolation techniques remove the aliasing by eliminating the overlapped high-frequency components. This elimination is similar to low-pass filtering [10, 12], which could degrade the resultant image quality, especially

in the more sparsely sampled red and blue planes. A good demosaicking should remove the aliasing artifacts while at the same time preserving as much high-frequency information as possible.

1.3.2 Denoising

There are two main approaches to image denoising: spatial domain filtering and transform domain filtering. The spatial domain filtering exploits the correlations between neighboring pixels to remove the noise; particular techniques include Median and Wiener filtering [51]. Similar to demosaicking, spatial filtering relies on the inherent inter-pixel correlations. Denoising can also be done in the transformed domain, which has been shown to achieve better performance than spatial filtering, as exemplified by a popular denoising technique based on thresholding of wavelet coefficients [4, 53–60]. The main reason behind the success of wavelet thresholding denoising is its better adaptation to anisotropic nature of inter-pixel correlations.

Wavelet transform decomposes the image into multi-scale high-frequency (detail) coefficients and one-subband low-frequency coefficients, as shown in Figure 1.4. For a typical grayscale image, wavelet coefficients in the high frequency subbands can be roughly classified into two categories, Noise and Signal [53], or Small and Large [4]. Small coefficients correspond to smooth regions of the image; large coefficients occur in the edge regions. The main idea of wavelet denoising is to modify the coefficients according to this classification. For example, in a noisy image the coefficients in smooth regions contain primarily noise. Hence such coefficients should be reduced or even put to zero. On the other hand, the proportion of noise in the large coefficients near edge regions is

Figure 1.4: Wavelet decomposition of “Lena”.

usually small. Hence these coefficients should be preserved.

Let x and $\tilde{x}$ be the wavelet coefficient and its noisy version, *i.e.*, $\tilde{x} = x + n$, where n is the noise which is assumed to be zero mean and independent from the signal. Let $q \in \{\mathcal{S}, \mathcal{L}\}$ be the state, $\mathcal{S}$ for small and $\mathcal{L}$ for large. Hard assignment based denoising is a two-step procedure, where the coefficients are first classified into $\mathcal{S}$ or $\mathcal{L}$ states and then shrunk using the following rule:

$$\hat{x} = \beta(q)\tilde{x} \quad (1.5)$$

where $\beta(q) \in [0, 1]$ is the shrinkage coefficient. Typically $\beta(\mathcal{S})$ is close to 0 while $\beta(\mathcal{L})$ is close to 1.

Hard assignment doesn't take into account the uncertainty associated with the state estimation. Better performance can be achieved by computing for each coefficient the probability that it is affected or unaffected by noise. The manipulation of the coefficient is then based on this probability. In [4, 55] the wavelet coefficients are modeled as a mixture of two zero-mean Gaussian

Figure 1.5: (a) Noisy image “Lena” and the denoised one by (b) wiener filtering, (c) thresholding of wavelet coefficients [3], and (d) uHMT [4].

distributions with variances $\sigma_{x|q}^2 + \sigma_n^2$, where $\sigma_{x|q}^2$ and σ_n^2 are the signal and noise variances, respectively. Let $p(q|\tilde{x})$ be the conditional state probabilities. It follows that the Minimum Mean Square Error (MMSE) estimate of x is [55]

$$\hat{x} = p(\mathcal{S}|\tilde{x})\beta(\mathcal{S})\tilde{x} + p(\mathcal{L}|\tilde{x})\beta(\mathcal{L})\tilde{x} \quad (1.6)$$

where

$$\beta(q)\tilde{x} = \frac{\sigma_{x|q}^2}{\sigma_{x|q}^2 + \sigma_n^2}\tilde{x}, \quad q \in \{\mathcal{S}, \mathcal{L}\} \quad (1.7)$$

are conditional MMSE estimates. Portilla *et al.* [61] used a similar framework, but modeled the states via a continuous scale parameter.

Shown in Figure 1.5 is the comparison between Wiener filtering [51], wavelet thresholding [3], and the uHMT approach [4]. The uHMT approach using soft assignment obtains the best perceptual performance.

1.3.3 Using high-frequency image content correlation for better performance

In addition to inter-pixel correlations, there are a number of other types of correlations that are present in the images. These additional correlations usually correspond to high-frequency image content, as opposed to inter-pixel correlations that are essentially low-frequency content correlations.

For example, it is well known that detail wavelet coefficients are correlated across scales or in local neighborhood [62]. Exploiting intra- or inter-scale correlations can improve the classification of wavelet coefficients and hence the performance of the hard assignment-based denoising. Bao and Zhang [60] showed this by applying shrinkage to inter-scale products of wavelet coefficients. Similarly, for the soft assignment-based denoising of the form (1.6), estimation performance depends on conditional state probabilities, which also can be improved by exploiting inter- and intra-scale correlations. This can be done via constraining the wavelet coefficient by its local neighborhood [61], or joint estimation on a Hidden Markov Tree (HMT) [4, 55], or using Markov Random Field modeling [56].

In color images there is substantial correlation between RGB color compo-

nents (inter-color correlations). This correlation is especially strong (correlation coefficient varies from 0.98 to 1 for most images) for high-frequency content, as has been shown by Gunturk *et al.* [2] on the example of detail wavelet coefficients. We replicated Gunturk *et al.*'s results for Daubechies-4 wavelet² [63] and the three finest scales in Table 1.1. These results demonstrate that the correlations remain strong even for coarser scales. Here LH, HL and HH are vertical, horizontal and diagonal detail wavelet subbands, respectively. Overall, the correlation in the HH subband appears to be the strongest among all the subbands.

In general, perfect correlation between two random variables does not necessarily suggest their equality. However, in typical color images the high-frequency coefficients of color components are not only correlated but also *approximately equal* to each other. To demonstrate this we plotted the 3D scatter diagrams of high-frequency wavelet coefficients in Figure 1.6. Here, the coordinates of each point are the magnitudes of the R, G, and B detail wavelet coefficients taken from the same image location of Image 1 in Figure 2.16. The points are compactly clustered along the line of vector $\mathbf{1} = (1, 1, 1)^T$. Similar scatter diagrams can be obtained for all other test images. These results are similar to Hel-Or's directional derivatives scatter plots [64], but are extended to all detail wavelet coefficients and two finest scales.

Inter-color correlations have been widely exploited in CFA demosaicking [1,2,16,17,49,64,65]. Gunturk *et al.* [2] proposed to substitute detail wavelet coefficients of green component into missing red/blue components by using an alternating projection (AP) method. Typical CFA demosaicking methods use color differences to reduce the high-frequency content of the images, as in effec-

²Its mother wavelet filter is (0.4830, 0.8365, 0.2241, -0.1294).

Table 1.1: Pairwise correlation of wavelet coefficients in the three finest scales.

Color Image	Scale	R & G			R & B			G & B		
		LH	HL	HH	LH	HL	HH	LH	HL	HH
1	1	0.983	0.993	0.995	0.974	0.991	0.994	0.991	0.996	0.994
	2	0.978	0.991	0.997	0.969	0.988	0.996	0.992	0.997	0.997
	3	0.947	0.968	0.989	0.934	0.954	0.983	0.995	0.995	0.995
2	1	0.811	0.879	0.969	0.731	0.833	0.963	0.979	0.989	0.980
	2	0.687	0.834	0.974	0.540	0.755	0.968	0.973	0.985	0.992
	3	0.256	0.584	0.857	0.012	0.364	0.822	0.958	0.957	0.985
3	1	0.942	0.964	0.957	0.880	0.929	0.937	0.887	0.950	0.939
	2	0.951	0.960	0.974	0.871	0.903	0.952	0.877	0.929	0.962
	3	0.917	0.893	0.913	0.737	0.631	0.747	0.767	0.741	0.826
4	1	0.892	0.967	0.979	0.906	0.970	0.980	0.989	0.993	0.983
	2	0.878	0.829	0.960	0.895	0.859	0.966	0.991	0.985	0.993
	3	0.654	0.637	0.836	0.711	0.698	0.864	0.980	0.977	0.984
5	1	0.985	0.982	0.989	0.966	0.965	0.984	0.977	0.971	0.979
	2	0.980	0.975	0.995	0.952	0.950	0.990	0.975	0.967	0.991
	3	0.945	0.939	0.970	0.850	0.868	0.929	0.930	0.924	0.963
6	1	0.989	0.994	0.992	0.960	0.982	0.973	0.972	0.989	0.979
	2	0.988	0.992	0.996	0.959	0.980	0.986	0.973	0.991	0.991
	3	0.979	0.967	0.985	0.952	0.945	0.965	0.972	0.981	0.986
7	1	0.964	0.969	0.975	0.943	0.946	0.963	0.962	0.963	0.975
	2	0.963	0.973	0.988	0.944	0.954	0.981	0.965	0.970	0.987
	3	0.913	0.944	0.950	0.882	0.901	0.924	0.929	0.934	0.945
8	1	0.982	0.990	0.988	0.975	0.982	0.983	0.990	0.990	0.986
	2	0.980	0.990	0.993	0.974	0.982	0.990	0.993	0.993	0.994
	3	0.960	0.975	0.975	0.938	0.954	0.962	0.990	0.990	0.990
9	1	0.986	0.995	0.981	0.963	0.991	0.976	0.973	0.991	0.977
	2	0.990	0.973	0.989	0.969	0.923	0.976	0.982	0.956	0.981
	3	0.982	0.944	0.974	0.929	0.816	0.932	0.961	0.898	0.963
10	1	0.978	0.985	0.975	0.967	0.977	0.971	0.976	0.978	0.965
	2	0.978	0.973	0.988	0.962	0.934	0.983	0.982	0.959	0.984
	3	0.950	0.975	0.914	0.883	0.911	0.901	0.964	0.951	0.946
11	1	0.977	0.978	0.987	0.964	0.970	0.981	0.990	0.991	0.991
	2	0.971	0.971	0.990	0.953	0.960	0.985	0.990	0.993	0.996
	3	0.919	0.895	0.953	0.875	0.855	0.933	0.981	0.980	0.990
12	1	0.949	0.979	0.957	0.939	0.974	0.964	0.953	0.979	0.953
	2	0.948	0.977	0.974	0.938	0.978	0.978	0.961	0.987	0.979
	3	0.917	0.950	0.882	0.898	0.919	0.889	0.956	0.959	0.946
Average	1	0.953	0.973	0.979	0.931	0.959	0.972	0.970	0.982	0.975
	2	0.941	0.953	0.985	0.911	0.930	0.979	0.971	0.976	0.987
	3	0.862	0.889	0.933	0.800	0.818	0.904	0.949	0.941	0.960

Figure 1.6: Scatter plots of detail wavelet coefficients. The coordinates of each point are the values of the R, G, and B components.

tive color interpolation (ECI) [16] or in improved ECI (iECI) [17] methods. The scatter plots in Figure 1.7 explain why these approaches have been so effective. In color difference spaces, such as YCbCr, the color difference vectors $\mathbf{d}_1$ and $\mathbf{d}_2$ are orthogonal to signal vector $\mathbf{1} = (1, 1, 1)^T$. Hence the projection onto these vectors reduces the detailed wavelet coefficients, making the respective image components smooth and hence easier to demosaick. Figure 1.3(d) uses the AP demosaicking algorithm to demonstrate the advantage of incorporating inter-color correlations in demosaicking.

The strong inter-color correlation can also improve the performance of denoising. Scheunders [66] used inter-color products of wavelet coefficients to

Figure 1.7: Basis vectors of a typical YCbCr space projection, $\mathbf{v} = (0.299, 0.587, 0.114)^T$, $\mathbf{d}_1 = (-0.147, -0.289, 0.436)^T$ and $\mathbf{d}_2 = (0.615, -0.515, -0.100)^T$ [5].

improve the denoising performance. Pižurica and Philips [67] obtained superior to Scheunders' performance by updating local activity parameter of their grayscale image denoiser with the average value of the wavelet coefficients at the same image location. Scheunders and Driesen [68] treated wavelet coefficients of color components as a signal vector and applied the linear minimum mean squared error (LMMSE) estimate to this vector by taking into account the inter-color correlations. Transforming images into a suitable color space is also a popular technique in denoising to exploit inter-color correlations. For example, Ben-Shahar [69] used Hue-Saturation-Value (HSV) color space. Chan *et al.* [70] have found that the chromaticity-brightness (CB) decomposition, another color space transformation, can lead to better restored images than denoising in RGB or HSV space. The CB color space has also been adopted in several partial differential equation (PDE) based denoising methods [71, 72]. As shown in Figure 1.7, the color difference vectors $\mathbf{d}_1$ and $\mathbf{d}_2$ are orthogonal

to signal vector $\mathbf{1} = (1, 1, 1)^T$, which indicates that the luminance/color difference space is also good space for denoising; see an example in Figure 1.8. More details can be found in Chapter 6.

Figure 1.8: (a) Original and (b) noisy color image, and denoised results in (c) RGB and (d) YCbCr color space using uHMT algorithm.

In case of videos, there is a correlation between the high-frequency content of neighboring frames (inter-frame correlations), which can be used to improve video denoising; see Chapter 7 for more details. With so many different types of correlations present it is important to know what is the best way to incorporate them to improve demosaicking and denoising. This is the topic of Part II of the thesis.

1.4 Main Contributions and Organization of the Thesis

The thesis is organized in two parts. Part I consists of Chapters 2 and 3, dealing with modeling and analysis of the error of the DSC processing chains and wavelet-based compression. Part II (Chapter 4) and Part III (Chapters 5, 6, and 7) propose improved demosaicking and denoising algorithms, respectively, through incorporation of high-frequency content correlations. Finally, the conclusions and recommendations for future research are outlined in Chapter 8.

The main contributions of the thesis are:

1. **Comparative analysis of the error in DSC image processing chains (Chapter 2).** In the conventional processing chain of single sensor DSCs, the images are captured with CFA and the CFA samples are demosaicked into a full-color image before compression. To avoid additional data redundancy created by the demosaicking process, an alternative processing chain has been proposed to move the compression process before the demosaicking. Recent empirical studies have shown that the alternative chain can outperform the conventional one in terms of image quality at low compression ratios. To provide theoretically sound basis for such conclusion, we propose analytical models for the reconstruction errors of the two processing chains. The developed models confirm the results of existing empirical studies and provide better understanding of DSC processing chains. The modeling also allows performance predictions for more advanced compression and demosaicking methods, thus providing important cues for future development in this area.

2. Error inhomogeneity of wavelet-based compression (Chapter 3).

Despite the popularity of wavelet-based image compression, its shortcoming of having error inhomogeneity, namely the error that is different for even and odd pixel locations, has not been previously analyzed and formally addressed. The difference can be substantial, up to 3.4-dB PSNR for some images and compression ratios. In this chapter, we show that the error inhomogeneity is caused by asymmetrical filtering of quantization errors after the upsampling step in wavelet synthesis process. Nonuniformity and correlation of quantization errors can also contribute to this error inhomogeneity, albeit to a smaller degree. In addition to explaining the source of error inhomogeneity, the developed model also allows predicting its amount for a given wavelet. Furthermore, we show how to redesign wavelet filters to reduce the error inhomogeneity at a cost of a small reduction in the overall PSNR performance. For applications that are sensitive to PSNR degradation, we also show how to design wavelet filters that can gradually trade-off PSNR performance for reduced error inhomogeneity.

3. Adaptive filtering demosaicking (Chapter 4). A good demosaicking method should preserve the high-frequency information of imagery as much as possible, since such information is essential for visual quality of demosaicked images. Inspired by a recent work of demosaicking filtering and exploiting the similarity of high-frequency content in three color components, our frequency analysis of CFA samples indicates that filtering a CFA image can better preserve high frequencies than filtering each single color component. This motivates us to design an efficient filter for estimating the luminance at green pixels of the CFA image, and devise an

adaptive filtering approach to estimating the luminance at red and blue pixels. Experimental results on simulated CFA images as well as raw CFA data verify that the proposed method outperforms the existing state-of-the-art methods both visually and in terms of peak signal-to-noise ratio, at a notably lower computational cost.

4. **Incorporating high-frequency content correlations into denoising (Chapters 5 - 7).** We propose several novel denoising approaches to incorporate various types of high-frequency correlations inherently present in images and videos (inter- and intra-scale, inter-color, and inter-frame).

- **Wavelet and minimum cut analysis:** Wavelet thresholding has proven to be an efficient edge-preserving denoising method for grayscale images, especially when it exploits the inter-scale correlations of wavelet coefficients. Intra-scale correlations can further improve the denoising performance, but the gain for grayscale images is generally small. In this chapter we demonstrate that the gain can become substantial for smooth images, such as color-difference components in YCbCr space. Here we propose a new denoising method, based on minimum cut algorithm, to exploit both the inter and intra-scale correlations of wavelet coefficients. The proposed method achieves up to 5-dB gain in peak signal-to-noise ratio for color-difference images and leads to fewer visual color artifacts.
- **Optimal color projection:** Recent work has shown that exploiting strong correlation between high-frequency content of different color components can improve the denoising performance. In this chapter we propose to incorporate the high correlation and similarity of

inter-color detail wavelet coefficients via an optimal luminance/color-difference space projection. Experimental results confirm that performing denoising on the projected color components yields superior to existing solutions denoising performance, both in PSNR and visual quality sense. We also develop a novel approach to estimate directly from the noisy image data the image and noise statistics, which are required to determine the optimal projection.

- **Vector estimate of wavelet coefficient:** Wavelet-based image denoising can be extended to a video by applying it to each video frame independently. The denoising performance can be improved by exploiting inter-frame correlations, for example, using appropriate temporal filtering. However, fixed temporal filters might not perform sufficiently well due to their inability to cope with the variability of inter-frame correlations across the video. We propose a vector extension of popular Hidden Markov Tree modeling that flexibly exploits the color and frame dependency of wavelet coefficients. Experimental results confirm that the vector estimator of wavelet coefficients yields denoising performance superior to that of existing solutions, both in CPSNR and visual quality sense. We also show that simple motion alignment and change compensation algorithms can enhance the inter-frame correlations and further improve the denoising performance.

Part I

Compression

Chapter 2

Reversing Demosaicking and Compression in Digital Still Cameras

2.1 Introduction

The DSC (Digital Still Camera) alternative chain, which is based on reversing demosaicking and compression [Figure 1.2(b)], can reduce the camera processing time by off-loading the demosaicking/denoising process to the end device, such as a computer, personal digital assistant (PDA) or printer with the direct printing capabilities. This simplifies the hardware architecture and can reduce the cost and power consumption of DSCs, which is especially beneficial to cameras situated on mobile phones and PDAs. It also allows different demosaicking

methods to be applied to suit the needs of applications. For example, computer image viewer can use the most advanced and complex demosaicking methods to suit high resolution computer displays, because computational cost is not an issue in this case [73]. On the other hand, if the image is to be viewed on a PDA screen, simpler demosaicking methods will suffice.

A disadvantage of the alternative chain is the increased implementation complexity of compression operation. Since the compression is done directly on Color Filter Array (CFA) images this might require customization to the CFA pattern (i.e., non-rectangular green sample lattice) [24–26]. It is also unclear where the alternative processing chain can achieve better quality compared to the standard. The conclusion of existing studies [21–26] is that alternative chain has better quality at least for sufficiently low compression ratios. However, empirical nature of these studies makes their results relevant to only existing compression and demosaicking algorithms. It is not clear whether the alternative chain will remain superior to conventional one when more advanced demosaicking and compression algorithms become available in the future.

In this chapter we analyze the conventional and alternative processing chains by detailed theoretical modeling of their reconstruction errors. We first introduce the previous work and motivation in Section 2.2. Next we study the interaction between compression and demosaicking errors in Section 2.3. Based on the study we propose a linear model for the error variance of each processing chain in terms of the variances of the compression and demosaicking errors. Using the proposed model, we show that the relative performance of the two processing chains primarily depends on two factors: the difference between the compression error variances of full-color and CFA images, and attenuation of

the demosaicking error in the conventional chain.

To compare the compression errors, we model their variances by extending existing 1D model for the errors of subband based compression [39] to 2D full-color and CFA images in Section 2.4. The derived models allow prediction of the compression error variances based on the available bitrate, inter-pixel and inter-color correlations. Full-color images have higher inter-pixel and inter-color correlations, but are three times larger in data size. We shall show that this leads to CFA image compression being better than full-color image compression at low compression ratios, and vice versa at high compression ratios. Moreover, the models developed allow us to predict the “break-point” compression ratio at which the overall reconstruction errors of the conventional and alternative chains coincide. The accuracy of the prediction is confirmed by experimental results.

Based on the compression error model and the predicted interaction between the compression and demosaicking errors, we compare the performances of the conventional and alternative processing chains for several compression and demosaicking methods in Section 2.5. Our conclusions are that the alternative processing chain can outperform the conventional one at compression ratios lower than 8:1-10:1 for JPEG compression, and lower than 16:1 for JPEG2000 compression. Since typical DSC compression ratios for visually lossless encoding are usually lower than 12:1 (or more than 2 bits per pixel) [8], this means that the alternative processing chain can be used in DSCs without loss of image quality.

Note that existing empirical studies [21–26] simplify the DSC processing chains by ignoring color adjustment steps (white balance, color correction, and

gamma correction). The reasons include 1) operating on real CFA images makes it difficult to quantify the performance difference due to the lack of ground truth, and 2) most existing demosaicking and compression algorithms are customized for gamma-corrected images. Hence most researchers simulate CFA images by downsampling the existing color and gamma corrected full-color images, which is equivalent to moving the color adjustment steps in front of the DSC processing chains, as shown in Figure 2.1. Since one of goals of our error modeling is to explain the existing empirical studies, we develop the theory based on conventional simplified framework (Figure 2.1)¹.

Figure 2.1: Simplified block diagrams of DSC image processing chains: (a) conventional, and (b) alternative.

¹To justify the usefulness of our analysis based on the simplified chains, we have also considered the following two processing chains, which are actually feasible, perform better by using the existing demosaicking and compression algorithms, but are more difficult to model due to the non-linearity of gamma correction. Our empirical study has shown that the relative performance obtained by these two feasible chains is consistent with that of the simplified chains.

In addition to providing better understanding of the DSC processing chains, the models developed can offer useful suggestions for the future development in this area. For example, some researchers used the demosaicked images, and not the original ones, as the references for performance evaluation [24,25]. Our analysis in Section 2.6 shows that such evaluation may lead to wrong conclusions about the relative performance of the two processing chains. The models also explain an interesting observation that applying JPEG compression to bilinearly demosaicked images actually reduces the overall reconstruction error, which contradicts our intuitive expectations (see Section 2.3.1).

2.2 Previous Work and Motivation

2.2.1 Alternative processing methods

In the alternative processing chain the demosaicking and compression are reversed, as shown in Figure 1.2. As a result, the CFA samples are compressed directly, without conversion to a full-color image. Standard grayscale compression techniques can be applied directly to CFA images by discarding pixel color labels, as shown in Figure 2.2. This, however, often leads to poor compression performance due to the artificial discontinuities from interlaced color components, which results in a rapidly changing or “busy” image [21]. Performance can be slightly improved by using JPEG2000 compression, which is capable of achieving an acceptable performance even in the presence of artificial discontinuities [23]. It has also been suggested to compress CFA color components separately [21,22]. However, exploiting inter-color correlations can substantially improve the compression performance.

Figure 2.2: Treating a CFA image as grayscale.

Figure 2.3: Color transformation of a 2×2 Bayer CFA unit.

Existing image compression standards use color transformation to decorrelate the color components [28,38]. Similar transformation can be used to remove artificial discontinuities and obtain better CFA compression, as was shown by Lee and Ortega [24], Parrein *et al.* [25], and Koh *et al.* [26]. The four color values in each 2×2 Bayer unit are converted to two luminance values (Y_1 and Y_2) and two chrominance values (U and V), as shown in Figure 2.3. The resultant chrominance values reside in a rectangular lattice with size four times smaller than the size of CFA, while the luminance values populate a quincunx lattice with half size of CFA, as shown in Figure 2.4(a).

Each chrominance plane (U or V) can be compressed using standard techniques. But the quincunx lattice of the luminance plane requires preprocessing.

Figure 2.4: Y plane and its 45° rotation.

Koh *et al.* [26] proposed to either split the quincunx lattice into two rectangular lattices Y_1 and Y_2 , or convolve it with a low-pass filter followed by column-wise down-sampling. To better exploit the correlation between Y_1 and Y_2 , Lee and Ortega [24], and Parrein *et al.* [25] proposed to rotate the quincunx lattice by 45°, converting it into a diamond shaped rectangular lattice, as shown in Figure 2.4(b).

2.2.2 Motivation and notation

Koh *et al.* [26] used JPEG compression in their alternative processing chain and found that it can outperform the conventional processing chain at low ($< 10:1$)² and high ($> 80:1$) compression ratios. Lee and Ortega [24] observed superiority of alternative processing with JPEG compression at low ($< 10:1$) compression ratios. Parrein *et al.* [25] used JPEG2000 and reached the same conclusion for compression ratios $< 15:1$. All these conclusions are based on empirical studies, making their results relevant to only existing compression and

²The sizes of full-color images are used as reference when evaluating the compression ratios.

demosaicking algorithms. Without a solid theoretical basis, it cannot be proven that the alternative chain will remain superior to conventional when advanced demosaicking and compression algorithms become available in the future.

In this chapter, we show that the superiority of the alternative processing chain at low compression ratios can be predicted by modeling compression error and its interaction with demosaicking error. That errors do interact can be easily seen from the fact that applying JPEG compression after bilinear demosaicking leads to overall error reduction, as shown in Table 2.1 and illustrated in Figure 2.5. Here CPSNR (color peak-signal-to-noise ratio) is defined as the average Mean Square Error (MSE) of the reconstructed color components, i.e.,

$$\text{CPSNR} = -10 \log\left(\frac{1}{3} \sum_{c \in \{R, G, B\}} \text{MSE}_c\right) \quad (2.1)$$

for RGB values normalized to $[0, 1]$. This would not happen if the errors were independent, since compression always leads to decrease in image quality. The models developed in the chapter explain this phenomenon in Section 2.3.1.

Table 2.1: CPSNR performance (in dB) of the two processing chain with bilinear demosaicking and JPEG/JPEG2000 compression (10:1).

Image	After demosaicking	Conventional		Alternative	
		JPEG	JPEG2000	JPEG	JPEG2000
Bike	23.44	25.18	23.27	22.85	23.27
Lighthouse	28.01	30.04	27.87	27.74	27.96
Sails	31.69	33.59	31.58	31.28	31.63

Figure 2.5: Cropped region of Bike image: (a) After bilinear demosaicking; (b) After bilinear demosaicking and JPEG (10:1) compression.

We use the following notation throughout the chapter:

$x, \bar{x}$	Original full-color and CFA images
$\delta x, \Delta x$	Compression and demosaicking errors
Ω^2	Error variance of the conventional processing chain
Ψ_o^2, Ψ_t^2	Error variances of the alternative processing chains
$\bar{\Omega}^2, \bar{\Psi}_o^2$	Error variances evaluated with demosaicked image as reference
ζ^2	Demosaicking error variance
$\xi_I^2, \xi_s^2, \xi_o^2$	Compression error variances of grayscale, original full-color, and CFA images
$\mathcal{G}_I, \mathcal{G}_s, \mathcal{G}_o$	Coding gains of grayscale, original full-color, and CFA images
a_c^2, b_d^2	Coefficients of error interaction models

2.3 Error Models of Processing Chains

The difference between the conventional and alternative processing chains lies in the reversal of the compression C and demosaicking D operations, as shown in Figure 2.6. Let x be the original (desired) full-color image, $\bar{x}$ be the CFA

(a) Conventional processing chain

(b) Alternative processing chain

Figure 2.6: Block diagrams of error models.

image, and

$$C(x) = x + \delta x$$

$$C(\bar{x}) = \bar{x} + \delta \bar{x}$$

$$D(\bar{x}) = \bar{x} + \Delta x$$

where δx and $\delta \bar{x}$ are the compression errors of x and $\bar{x}$, while Δx is the demosaicking error. We use index i to designate individual pixels and treat demosaicking and compression errors in each pixel as random variables. We assume the compression error to be stationary and let $\xi_s^2 \triangleq E(\delta^2 x_i)$ and $\xi_o^2 \triangleq E(\delta^2 \bar{x}_i)$ be the compression error variances for the full-color and CFA images respectively, where each expectation is taken over all possible pixel locations. On the other hand, the demosaicking error is clearly not stationary, because it is zero at the CFA sample locations. However, to simplify the analysis we approximate it stationary and represent the demosaicking error variance $\zeta^2 \triangleq E(\Delta^2 x_i)$.

Note that the errors generated by compression and demosaicking are correlated in general. Moreover, the subsequent operation can affect the error of the first. Hence the total (reconstruction) error variance of a processing chain is not equal to the sum of the error variances of individual operations. In what follows, we evaluate the total error variances of the two processing chains by

modeling the interaction between the compression and demosaicking errors.

2.3.1 Conventional: Demosaicking-Compression

From Figure 2.6(a), the error in the conventional processing chain can be expressed as

$$\begin{aligned} C[D(\bar{x})] - x &= C(x + \Delta x) - C(x) + C(x) - x \\ &= C(x + \Delta x) - C(x) + \delta x \end{aligned}$$

The error is a sum of two components: compression error of the original image δx and the difference between the compressed versions of the demosaicked and original images $C(x + \Delta x) - C(x)$. The total error variance is then

$$\Omega^2 = E\{[C(x_i + \Delta x_i) - C(x_i)]^2\} + E(\delta^2 x_i) + 2E\{[C(x_i + \Delta x_i) - C(x_i)]\delta x_i\}$$

Clearly, for a given compression method and bitrate, $E\{[C(x_i + \Delta x_i) - C(x_i)]^2\}$ depends on the demosaicking error variance $\zeta^2 \triangleq E(\Delta^2 x_i)$. Let $f(\zeta^2) = E\{[C(x_i + \Delta x_i) - C(x_i)]^2\}$, then $f(0) = 0$. Assuming that the demosaicking error is small, we can expand $f(\zeta^2)$ using Taylor series and preserve only the linear term, i.e. $f(\zeta^2) \approx a_c^2 \zeta^2$, where parameter $a_c^2 = E\{[\frac{\partial C(x_i)}{\partial x_i}]^2\}$ in general can depend on the compression technique and the bitrate used. Similarly, $E\{[C(x_i + \Delta x_i) - C(x_i)]\delta x_i\} \approx a_c E(\Delta x_i \delta x_i) = \rho_a a_c \xi_s \zeta$, where ρ_a is the correlation coefficient between two errors. Combining these approximations yields the following expression for the total error variance of

Figure 2.7: Relationship between Ω^2 and ξ_s^2 for 15 test images. Each curve corresponds to one test image.

the conventional processing chain:

$$\Omega^2 = \xi_s^2 + a_c^2 \zeta^2 + 2\rho_a a_c \xi_s \zeta \quad (2.3)$$

The model (2.3) is qualitatively supported by the experimental results shown in Figures 2.7 and 2.8. We plot the total error variance Ω^2 as a function of the compression error variance ξ_s^2 in Figure 2.7, and as a function of demosaicking error variance ζ^2 in Figure 2.8. The error variances Ω^2 , ξ_s^2 and ζ^2 are all estimated by Mean Square Error (MSE) over the entire image. We control ξ_s^2 by varying the compression ratio, and ζ^2 by using several existing algorithms with varying demosaicking performance. Each curve corresponds to one of the 15 test images.

From Figure 2.7, the observed curves are linear for sufficiently large ξ_s^2 , with a slope close to 1. This is expected, since according to (2.3), $\Omega^2 \approx \xi_s^2$ for

Figure 2.8: Relationship between Ω^2 and ζ^2 for 15 test images. Each curve corresponds to one test image.

large $\xi_s^2 \gg \zeta^2$. When ξ_s^2 is small, the other two terms in (2.3) become more significant and can break the linearity. This is especially obvious for the case of bilinear demosaicking with large demosaicking error variance ζ^2 . The curves in Figure 2.8 are also linear for large demosaicking errors and small compression errors, which agrees with the model $\Omega^2 \approx a_c^2 \zeta^2$. For large compression errors (compression ratio = 30:1), linearity breaks due to the increasing influence of the cross term in (2.3).

The curves in Figure 2.8 can be used to estimate the parameters a_c^2 and ρ_a . We used standard Matlab function “polyfit” to fit the curves to the model (2.3), and the results are shown in Table 2.2. We can provide a simple intuition for the observed values of a_c^2 . Demosaicking errors mostly occur near the edges and hence contain a lot of high frequency information. JPEG compression is poor at preserving these high frequencies and results in smaller demosaicking error variance ($a_c^2 < 1$). JPEG2000, however, can better preserve high-frequency

Table 2.2: Estimated values of a_c^2 and ρ_a in the model of conventional processing chain [Eq. (2.3)].

Compression ratio		5:1	10:1	15:1	30:1	60:1
JPEG	a_c^2	0.567	0.518	0.449	0.281	0.021
	ρ_a	0.176	0.061	0.008	-0.014	-0.054
JPEG2000	a_c^2	1.027	1.012	0.985	0.867	0.654
	ρ_a	-0.108	-0.118	-0.121	-0.109	-0.089

content, leading to $a_c^2 \approx 1$. However, for very large compression ratios, both techniques will suppress the high frequencies ($a_c^2 \rightarrow 0$). Also note that for small compression ratios (and ultimately lossless compression) a_c^2 should approach 1 for both compression techniques, since the total error will be equal to the demosaicking error.

Despite the dependence of a_c^2 on the compression ratio, it is surprising that it remains approximately constant for the a wide range of compression ratios 5:1 - 15:1 (JPEG) and 5:1 - 30:1 (JPEG2000), as shown in Table 2.2. We have also found that ρ_a is small, with amplitude not exceeding 0.1 on average, as shown in Table 2.2. Hence inside this range of compression ratios we can simplify the model (2.3) to the following:

$$\Omega^2 = \xi_s^2 + a_c^2 \zeta^2 \quad (2.4)$$

where $a_c^2 = 0.5$ for JPEG compression and $a_c^2 = 1$ for JPEG2000. Figure 2.9 shows that such a simplified model provides sufficiently good fit to the experimental results, except for the bilinear demosaicking. The error of bilinear demosaicking is very large (five times larger than Hamilton's, see Figure 2.9), and hence amplifies the effect of decreasing a_c^2 . On the other hand, the decreasing trend of a_c^2 is deemphasized in advanced demosaicking methods.

Although the simple model (2.4) may overestimate the total error of con-

Figure 2.9: Predicted total error Ω^2 of conventional chain using simplified model (2.4).

ventional chain, it is sufficient to explain the phenomenon in Table 2.1. The error after demosaicking and compression, $\Omega^2 \leq \xi_s^2 + a_c^2 \zeta^2$, can become smaller than the error after demosaicking, namely ζ^2 , if $\xi_s^2 < (1 - a_c^2) \zeta^2$. This inequality can be achieved for $a_c^2 < 1$ and sufficiently large demosaicking error, as is in the case of JPEG compression and bilinear demosaicking. The phenomenon is not observed in JPEG2000 even at low compression ratios (see Table 2.1), since $a_c^2 \approx 1$ (Table 2.2).

2.3.2 Alternative: Compression-Demosaicking

From Figure 2.6(b), the error in the alternative processing chain can be expressed as

$$\begin{aligned} D[C(\bar{x})] - x &= D(\bar{x} + \delta\bar{x}) - D(\bar{x}) + D(\bar{x}) - x \\ &= D(\bar{x} + \delta\bar{x}) - D(\bar{x}) + \Delta x \end{aligned}$$

The error is a sum of two components: the demosaicking error of the CFA image and the difference between the demosaicked versions of the compressed and the original CFA images $D(\bar{x} + \delta\bar{x}) - D(\bar{x})$. Hence, the total error variance is

$$\Psi^2 = E \{ [D(\bar{x}_i + \delta\bar{x}_i) - D(\bar{x}_i)]^2 \} + E(\Delta^2 x_i) + 2E \{ [D(\bar{x}_i + \delta\bar{x}_i) - D(\bar{x}_i)] \Delta x_i \} \quad (2.5)$$

Clearly, for a given demosaicking method, $E \{ [D(\bar{x}_i + \delta\bar{x}_i) - D(\bar{x}_i)]^2 \}$ depends on the compression error variance $\xi_o^2 \triangleq E(\delta^2 \bar{x}_i)$. Let $d(\xi_o^2) = E \{ [D(\bar{x}_i + \delta\bar{x}_i) - D(\bar{x}_i)]^2 \}$. Similar to the conventional chain, we expand $d(\xi_o^2)$ by Taylor series and retain only the linear item. Then $d(\xi_o^2) \approx b_d^2 \xi_o^2$ and $E \{ [D(\bar{x}_i + \delta\bar{x}_i) - D(\bar{x}_i)] \delta x_i \} \approx b_d E(\Delta x_i \delta \bar{x}_i) = \rho_b b_d \xi_o \zeta$, where ρ_b is the correlation coefficient between the CFA image compression error and demosaicking error, and the parameter $b_d^2 = E \{ [\frac{\partial D(x_i)}{\partial x_i}]^2 \}$ can in general depend on the demosaicking method. Combining these approximations yields the following expression for the total error variance of the alternative processing chain:

$$\Psi^2 = \zeta^2 + b_d^2 \xi_o^2 + 2\rho_b b_d \zeta \xi_o \quad (2.6)$$

To verify the validity of the error model (2.6), we applied alternative pro-

Figure 2.10: Relationship between Ψ_o^2 and ζ^2 for 15 test images. Each curve corresponds to one test image.

cessing (compressing CFA images as grayscale images) to 15 test images and the results are shown in Figures 2.10 and 2.11. The total error variance Ψ^2 is plotted as a function of the demosaicking error variance ζ^2 in Figure 2.10, and as a function of the compression error variance ξ_o^2 in Figure 2.11. Similar to Section 2.3.1, we estimate Ψ^2 , ζ^2 and ξ_o^2 by taking the MSE over the entire image, and control ζ^2 by varying usage of demosaicking algorithms and ξ_o^2 by varying compression ratio.

The curves shown in Figure 2.10 are linear for low compression ratios, with the slope close to 1. This is expected, since low compression ratios lead to compression errors smaller than demosaicking errors, $\xi_o^2 \ll \zeta^2$, and hence $\Psi^2 \approx \zeta^2$ according to (2.6). When the compression error is comparable to (or larger than) the demosaicking error, the other two terms become significant and break the linearity. In Figure 2.11 the curves are also linear for large compression errors and small demosaicking errors, which agrees with the error model (2.6)

Figure 2.11: Relationship between Ψ_o^2 and ξ_o^2 for 15 test images. Each curve corresponds to one test image.

because in this case $\Psi^2 \approx b_d^2 \xi_o^2$. For large demosaicking errors (e.g. those from bilinear demosaicking), linearity breaks due to the increasing influence of the cross term in (2.6).

We can use the curves in Figure 2.11 to estimate the parameters b_d^2 and ρ_b , shown in Table 2.3, and provide an intuition for the observed values of b_d^2 . As expected, b_d^2 remains approximately the same with different compression methods, since the model suggests that b_d^2 should depend only on the demosaicking process. Surprisingly, the value of b_d^2 is also nearly the same (and slightly larger than 1) for most demosaicking methods, except for bilinear demosaicking where $b_d^2 < 1$. This is because bilinear demosaicking is essentially a low-pass filtering process which can reduce the preceding compression error by suppressing high frequencies. The other advanced demosaicking methods are better at preserving high frequencies and hence have little effect on the preceding compression error

Table 2.3: Estimated values of b_d^2 and ρ_b in the error model of alternative processing chain [Eq. (2.6)].

		Bilinear	Hamilton [1]	AP [2]	iECI [17]
JPEG	b_d^2	0.876	1.081	1.053	1.083
	ρ_b	-0.090	-0.090	-0.111	-0.204
JPEG2000	b_d^2	0.888	1.119	1.109	1.101
	ρ_b	-0.037	-0.024	-0.038	-0.056

$$b_d^2 \approx 1.$$

Similar to ρ_a , parameter ρ_b is also small, not exceeding 0.1 in amplitude on average, as shown in Table 2.3. Hence we can simplify the model (2.6) to the following:

$$\Psi^2 = \zeta^2 + b_d^2 \xi_o^2 \quad (2.7)$$

where $b_d^2 = 0.88$ for bilinear demosaicking and $b_d^2 = 1.1$ for other advanced ones.

Figure 2.12 shows that the prediction from the simple model presents a very good fit to the experimental results.

Figure 2.12: Predicted the error of alternative chain using simple model (2.7).

2.3.3 Quick comparison of processing chains

To analyze the relative performance of the conventional and alternative processing chains, we consider two extreme cases of low and high compression ratios. Based on the simple models (2.4) and (2.7), we obtain the following expression for the difference of total error variances

$$\Omega^2 - \Psi^2 = (\xi_s^2 - b_d^2 \xi_o^2) + (a_c^2 - 1)\zeta^2 \quad (2.8)$$

At high compression ratios, the compression error is larger than the demosaicking error and the relative performance depends mainly on $\xi_s^2 - b_d^2 \xi_o^2$. At low compression ratios, both compression and demosaicking error terms should be preserved, because while $(\xi_s^2 - b_d^2 \xi_o^2) < \zeta^2$, parameter a_c^2 can be close to 1 (see Table 2.2), leading to cancelation of demosaicking error term.

According to (2.8), the superiority of one processing chain over the other depends on the values of $\xi_s^2 - b_d^2 \xi_o^2$ and a_c^2 . For example, for bilinear demosaicking and JPEG compression at low compression ratios, $a_c^2 < 1$ and $\zeta^2 \gg \xi^2$ and hence we can conclude that the conventional processing chain outperforms the alternative one. On the other hand, for JPEG2000 compression at low compression ratios we have $a_c^2 \approx 1$ and hence the demosaicking error can be neglected in comparison with the compression error. As we shall show later in Section 2.5, in this case $\xi_s^2 - b_d^2 \xi_o^2 > 0$ and the alternative processing chain outperforms the conventional one.

In order to make more precise predictions about the relative performance of the two processing chains, we develop a model of compression error variance in the next section.

2.4 Compression Error Model

The compression error model (1.1) for 1-D signal can be extended to images modeled by 2D Markov random processes [40, 41, 44–47]. In our analysis, we use a second order stationary Markov random process as the image model [51]. For a grayscale image $I(m, n)$ (or an individual component of a color image), we define the pixel variance and inter-pixel correlation as follows:

$$\begin{aligned}\sigma_I^2 &= E\{I^2(m, n)\} \\ \rho_{i,j} &= \frac{E\{I(m, n)I(m+i, n+j)\}}{\sigma_I^2} = \rho_0^{\sqrt{i^2+j^2}}\end{aligned}\quad (2.9)$$

Here ρ_0 is the correlation between two pixel adjacent to each other in the horizontal or vertical direction.

However, the quantization model — $\sigma_{qk}^2 = \epsilon^2 \sigma_k^2 2^{-2R_k}$ [42], the foundation of (1.1), is valid only at high bitrates R_k [43]. To show this, in Figure 2.13 we plot the logarithm of the compression error averaged over 15 test images as a function of bitrate R . The plot is linear for relatively high bitrates ($R > 1.5$ bpp) with slope -2 , as predicted by the model, but becomes nonlinear at smaller bitrates. We can generalize model (1.1) to lower bitrates by allowing β to be a function of bitrate:

$$\xi_I^2 = \epsilon^2 \sigma_I^2 2^{-2\beta(R_I)R_I} \mathcal{G}_I^{-1} \quad (2.10)$$

Based on the generalized error model (2.10), we can now derive the compression error models for full-color and CFA images.

Figure 2.13: Average logarithm of compression error variance *versus* bitrate.

2.4.1 Full-color image compression

Components of color images (R, G, and B) are typically compressed independently by treating each component as a grayscale image. Let $\mathcal{C} \in \{R, G, B\}$ be an arbitrary color component. Using the error model (2.10), the compression error of $\mathcal{C}$ is given by $\xi_c^2 = \epsilon^2 \sigma_c^2 2^{-2\beta(R_c)R_c} \mathcal{G}_I^{-1}$. Here σ_c^2 and R_c are the pixel variance and the assigned bitrate of component $\mathcal{C}$, respectively, and $\sum_{\mathcal{C}} R_c = R$. Hence, the average compression error of all color components is

$$\xi_s^2 = \frac{1}{3} \epsilon^2 \mathcal{G}_I^{-1} \sum_{\mathcal{C}} \sigma_c^2 2^{-2\beta(R_c)R_c} \quad (2.11)$$

A special case is compression in RGB space, where R, G, and B color components are compressed independently. Assuming that the RGB color components all have the same variance σ_I^2 and inter-pixel correlation, choosing equal bitrate for each component, i.e., $R_c = \frac{R}{3}$, minimizes the overall compression error. Then (2.11) becomes

$$\xi_s^2 = \epsilon^2 \sigma_I^2 2^{-2\beta(\frac{R}{3})\frac{R}{3}} \mathcal{G}_I^{-1} \quad (2.12)$$

Table 2.4: Similarity between $\rho_0 \cdot \nu_{s_1, s_2}$ and its exact values.

Test Image	R&G		G&B		R&B	
	$\rho_0 \cdot \nu$	Exact	$\rho_0 \cdot \nu$	Exact	$\rho_0 \cdot \nu$	Exact
1	0.671	0.652	0.449	0.398	0.842	0.847
2	-0.356	-0.380	-0.957	-0.992	0.474	0.466
3	0.682	0.677	0.105	0.088	0.429	0.419
4	0.183	0.166	0.170	0.152	0.925	0.925
5	0.833	0.830	0.607	0.584	0.755	0.746

However, the compression performance can be improved by decorrelating RGB components. This can be accomplished by transforming the image into an appropriate color space, such as YUV or YCbCr space [28, 38]. Let $S(m, n) = [R(m, n), G(m, n), B(m, n)]^T$ be the color vector in RGB space and $S_t(m, n) = [Y(m, n), U(m, n), V(m, n)]^T$ be the corresponding color vector in the transformed space, where $S_t(m, n) = T_s S(m, n)$ and T_s is the color transformation matrix.

Let the inter-color correlation be ν_{s_1, s_2} , and assume the inter-color correlations between non-coinciding pixels

$$\frac{E\{S_1(m, n)S_2(m + i, n + j)\}}{\sigma_I^2} = \rho_{i,j}\nu_{s_1, s_2} \quad (2.13)$$

This assumption is sufficiently precise, as evidenced by the experimental results in Table 2.4.

Let v_c^T be the row of T_s corresponding to color component $\mathcal{C} \in \{Y, U, V\}$. It follows from (2.13) that the inter-pixel correlation in component $\mathcal{C}$ is the same as in RGB color components:

$$\frac{E\{\mathcal{C}(m, n)\mathcal{C}(m + i, n + j)\}}{\sigma_c^2} = \frac{\rho_{i,j}v_c^T E\{S(m, n)S(m, n)^T\}v_c}{\sigma_c^2} = \rho_{i,j} \quad (2.14)$$

However, the variances of the color transformed components are no longer equal

to each other. Hence, assigning equal bitrate to each component does not necessarily minimize the overall error.

To determine the optimal bitrate allocation, we can use error model (2.10) by treating the color transformation as a 3-subband decomposition. Then the analysis and synthesis filters h_k and g_k are rows of T_s and T_s^{-1} respectively, and the inter-pixel correlations in (2.10) are replaced by inter-color correlations. Furthermore, parameters $\mathcal{A}_c$ and $\mathcal{B}_c$ for $\mathcal{C} \in \{Y, U, V\}$ can be derived from (1.3) and (1.4). The overall compression error becomes

$$\xi_s^2 = \epsilon^2 \sigma_I^2 2^{-2\beta(\frac{R}{3})\frac{R}{3}} \prod_{\mathcal{C} \in \{Y, U, V\}} (\mathcal{A}_c \mathcal{B}_c)^{\frac{1}{3}} \prod_{k=0}^{K-1} (A_k B_k)^{\alpha_k} = \epsilon^2 \sigma_I^2 2^{-2\beta(\frac{R}{3})\frac{R}{3}} \mathcal{G}_s^{-1} \quad (2.15)$$

where $\mathcal{G}_s$ is the new coding gain. This is similar to (2.12), except for the extra term $\prod_{\mathcal{C} \in \{Y, U, V\}} (\mathcal{A}_c \mathcal{B}_c)^{\frac{1}{3}}$, which is related to the color transformation T_s . Without color transformation, T_s becomes the identity matrix and $\mathcal{A}_c = \mathcal{B}_c = 1$, and (2.15) becomes (2.12).

2.4.2 CFA image compression

Unlike a grayscale image, the CFA image $O(m, n)$ may not be locally stationary because of the difference in inter-color correlation. For example $E\{O(2m, 2n)O(2m, 2n + 1)\} = \rho_0 \nu_{GB} \sigma_I^2$ is not equal to $E\{O(2m + 1, 2n + 1)O(2m + 1, 2n)\} = \rho_0 \nu_{RG} \sigma_I^2$ if $\nu_{GB} \neq \nu_{RG}$. For simplicity, we approximate the CFA image as a stationary process with the correlation averaged over six inter-pixel correlations (two horizontal, two vertical, and two diagonal) in a 2×2

Bayer unit (see Figure 4). Then from the error model (2.10) we obtain

$$\xi_o^2 = \epsilon^2 \sigma_I^2 2^{-2\beta(R)R} \prod_{k=0}^{K-1} (A_{ok} B_k)^{\alpha_k} = \epsilon^2 \sigma_I^2 2^{-2\beta(R)R} \mathcal{G}_o^{-1} \quad (2.16)$$

where $\mathcal{G}_o$ is the corresponding coding gain, computed according to the new inter-pixel correlations.

However, treating a CFA image as grayscale leads to poor compression performance, as discussed earlier in Section 2.2. Performance can be improved by using a luminance-color difference transformation. The red (R), blue (B), and two green samples (G_1 and G_2) in a 2×2 Bayer unit are converted into Y_1 , Y_2 , U , and V samples (as shown in Figure 4) using a linear transformation defined by a 4×4 matrix. With this transformation, the compression error variance becomes

$$\xi_t^2 = \epsilon^2 \sigma_I^2 2^{-2\beta(R)R} \prod_{\mathcal{C} \in \{Y_1, Y_2, U, V\}} \left(\mathcal{A}_c \mathcal{B}_c \prod_{k=0}^{K-1} (A_{ck} B_k)^{\alpha_k} \right)^{\frac{1}{4}} = \epsilon^2 \sigma_I^2 2^{-2\beta(R)R} \mathcal{G}_t^{-1} \quad (2.17)$$

where $\mathcal{G}_t$ is the corresponding coding gain. Different from that of a full-color image (2.14), the inter-pixel correlations in a CFA image will change under color transformation. Hence parameter A_{ck} in (2.17) depends on the color component $\mathcal{C}$, which explains the difference between the expressions of coding gains in (2.15) and (2.17).

The correlated components Y_1 and Y_2 can be more efficiently compressed as one image Y [24–26]. Since the samples of the combined image are arranged on a quincunx lattice (Figure 2.4), existing work has suggested to rotate Y by 45° , converting a quincunx lattice to a rectangular lattice with diamond shaped support [24,25]. The formula (2.17) is still applicable if we recalculate parameter

A_{ck} based on the diagonal inter-pixel correlations in a quincunx lattice.

2.4.3 Evaluation of the compression error models

Recall from Section 2.3.3 that one of the factors that determine the relative performance of two processing chains is the sign of $\xi_s^2 - b_d^2 \xi_o^2$, or equivalently the sign of $\frac{\xi_s^2}{\xi_o^2} - b_d^2$. From (2.15) and (2.16), we obtain

$$\frac{\xi_s^2}{\xi_o^2} = 2^{2\beta_r(R)R} \frac{\mathcal{G}_o}{\mathcal{G}_s} \quad (2.18)$$

where $\beta_r(R) = \beta(R) - \frac{\beta(\frac{R}{3})}{3}$, while $\mathcal{G}_s$ and $\mathcal{G}_o$ are the respective coding gains. Clearly, the logarithmic ratio (log-ratio) between the two error variances is proportional to $\beta_r(R)R$, and can in general be a nonlinear function of the bitrate. However, the experimental results show that the log-ratio is approximately linear for bitrate $R > 1.5$ bpp (bits per pixel), as shown in Figure 2.14, with the slope 0.84 for JPEG and 1.20 for JPEG2000. Note that for a very large bitrate ($\frac{R}{3} > 1.5$ bpp), $\beta(R) \approx \beta(\frac{R}{3}) \approx 1$ (see Figure 2.13) and hence the slope should be close to $\frac{4}{3} \approx 1.33$. But in the bitrate range of our interest $1.5 < R < 5$, which corresponds to the working bitrate of most DSCs [8], $\beta(\frac{R}{3}) > 1$ and this explains the observed reduced slope values.

Figure 2.14 also shows that the break-point bitrate, corresponding to $\frac{\xi_s^2}{\xi_o^2} = b_d^2 \approx 1$ (Table 2.3), is $R_b \approx 3.8$ bpp for JPEG and $R_b \approx 2.2$ bpp for JPEG2000. Since the break-points occur at the linear portion of the curves, we can predict their position by treating $\beta_r(R)$ in (2.18) as a constant, which can be estimated by fitting (2.18) to actual data at high bitrates. The curves of the predicted log-ratios are shown in Figure 2.14. Our predicted break-point for JPEG2000

Figure 2.14: Ratio between the compression error variances of the full-color and CFA images.

is $R_b = 2.2$ bps, which practically coincides with the actual value. For JPEG, the predicted value is $R_b = 3.6$ bps, which is slightly smaller than the actual $R_b = 3.8$ bps. We also conclude that $\frac{\xi_s^2}{\xi_o^2} < b_d^2$ for $R < R_b$ and $\frac{\xi_s^2}{\xi_o^2} > b_d^2$ for $R > R_b$, at least for the tested range of bitrates³. This fact will be used in comparison of the two processing chains in Section 2.5.

2.5 Performance Comparison of Two Processing Chains

In this section we analyze the relative performance of the conventional and alternative processing chains based on the proposed models for the compression error and the interaction of compression and demosaicking errors.

³Figure 2.14 also shows that the ratio $\frac{\xi_s^2}{\xi_o^2}$ in JPEG increases at bitrate $R < 0.8$ bps. This happens because JPEG's $\beta(R)$ becomes too large at very low bitrates, leading to negative $\beta_r(R)$. It is possible that the ratio can become larger than b_d^2 at extremely low bitrates, causing superiority of alternative processing chain, which is consistent with [26]. However, such high compression ratios ($> 80:1$) lead to very poor image quality, and is out of the interest.

Table 2.5: Superiority of processing chains (conventional or alternative).

Compression ratios	JPEG and		JPEG2000
	simple demosaicking	advanced demosaicking	
Low	Conventional	Alternative	Alternative
High	Conventional	Conventional	Conventional

In summary, the relative performance of the two processing chains for high and low compression ratios depends on the sign of $\xi_s^2 - b_d^2 \xi_o^2 + (a_c^2 - 1)\zeta^2$; see Section 2.3.3. For high compression ratios, $\xi_s^2 - b_d^2 \xi_o^2$ has a large negative value and $a_c^2 \leq 1$; hence, the conventional processing chain always outperforms the alternative one. At low compression ratios, $\xi_s^2 - b_d^2 \xi_o^2$ has a small positive value, and the answer depends on the value of parameter a_c^2 . For JPEG2000, $a_c^2 \approx 1$ makes $(a_c^2 - 1)\zeta^2 \approx 0$ and leads to superiority of the alternative chain. But in JPEG, $a_c^2 < 1$ and for large demosaicking errors (e.g., that of bilinear demosaicking) the term $(a_c^2 - 1)\zeta^2$ becomes large and negative, making the conventional chain superior. However, at low compression ratios, the alternative chain can still outperform the conventional one for sufficiently small errors obtained by advanced demosaicking methods (AP, iECI). Table 2.5 summarizes these predictions.

The positivity of $\xi_s^2 - b_d^2 \xi_o^2$ (or equivalently $\frac{\xi_s^2}{\xi_o^2} - b_d^2$) is the reason why the alternative processing is capable of outperforming the conventional one. According to (2.18), $\frac{\xi_s^2}{\xi_o^2}$ depends on the ratio of the coding gains $\frac{\mathcal{G}_o}{\mathcal{G}_s}$ and the bitrate via $2^{2\beta_r(R)R}$. Full-color images have larger inter-pixel and inter-color correlations compared with the corresponding CFA images, leading to larger coding gains $\mathcal{G}_s > \mathcal{G}_o$. However, full-color images also have three times as many pixel values as that of CFA images. This is captured by the term $2^{2\beta_r(R)R}$. This term dominates at high bitrates leading to $\frac{\xi_s^2}{\xi_o^2} = 2^{2\beta_r(R)R} \frac{\mathcal{G}_o}{\mathcal{G}_s} > b_d^2$ and hence brings about the superiority of the alternative processing chain.

Qualitative predictions in Table 2.5 agree very well with the experimental results shown in Figure 2.15. In this figure we compare the signal-to-noise ratio (SNR) performance of the conventional and alternative processing chains, where the latter applied a simple grayscale compression for CFA images. As expected, the alternative processing chain outperforms the conventional one at low compression ratios (high bitrates) for JPEG2000 compression. The same is true for JPEG compression combined with AP and iECI demosaicking methods. However, in the combination of JPEG compression and bilinear demosaicking, the conventional chain performs better for all tested compression ratios, as shown in Figure 2.15.

Apart from the special case of JPEG compression combined with bilinear demosaicking, all other combinations lead to the following simple conclusion: the conventional processing chain performs better at high compression ratios and the alternative processing chain performs better at low compression ratios. To delineate high and low compression ratios we can use a break-point compression ratio t_b (or a break-point bitrate $R_b = \frac{24}{t_b}$) at which the processing chains have equal performance.

To show that the developed model can be used to predict the break-point bitrate R_b , we conducted the following experiment. For each test image we estimated its statistical parameters, such as σ_I^2 , ρ_0 and $\rho_{s_1 s_2}$, and used these to predict the compression error using (2.15) or (2.16). Since the demosaicking error does not depend on the bitrate, we evaluated it empirically. The compression and demosaicking errors were then combined using the simplified models (2.4) and (2.7) and plotted as a function of bitrate (Figure 2.15). The results show that for JPEG with AP or iECI demosaicking method $R_b \approx 4.5$ bpp, while

2.5. Performance Comparison of Two Processing Chains

78

Figure 2.15: Predicted *versus* actual performance comparison between conventional and alternative processing chains. The actual results are computed from the average of the last 15 images in Figure 2.16.

Figure 2.16: Test images (referred to as Image 1 to Image 30, enumerated from left-to-right and top-to-bottom).

the predicted $R_b \approx 4.0$ bpp. For JPEG2000, both experimental and predicted results yield $R_b \approx 2.5$ bpp.

Prediction results can be extended to more advanced (color transformation based) CFA image compression methods by using (2.17). The corresponding curves are shown in Figure 2.15 (designated as Ψ_t^2) and indicate that the break-points occur at lower bitrates. This is because color transformation removes discontinuities in CFA images caused by interlaced color components, leading to a larger coding gain $\mathcal{G}_t > \mathcal{G}_o$. As the image size remains unchanged, larger coding gain leads to smaller compression error $\xi_t^2 < \xi_o^2$, moving the break-points to lower bitrates.

To show the accuracy of our analysis, we have compared the predicted break-points to several results that have been recently reported in the literature. For example, using color transformation and JPEG based CFA image compression,

the predicted break-point is $R_b \approx 3.0$ bpp for Hamilton and $R_b \approx 2.5$ bpp for AP and iECI. These bitrates correspond to compression ratios $t_b \approx 8:1$ - $10:1$, which are consistent with the results in [26]. For JPEG2000 based CFA image compression, our model predicts $R_b \approx 1.5$ bpp, corresponding to compression ratio $t_b \approx 16:1$, which coincides with the results in [25]⁴.

Figure 2.17: Performance comparison between the conventional processing chain using bilinear and the alternative one using iECI.

In the alternative processing chain the demosaicking operation can be moved to a PC, allowing for more advanced demosaicking methods to be used. Doing

⁴The 5:1 ratio reported in the paper takes the size of a CFA image as reference. This translates to 15:1 if we reference to the size of a full-color image.

Figure 2.18: Comparison of the images reconstructed from a real CFA image by using *bilinear* demosaicking and JPEG2000 compression (10:1).

this will further increase the break-point compression ratios. Figure 2.17 compares the conventional processing chain using bilinear and the alternative one using iECI. Even for simple CFA image compression (as grayscale) the break-point becomes $t_b \approx 15:1$ for JPEG, and $30:1$ for JPEG2000. For more advanced CFA image compression we expect the actual break-points to reach $t_b \approx 30:1$ for JPEG, or $100:1$ for JPEG2000.

Figure 2.18 shows an example obtained by using a real CFA image from DSC. Besides some blurring effect due to the use of a simple (bilinear) demosaicking method, the reconstructed images obtained by the two processing chains have similar quality under a typical DSC compression ratio 10:1. On the other hand, such a simple demosaicking method yields color artifacts in a test CFA image with complex details and sharp edges [see Figure 2.19(b)(c)]. In this case using advanced demosaicking methods (AP, iECI) in the alternative processing chain can suppress the artifacts, leading to a superior image with sharper edges [see Figure 2.19(e)].

It is worth noting that even though there is a small difference between the predicted and actual error values, which is mainly due to the simplifications

(a) Test image

(b) Conventional, Bilinear

(c) Alternative, Bilinear

(d) Conventional, iECI

(e) Alternative, iECI

Figure 2.19: Comparison of the reconstructed images after demosaicking and JPEG2000 compression (10:1) of CFA data downsampled from a test image.

made in deriving the error models of the processing chains, such a discrepancy has very little effect on predicting the superiority of a particular processing chain. If necessary, the error models could be made more complex by considering additional terms in the Taylor series expansions as well as the cross terms of compression and demosaicking errors discussed in Section 2.3.

Apart from predicting the break-point compression ratio, our model also provides an important insight for choosing suitable reference images for performance evaluation, as described in the next section.

2.6 Suitable Reference Images for Performance Evaluation

Some recent performance evaluation studies of the alternative and conventional processing chains [24, 25] used demosaicked images as the ground truth. While this allows performance comparison on the real CFA data, it may lead to wrong conclusions about the relative performance of the two processing chains.

If a demosaicked image is used as a reference, the error of the conventional processing chain $\bar{\Omega}^2$ is equal to the compression error of the demosaicked image. Demosaicking smoothes the image and hence reduces the error of the compression process that follows. This reduction is smaller for advanced demosaicking methods, since the resultant image is closer to the original. We can model this as

$$\bar{\Omega}^2 = \gamma^2 \xi_s^2 \quad (2.19)$$

where $\gamma^2 \leq 1$. The parameter γ^2 is close to 1 for advanced demosaicking

methods (such as iECI). For the alternative processing chain, we can use the same method as described in Section 2.3.2 to obtain the error variance

$$\bar{\Psi}_o^2 = E \{ (D[C(\bar{x})] - D(\bar{x}))^2 \} = b_d^2 \xi_o^2 \quad (2.20)$$

where b_d^2 is the same parameter as in (2.6). Performance comparison is then based on the sign of

$$\bar{\Omega}^2 - \bar{\Psi}_o^2 = \gamma^2 \xi_s^2 - b_d^2 \xi_o^2 \quad (2.21)$$

This is similar to $\xi_s^2 - b_d^2 \xi_o^2$, which is the first term in (2.8). However, the second term in (2.8), corresponding to the demosaicking error, is missing.

Using (2.21) instead of (2.8) may or may not affect the conclusions about the relative performance of the two processing chains. For example, in JPEG2000 the demosaicking error has little effect on the relative performance. In this case $a_c^2 \approx 1$ in (2.8) and the main difference between (2.21) and (2.8) is in the parameter γ^2 , which may lead to a lower R_b [see Figure 2.20(c)(d)] but preserves the relationship between the processing chains at low and high compression ratios.

On the other hand, the demosaicking error ζ^2 plays a more important role in JPEG based processing, especially at low compression ratios, as discussed in Section 2.5. For example, the large errors of bilinear demosaicking lead to superiority of the conventional processing chain at all compression ratios. But it follows from (2.21) that the alternative processing chain should perform better than the conventional one at low compression ratios, as shown in Figure 2.20(a)(b).

In summary, using demosaicked image as a reference leads to correct con-

(a) JPEG, Bilinear

(b) JPEG, iECI

(b) JPEG2000, Bilinear

(b) JPEG2000, iECI

Figure 2.20: Relative performance of alternative and conventional processing chains for different reference images: left column, demosaicked image; right column, original full color images.

clusions when JPEG2000 is combined with advanced demosaicking methods, because in this case $\gamma^2 \approx 1$. This explains why the results in [25] coincide with ours, even though different reference images were used. However, combining JPEG with bilinear demosaicking could lead to wrong conclusions, as in the work by Lee and Ortega [24].

2.7 Conclusions

Alternative processing chain offers advantage in DSC processing by off-loading the demosaicking operation to an end device such as a computer or PDA. However, to be widely adopted the alternative processing should not lead to notable degradation in image quality. Recently reported empirical results [24–26] have indicated that the alternative processing is able to provide similar or even better image quality than the conventional processing at low compression ratios. In this chapter we provide better insight into the relative performance of the two processing chains by developing the analytical models for the reconstruction error variances.

The models developed not only confirm the empirical results reported in the literature, but can also accurately predict the break-point compression ratio, below which the alternative processing outperforms the conventional one. The reason of superiority is that the CFA images compressed in the alternative processing have fewer pixel values, leading to smaller error at low compression ratios despite the fact that more bits are used to code each pixel.

Our analysis also suggests that using CFA image compression in conjunction with a proper color transformation, the alternative processing chain can outper-

form the conventional one at compression ratios less than 8:1-10:1 with JPEG, and less than 16:1 with JPEG2000. If an advanced demosaicking method is used in the alternative processing, the range of superiority can be even larger, which well covers the typical compression ratios used in current DSCs [8]. It should be noted that one obstacle for wide application of the alternative processing is the customized compression techniques required for the interlaced CFA image. While such compression techniques already exist, they have yet to be standardized.

Chapter 3

Error Inhomogeneity of Wavelet Based Compression

3.1 Introduction

Due to its superior compression performance, wavelet-based JPEG2000 compression is set to replace JPEG in the near future [38]. For example, in Chapter 2 we have shown that JPEG2000 compression is more suitable for DSCs based on alternative processing chain, thus increasing the possibility of its application in future DSCs. However, wavelet-based compression (JPEG2000) can lead to spatially varying error, i.e. some locations in the reconstructed image can have more or less distortion than the others. This fact has not been so far reported in the literature, but can be easily demonstrated using the following experiment. Let us divide the reconstructed image into four sub-images, as shown in

Figure 3.1. Then it can be shown experimentally that regardless of the image content, the sub-image 0 has consistently higher PSNR performance than other sub-images and sub-image 3 has the lowest. The difference in PSNR can be quite substantial, up to 3.4 dB for some images and compression ratios. We refer to this phenomenon as the error inhomogeneity of wavelet based coding.

Figure 3.1: Image division into four sub-images.

In this chapter we show that the error inhomogeneity is caused by asymmetrical filtering of quantization errors after the upsampling step in wavelet synthesis process. We show that error variances of even and odd samples of the reconstructed signal depend on different synthesis filter coefficients, and hence in general are not equal to each other. Moreover, for popular wavelets, such as Daubechies 9-7 and 5-3, the variance of the even samples is consistently larger than the variance of the odd samples, leading to the observed error inhomogeneity. Nonuniformity and correlation of quantization errors can also affect the inhomogeneity, albeit to a smaller degree. To quantify these observations, we develop a model that allows predicting the amount of inhomogeneity for a given wavelet.

Predicting error inhomogeneity can help in a variety of applications that rely on the statistical modeling of the noise in a local neighborhood. For exam-

ple, better quality resizing of the wavelet-compressed image can be obtained by assigning larger weights to pixels of the sub-image 0. Another example comes from image superresolution problem, where the goal is to estimate a high resolution image from a set of its low resolution versions. When low resolution images undergo wavelet compression, knowledge of error inhomogeneity could lead to better high resolution image prediction accuracy.

For some applications simple awareness of the inhomogeneity may not be sufficient. For such applications we show how to redesign wavelet filters to reduce the error inhomogeneity, at a cost of small reduction in the overall PSNR performance. For applications that are sensitive to PSNR degradation, we show how to design wavelet filters that gradually trade-off the PSNR performance for reduced inhomogeneity.

The chapter is organized as follows. We first present several experimental results in Section 3.2. In Section 3.3 we develop a model for the reconstruction error of even and odd samples of the signal. We then use the developed model to predict inhomogeneity for several typical assumptions on the quantization error statistics (Section 3.4). In Sections 3.5 and 3.6 we explain a design procedure for creating wavelet filters with reduced error inhomogeneity at a cost of a controlled overall performance degradation. We conclude with several simple applications and discussion in Sections 3.7 and 3.8.

3.2 Supporting Experiments and Motivation

Error inhomogeneity in images reconstructed after wavelet compression can be easily demonstrated by dividing the images into four sub-images, as shown in

(a) Grayscale images

(b) Color images

Figure 3.2: Test images used in the experiments

Figure 3.1, and calculating the error variance for each sub-image. We conducted this experiment on 24 test images (Figure 3.2), and the results are shown in Tables 3.1-3.4. Here by inhomogeneity we mean the difference of PSNR performance between sub-images 0 and 3. The inhomogeneity can be observed for different compression methods (see Tables 3.1 and 3.2), wavelet filters (see Tables 3.1, 3.3 and 3.4), and compression bitrates (see Table 3.5). Even though inhomogeneity varies from image to image, its statistical P-value $P < 0.00001$ is sufficiently small, ruling out the possibility of random occurrence. The amount of inhomogeneity is large enough for visual observation, as shown on the ex-

ample of “barbara” and “lena” in Figure 3.3. For both images, some parts of sub-image 3 are lacking texture that is well preserved in the sub-image 0.

Table 3.1: Comparison of sub-image PSNR for 19 grayscale test images (bitrate = 0.50 bpp, SPIHT, 9-7 wavelet)

Image	PSNR (dB)	The PSNR of sub-images (dB)				Inhomogeneity
		0	1	2	3	
AIRFIELD	28.57	29.18	28.59	28.62	27.96	1.22
BABOON	27.03	27.77	26.95	26.99	26.49	1.28
BARBARA	31.39	32.78	31.49	31.15	30.47	2.31
BRIDGE	27.27	27.92	27.25	27.25	26.73	1.19
CLOCK	47.36	48.07	47.13	47.57	46.77	1.30
COUPLE	32.45	33.25	32.40	32.52	31.76	1.49
CROWD	33.88	34.63	33.93	33.88	33.21	1.42
GIRL	37.46	38.24	37.05	37.92	36.78	1.46
GOLD	33.08	33.64	33.18	32.94	32.61	1.03
HARBOR	30.01	30.82	29.92	30.03	29.38	1.45
LAX	27.53	28.01	29.92	27.52	27.05	0.96
LENA	37.21	37.93	37.27	37.15	36.60	1.33
MAN	33.08	33.85	32.98	33.10	32.49	1.36
PAOLINA	37.13	37.46	37.15	37.14	36.81	0.65
PEPPERS	35.81	36.22	35.77	35.84	35.43	0.79
SAILBOAT	31.57	32.38	31.57	31.61	30.84	1.54
TIFFANY	37.25	38.06	37.17	37.10	36.77	1.29
WOMAN	42.54	43.28	42.52	42.55	41.92	1.36
ZEBRA	26.77	27.55	26.58	26.93	26.14	1.41
		Statistical P-value				10^{-12}

Simple awareness of error inhomogeneity can already be sufficient to improve the performance for some applications. For example, interpolation predicts the intensity at a given location from the intensities of the neighboring pixels. Hence assigning larger weights to pixels in sub-image 0 and smaller weights to pixels in sub-image 3 will improve interpolation quality. However, other application may require removing or at least reducing the error inhomogeneity prior to processing. This is exactly the topic of the rest of the chapter.

Figure 3.3: Comparison of sub-images 0 and 3 for different wavelets and bitrates. (a-b) 9-7 wavelet, 0.50bpp, (c-d) 9-7 wavelet, 0.20bpp, (e-f) 9-3 wavelet, 0.20bpp.

Table 3.2: Comparison of sub-image PSNR for 5 color images (bitrate = 0.50 bpp, JPEG2000, 9-7 wavelet)

Image	PSNR (dB)	The PSNR of sub-images (dB)				Inhomo- geneity
		0	1	2	3	
HOUSE	34.05	34.56	34.11	34.11	33.50	1.06
	36.21	37.11	36.33	36.51	35.14	1.97
	33.91	34.17	33.97	33.93	33.59	0.58
TRIGLE	38.65	39.11	38.73	38.57	38.22	0.89
	38.20	39.02	38.25	38.12	37.53	1.49
	38.38	39.43	38.72	38.37	37.28	1.10
BIKE	36.63	37.09	36.58	36.66	36.22	0.87
	37.19	38.04	37.09	37.25	36.53	1.51
	34.60	35.36	34.56	34.62	33.99	1.37
GIRL	37.60	38.19	37.64	37.60	37.06	1.13
	39.12	39.81	39.17	39.12	38.47	1.34
	36.81	37.34	36.80	36.86	36.31	1.03
SILVER	40.14	40.13	40.12	40.16	40.13	0.01
	41.18	41.03	41.14	41.26	41.31	-0.13
	39.19	39.22	39.15	39.22	39.17	0.02
Statistical P-value						10^{-5}

3.3 Error Modeling for Even and Odd Signal Samples

Wavelet image compression uses separable 2D wavelet transform that allows implementation via two 1D wavelet transforms. Hence we consider simpler 1D case that can be trivially extended to 2D case. Shown in Figure 3.4 is a standard 1D block scheme of the wavelet-based compression [74]. Here filters h_0, h_1 comprise the analysis filter bank, filters g_0, g_1 the synthesis filter bank, and $Q(\cdot)$ denotes quantization. Let $x(n)$ and $\bar{x}(n)$ be the original signal and its reconstructed version respectively, $w_i(n)$ and $\bar{w}_i(n)$ $i = 0, 1$ be the wavelet coefficients before and after quantization. Note that $\bar{x}(n)$ is obtained by convolving the upsampled version of quantized coefficient sequence $\bar{w}_i(n)$ with the synthesis filters g_i , as shown in Figure 3.5. Let the wavelet quantization error be $q_i = w_i(n) - \bar{w}_i(n)$, $i = 0, 1$, and the error of whole processing chain

Table 3.3: Comparison of sub-image PSNR for 19 grayscale test images (bitrate = 0.50 bpp, SPIHT, 9-3 wavelet)

Image	PSNR (dB)	The PSNR of sub-images (dB)				Inhomogeneity
		0	1	2	3	
AIRFIELD	28.31	28.98	28.40	28.44	27.54	1.44
BABOON	26.56	27.73	26.58	26.47	25.70	1.97
BARBARA	30.26	32.14	30.68	30.16	28.74	3.40
BRIDGE	27.00	28.12	27.05	26.97	26.10	2.02
CLOCK	46.68	47.19	46.49	46.85	46.23	0.96
COUPLE	32.04	33.42	31.92	32.24	30.94	2.48
CROWD	33.08	34.37	33.32	33.16	31.86	2.51
GIRL	36.82	37.61	36.49	37.55	35.87	1.74
GOLD	32.79	33.78	32.93	32.69	31.94	1.84
HARBOR	29.78	30.93	29.74	29.78	28.92	2.01
LAX	27.29	28.15	27.37	27.41	26.41	1.74
LENA	36.62	37.51	36.72	36.85	35.60	1.91
MAN	32.57	33.79	32.54	32.69	31.55	2.24
PAOLINA	37.05	37.49	37.08	37.08	36.58	0.91
PEPPERS	35.63	36.14	35.66	35.78	35.03	1.11
SAILBOAT	31.08	32.18	31.12	31.25	30.05	2.13
TIFFANY	36.80	37.90	36.87	36.79	35.87	2.03
WOMAN	42.05	42.91	42.09	42.19	41.20	1.71
ZEBRA	25.98	27.26	25.49	26.75	24.86	2.40
Statistical P-value						10^{-11}

is $e = x(n) - \bar{x}(n)$. For simplicity of notation we denote the upsampled and downsampled by a factor k versions of $f(n)$ as $f(\frac{n}{l})$ and $f(ln)$ respectively. We use $l = 2$ due to popularity of two-band filters in wavelet compression.

Figure 3.4: Block scheme of wavelet analysis and synthesis operations

Upsampling in the synthesis part of the process leads to the following interesting observation. When computing even samples of $\bar{x}(n)$, the odd coefficients

Table 3.4: Comparison of sub-image PSNR for 19 grayscale test images (bitrate = 0.50 bpp, JPEG2000, 5-3 wavelet)

Image	PSNR (dB)	The PSNR of sub-images (dB)				Inhomogeneity
		0	1	2	3	
AIRFIELD	27.89	28.63	28.00	27.97	27.09	1.54
BABOON	26.40	27.66	26.33	26.40	25.48	2.18
BARBARA	30.86	31.81	31.03	30.79	30.01	1.80
BRIDGE	26.82	28.14	26.95	26.66	25.85	2.29
CLOCK	45.92	45.92	46.19	45.57	46.01	-0.09
COUPLE	32.27	33.03	32.36	32.30	31.50	0.63
CROWD	32.83	33.75	32.99	32.94	31.87	1.88
GIRL	36.77	37.32	36.54	37.29	36.06	1.26
GOLD	32.62	33.61	32.70	32.55	31.81	1.80
HARBOR	29.66	30.94	29.60	29.64	28.73	2.21
LAX	27.03	27.96	27.24	27.08	26.07	1.89
LENA	36.33	36.92	36.50	36.43	35.58	1.34
MAN	32.28	33.29	32.35	32.32	31.37	1.92
PAOLINA	36.64	37.08	36.72	36.61	36.19	0.89
PEPPERS	35.27	35.72	35.32	35.33	34.74	0.98
SAILBOAT	31.05	32.15	31.01	31.21	30.08	2.07
TIFFANY	36.53	37.16	36.58	36.49	35.95	1.21
WOMAN	41.23	41.63	41.34	41.26	40.74	0.89
ZEBRA	25.99	26.95	25.66	26.47	25.11	1.84
Statistical P-value						10^{-8}

of g_0 filter are multiplied by zero, as shown in Figure 3.5. On the other hand, when computing odd samples of $\bar{x}(n)$ it is the even coefficients of g_0 filter that are zeroed out. Because g_1 filter has one time-unit shift compared to g_0 , a reverse relationship is true for g_1 , where only odd filter coefficients contribute to even samples of $\bar{x}(n)$ and vice versa.

Let's denote a sequence of even and odd samples of a sequence $f(n)$ as $\hat{f} = f(2n)$ and $\check{f} = f(2n + 1)$ respectively. Then we can describe the synthesis process as

$$\begin{aligned}\hat{x} &= \hat{g}_0 * \bar{w}_0 + \check{g}_1 * \bar{w}_1 \\ \check{x} &= \check{g}_0 * \bar{w}_0 + \hat{g}_1 * \bar{w}_1\end{aligned}\tag{3.1}$$

Table 3.5: Comparison of sub-image PSNR in different bit-rates for “Barbara” (SPIHT, 9-7 wavelet)

Bitrate (bit/pixel)	PSNR (dB)	The PSNR of sub-images (dB)				Inhomogeneity
		0	1	2	3	
0.10	24.26	24.69	23.84	24.61	23.95	0.74
0.20	26.65	27.64	26.12	27.05	26.00	1.64
0.30	28.56	29.66	28.13	28.72	27.92	1.74
0.40	30.10	31.22	30.10	30.05	29.26	1.96
0.50	31.39	32.78	31.49	31.15	30.47	2.31
0.60	32.53	33.70	32.76	32.06	31.83	1.87
0.70	33.75	34.62	33.78	33.79	32.95	1.67
0.80	34.66	35.87	34.84	34.57	33.65	2.22
0.90	35.59	36.90	35.82	35.44	34.54	2.36
1.00	36.41	37.51	36.69	36.07	35.61	1.91
Statistical P-value						10^{-7}

Figure 3.5: Filtering process (*) in wavelet synthesis.

where $*$ represents the filtering operation. According to the perfect reconstruction property of wavelet transform, the only source of error in $\bar{x}$ is due to quantization of wavelet coefficients. Hence for the reconstruction error we can write

$$\begin{aligned}\hat{e} &= \hat{g}_0 * q_0 + \check{g}_1 * q_1 \\ \check{e} &= \check{g}_0 * q_0 + \hat{g}_1 * q_1\end{aligned}\tag{3.2}$$

Let $q_{i,n}(k) = q_i(n+k)$ be a circular shift of sequence q_i by n samples. Treating sequences $q_{i,n}$ (fixed n) and g_i as row vectors, we can rewrite the n -th error

sample of (3.2) in a matrix form:

$$\begin{aligned}\hat{e}(n) &= \hat{g}_0 q_{0,n}^T + \check{g}_1 q_{1,n}^T \\ \check{e}(n) &= \check{g}_0 q_{0,n}^T + \hat{g}_1 q_{1,n}^T\end{aligned}\tag{3.3}$$

From here the error variance is

$$\begin{aligned}\sigma_e^2 &= E\{\hat{e}^2(n)\} = \hat{g}_0 E\{q_{0,n}^T q_{0,n}\} \hat{g}_0^T + \check{g}_1 E\{q_{1,n}^T q_{1,n}\} \check{g}_1^T + 2\hat{g}_0 E\{q_{0,n}^T q_{1,n}\} \check{g}_1^T \\ \sigma_e^2 &= E\{\check{e}^2(n)\} = \check{g}_0 E\{q_{0,n}^T q_{0,n}\} \check{g}_0^T + \hat{g}_1 E\{q_{1,n}^T q_{1,n}\} \hat{g}_1^T + 2\check{g}_0 E\{q_{0,n}^T q_{1,n}\} \hat{g}_1^T\end{aligned}\tag{3.4}$$

where the expectation is taken over all samples of sequence e . This expression is the key to understanding of error inhomogeneity: the error variances of even and odd samples of the reconstructed signal depend on different synthesis filter coefficients, and hence in general are not equal to each other.

In what follows, we consider several special cases and show that not only $\sigma_e^2 \neq \sigma_e^2$ but that σ_e^2 is consistently smaller than σ_e^2 regardless of the image content.

3.4 Error Inhomogeneity Prediction

Let $\xi_i^2 = E\{q_i^2(n)\}$ and $\Sigma_i = E\{q_{i,n}^T q_{i,n}\}$, $i = 0, 1$ be the variance and covariance matrix of the quantization errors, $\Sigma_{01} = E\{q_{0,n}^T q_{1,n}\}$ be the cross covariance matrix between the quantization errors in low and high frequency bands. Note that $\Sigma_0 \in \mathbb{R}^{K_0 \times K_0}$, $\Sigma_1 \in \mathbb{R}^{K_1 \times K_1}$, and $\Sigma_{01} \in \mathbb{R}^{K_0 \times K_1}$, where K_0 and K_1 are the

half sizes of filters g_0 and g_1 respectively. Then we can rewrite (3.4) as

$$\begin{aligned}\sigma_{\hat{e}}^2 &= \hat{g}_0 \Sigma_0 \hat{g}_0^T + \check{g}_1 \Sigma_1 \check{g}_1^T + 2\hat{g}_0 \Sigma_{01} \check{g}_1^T \\ \sigma_{\check{e}}^2 &= \check{g}_0 \Sigma_0 \check{g}_0^T + \hat{g}_1 \Sigma_1 \hat{g}_1^T + 2\check{g}_0 \Sigma_{01} \hat{g}_1^T\end{aligned}\quad (3.5)$$

We define error inhomogeneity EI as the difference in PSNR between even and odd samples of the reconstructed signal:

$$\begin{aligned}EI &= \text{PSNR}_{\hat{e}} - \text{PSNR}_{\check{e}} \\ &= 10 \log_{10} \frac{peak_s^2}{\sigma_{\hat{e}}^2} - 10 \log_{10} \frac{peak_s^2}{\sigma_{\check{e}}^2} \\ &= 10 \log_{10} \frac{\sigma_{\check{e}}^2}{\sigma_{\hat{e}}^2}\end{aligned}\quad (3.6)$$

where $peak_s$ is the peak value of the signal. In Sections 3.4.1-3.4.3, we show how to compute EI for several different assumptions on the statistics of quantization errors.

3.4.1 Case 1: Uniform uncorrelated quantization errors

We first assume that the quantization errors are uniform and uncorrelated, i.e.

$\Sigma_{01} = 0$, $\Sigma_i = \xi_i^2 I$ and $\xi_0^2 = \xi_1^2 = \xi^2$. Then from (3.5) we obtain

$$\begin{aligned}\sigma_{\hat{e}}^2 &= \xi^2 \left[\sum_k \hat{g}_0^2(k) + \sum_k \check{g}_1^2(k) \right] = \xi^2 \left[\hat{V}_0(0) + \check{V}_1(0) \right] \\ \sigma_{\check{e}}^2 &= \xi^2 \left[\sum_k \check{g}_0^2(k) + \sum_k \hat{g}_1^2(k) \right] = \xi^2 \left[\check{V}_0(0) + \hat{V}_1(0) \right]\end{aligned}\quad (3.7)$$

where $V(\cdot)$ is proportional to the autocorrelation function of $g(k)$, such that $V(0) = \sum_k g^2(k)$ and $V(1) = \sum_k g(k)g(k+1)$. Then from (3.6)

$$EI = 10 \log_{10} \frac{\check{V}_0(0) + \hat{V}_1(0)}{\hat{V}_0(0) + \check{V}_1(0)} \quad (3.8)$$

and we see that inhomogeneity depends only on the values of the wavelet filter coefficients. The L_2 norms of low and high pass wavelet synthesis filters for several popular wavelets are shown in Table 3.6. According to this data, $\hat{V}_0(0) > \check{V}_0(0)$, $\hat{V}_1(0) > \check{V}_1(0)$, and $\hat{V}_0(0) - \check{V}_0(0) < \hat{V}_1(0) - \check{V}_1(0)$, leading to positive error inhomogeneity. This is confirmed by the experimental results in Table 3.7.

The experimental results also show that the equation (3.8) slightly underestimates the inhomogeneity for most wavelets, which is likely due to violation of uniformity and uncorrelatedness assumption. To better explain the experimental results we relax the uniformity assumption in Section 3.4.2.

Table 3.6: Norms of even/odd samples of synthesis wavelet filters

Wavelet	$\hat{V}_0(0)$	$\check{V}_0(0)$	$\hat{V}_1(0)$	$\check{V}_1(0)$
5-3	0.5000	0.2500	1.1875	0.2500
9-3	0.5000	0.2500	1.0535	0.3613
9-7	0.6250	0.3579	0.7544	0.2860
17-11	0.5796	0.3620	0.7054	0.3664

Table 3.7: Predicted (3.6) and empirically evaluated (over 19 test images) error inhomogeneity (in dB) for several popular wavelets

Wavelet	Predicted (3.6)	Experimental
5-3	2.8255	2.754
9-3	1.7993	1.927
9-7	0.8672	1.307
17-11	0.5243	0.979

3.4.2 Case 2: Nonuniform uncorrelated quantization errors

Optimal assignment of bitrate to wavelet subbands may lead to nonuniform subband quantization errors. Using the quantization model [39, 42] it can be shown that at high bitrates the optimal bitrate allocation leads to

$$r = \frac{\xi_0^2}{\xi_1^2} = \frac{V_1(0)}{V_0(0)} \neq 1 \quad (3.9)$$

For all tested wavelets $r > 1$, as follows from Table 3.6. Specifically, for 5-3 and 9-3 wavelets r is close to 2, while for 9-7 and 17-11 wavelets r is close to 1. We observed that the ratio r becomes larger for low bitrates where the quantization model [42] no longer provides a good approximation, see Figure 3.6. Low bitrate compression leads to almost complete loss of high frequencies (quantization error becomes approximately constant and equal to the variance of wavelet coefficients). On the other hand, the quantization error variance of low frequencies ξ_0^2 will continue to increase, leading to larger r at low bitrates. Regardless of the bitrate, 5-3 and 9-3 wavelets still yield larger r compared with 9-7 and 17-11 wavelets, as predicted by (3.9).

To account for nonuniform quantization errors we relax our previous assumption and allow $\xi_0^2 \neq \xi_1^2$. Then from (3.5) we obtain

$$\begin{aligned} \sigma_{\tilde{e}}^2 &= \xi_1^2 \left(r \hat{V}_0(0) + \check{V}_1(0) \right) \\ \sigma_{\tilde{e}}^2 &= \xi_1^2 \left(r \check{V}_0(0) + \hat{V}_1(0) \right) \end{aligned} \quad (3.10)$$

Figure 3.6: Ratio r between quantization error variances

which leads to error inhomogeneity

$$EI = 10 \log_{10} \frac{r\check{V}_0(0) + \hat{V}_1(0)}{r\hat{V}_0(0) + \check{V}_1(0)} \quad (3.11)$$

When $r = 1$ (see Section 3.4.1), EI is positive for all selected wavelets. However, since for tested wavelets $\hat{V}_0(0) > \check{V}_0(0)$, as follows from Table 3.6, multiplication by $r > 1$ in (3.11) makes inhomogeneity smaller. For low bitrates when r becomes very large, our model (3.11) predicts that the inhomogeneity can even become negative, as shown in Figure 3.7(a).

The decreasing inhomogeneity trend is confirmed by the experimental results in Figure 3.7(b). However, experiments also show that inhomogeneity remains positive even for $r > 10$, which contradicts the model (3.11). This inconsistency can be resolved by relaxing the uncorrelatedness assumption, as we show in the next section.

Figure 3.7: Error homogeneity as a function of r for several wavelets: (a) predicted using (3.11) (b) evaluated by averaging over 19 test images

3.4.3 Case 3: Nonuniform and correlated quantization error

Since the wavelet transform seeks to decorrelate the image, we can assume the quantization error correlations to be small. For simplicity, we only consider the correlation between the quantization errors of neighboring coefficients by assuming $E\{q_i(n)q_i(n+k)\} = 0$ for $k > 1$. Let $c_i = \frac{E\{q_i(n)q_i(n+1)\}}{\xi_i^2}$. Then from (3.5) we obtain

$$\begin{aligned}\sigma_e^2 &= \xi_0^2 \hat{V}_0(0) + \xi_1^2 \check{V}_1(0) + 2c_0 \xi_0^2 \hat{V}_0(1) + 2c_1 \xi_1^2 \check{V}_1(1) \\ \sigma_e^2 &= \xi_0^2 \check{V}_0(0) + \xi_1^2 \hat{V}_1(0) + 2c_0 \xi_0^2 \check{V}_0(1) + 2c_1 \xi_1^2 \hat{V}_1(1)\end{aligned}\quad (3.12)$$

The error homogeneity is then

$$EI = 10 \log_{10} \frac{r \check{V}_0(0) + \hat{V}_1(0) + 2c_0 r \check{V}_0(1) + 2c_1 \hat{V}_1(1)}{r \hat{V}_0(0) + \check{V}_1(0) + 2c_0 r \hat{V}_0(1) + 2c_1 \check{V}_1(1)} \quad (3.13)$$

According to Table 3.8, for tested wavelets $\hat{V}_i(1) \leq 0$ and $\check{V}_i(1) \geq 0$. Typically in low frequency subband the correlation between neighboring wavelet coefficients tends to be positive, while in high frequency subband it is negative. According to our experiments, the quantization error correlations follow the same trend, i.e. $c_0 \geq 0$ and $c_1 \leq 0$, as shown in Figure 3.8. Hence $c_0 r \hat{V}_0(1) + c_1 \check{V}_1(1) \leq 0$ and $c_0 r \check{V}_0(1) + c_1 \hat{V}_1(1) \geq 0$, making σ_ε^2 smaller and σ_ε^2 larger, thus increasing the error inhomogeneity. As shown in Figure 3.9, prediction using (3.13) provides a much better fit to actual inhomogeneity compared to results of (3.11). In particular, the modeled inhomogeneity is positive even for low bitrates, which is an improvement over (3.11). It is likely that even better fit can be obtained by taking second and higher-order correlations into account.

Table 3.8: Values of $\hat{V}_i(1)$ and $\check{V}_i(1)$ wavelet parameters

Wavelet	$\hat{V}_0(1)$	$\check{V}_0(1)$	$\hat{V}_1(1)$	$\check{V}_1(1)$
5-3	0	0.2500	-0.1875	0.0625
9-3	0	0.1250	-0.3633	0.1206
9-7	-0.0642	0.1208	-0.1970	0.1244
17-11	-0.0624	0.1065	-0.1665	0.1102

(a) In low-low frequency band

(b) In high-high frequency band

Figure 3.8: Correlation between quantization errors of neighboring coefficients

Figure 3.9: Predicted inhomogeneity under various assumptions on the statistics of quantization errors. Here we extend our 1D prediction equations to 2D by applying them to both horizontal and vertical directions and take the reconstruction error in the LL subband (due to multi-scale decomposition) as the quantization error.

To summarize, we have shown that the main contribution to error inhomogeneity comes from the unbalance between even and odd parts of synthesis wavelet filters. Additional bitrate-dependent contributions come from nonuniformity and correlation of quantization errors. The derivations can be extended to account for multi-level decompositions, at a cost of substantial increase in complexity, as exemplified in Appendix A. While being able to predict the amount of inhomogeneity is sufficient for some applications, it might be necessary to remove or at least decrease the value of inhomogeneity. This can be accomplished by either redesigning wavelet filters or changing quantization error statistics via a proper bitrate allocation strategy. The latter approach leads to a complex nonlinear optimization problem that appears intractable. Hence we seek to reduce the inhomogeneity by designing a new balanced wavelet filter in Section 3.5.

3.5 Balanced Wavelet Design

For our design we have chosen Daubechies biorthogonal wavelets [63]. Orthogonal wavelets are less popular due to their non-linear phase, leading to visually unacceptable phase distortion near edges [75]. In addition to having linear phase, Daubechies biorthogonal wavelets maximize the number of vanishing moments for a given filter length. This has made them very popular in image compression applications [40, 76]; in particular, 9-7 and 5-3 Daubechies wavelets are the only ones adopted by JPEG2000 standard [38].

Let $G_i(z)$ be the $\mathcal{Z}$ -transforms of the synthesis filters g_i for $i = 0, 1$. Then the $\mathcal{Z}$ -transform of the analysis filters h_i are obtained by quadrature mirroring of the synthesis filters [77]:

$$\begin{aligned} H_0(z) &= z^{-1}G_1(-z) \\ H_1(z) &= z^{-1}G_0(-z) \end{aligned} \tag{3.14}$$

Let $P(z) = H_0(z)G_0(z)$, then for perfect reconstruction (PR) [77] we need the following condition:

$$P(z) + P(-z) = 2 \tag{3.15}$$

Designing a wavelet means finding a low-pass filter P satisfying the PR condition (3.15) and then factorizing P into the analysis and synthesis filters H_0 and G_0 . For example, the Daubechies wavelets are designed by factorizing the Lagrange half-band filters (LHBF), which have zero of the largest order at $z = -1$ [63]:

$$P(z) = L_K(z) = z^K \left(\frac{1+z^{-1}}{2} \right)^{2K} R_K(z) \tag{3.16}$$

Here $R_K(z)$ is the lowest order polynomial for which $L_K(z)$ satisfies the PR condition. The order of zero at $z = -1$ is proportional to the number of vanishing moments, and hence $L_K(z)$'s achieve maximally flat frequency response for given filter length:

$$\left. \frac{d^q P(e^{j\omega})}{d\omega^q} \right|_{\omega=\pi} = 0, \text{ for } q = 1, \dots, 2K - 1 \quad (3.17)$$

By setting $Z = \cos \omega = (z + z^{-1})/2$, we can rewrite (3.16) as

$$L_K(Z) = (Z + 1)^K R_K(Z) \quad (3.18)$$

Using this variable substitution guarantees linear phase for any factorization of (3.18).

Our goal is to design a balanced wavelet with reduced (or even zero) error inhomogeneity. Ideally, for the nonuniform and correlated model (3.13), the designed wavelet has to satisfy the following condition:

$$r\check{V}_0(0) + \hat{V}_1(0) + 2c_0r\check{V}_0(1) + 2c_1\hat{V}_1(1) = r\hat{V}_0(0) + \check{V}_1(0) + 2c_0r\hat{V}_0(1) + 2c_1\check{V}_1(1) \quad (3.19)$$

Clearly, a factorization of LHBF does not necessarily satisfy the condition (3.19). One possible solution is to release one order of zero at $Z = -1$ in LHBF (3.18):

$$L'_K(Z) = (Z + 1)^{K-1}(Z + v)R'_K(Z) \quad (3.20)$$

where v is a free parameter that can be tuned to satisfy the additional constraint, and $R'_K(Z)$ is set to guarantee the PR condition (3.15). A similar idea has recently been used by Tay [41] to rationalize the biorthogonal wavelet filters.

The balance condition (3.19) is difficult to satisfy in general, since the parameters r and c_i depend on bitrate and image content. Hence, without loss of generality, we design a wavelet using a simplified balance condition derived from the uniform and uncorrelated model (3.8):

$$\hat{V}_0(0) + \check{V}_1(0) = \check{V}_0(0) + \hat{V}_1(0) \quad (3.21)$$

However, if necessary, a similar design procedure can be carried out for any balance condition, including (3.19) and its possible higher order extensions.

In Sections 3.5.1-3.5.2 we explain the proposed design procedure on the example of 9-7 and 9-3 wavelets. We have chosen the 9-7 wavelet for its popularity and the 9-3 wavelet because it leads to large inhomogeneity (see Table 3.7) while having zero of the third order at $Z = -1$, allowing one order to be released.

3.5.1 Balancing 9-7 wavelet

The original 9-7 wavelet¹ is obtained by factorizing LHBF for $K = 4$. Releasing one order of zero yields

$$\begin{aligned} H_0 &= t_0(Z+1)(Z^3 + AZ^2 + BZ + C) \\ G_0 &= t_1(Z+1)^2(Z + \gamma) \end{aligned} \quad (3.22)$$

where t_0 and t_1 are the normalization parameters, and the free factor $(Z + v)$ has been absorbed by $(Z^3 + AZ^2 + BZ + C)$. Since the product of the two filters $P(Z) = L'_K(Z) = H_0G_0$ should satisfy the PR condition (3.15), the coefficients

¹The 9-7 wavelet has been used by Tay [41], and hence our description follows a structure similar to [41] but with a different constraint.

Table 3.9: Balanced 9-7 wavelet filter coefficients

Analysis filter h_0		Synthesis filter g_0	
$h_0(0)$	0.78052828771805	$g_0(0)$	0.82778141134444
$h_0(-1, 1)$	0.39420312029991	$g_0(-1, 1)$	0.42791622195691
$h_0(-2, 2)$	-0.08680958499528	$g_0(-2, 2)$	-0.06033731507895
$h_0(-3, 3)$	-0.04064972970664	$g_0(-3, 3)$	-0.07436283136363
$h_0(-4, 4)$	0.05009883172953	—	—

in front of Z^2 , Z^4 and Z^6 must be zero. This yields three constraints for four unknowns A , B , C and γ :

$$\begin{aligned}
A &= -(3 + \gamma) \\
B &= \frac{9\gamma^3 + 35\gamma^2 + 48\gamma + 24}{3\gamma^2 + 9\gamma + 8} \\
C &= \frac{8(1 + \gamma)^2}{3\gamma^2 + 9\gamma + 8}
\end{aligned} \tag{3.23}$$

Original 9-7 wavelet specifies an additional constraint by requiring $(Z^3 + AZ^2 + BZ + C)$ to have a zero at $Z = -1$, leading to $\gamma = -1.68$. For a balanced wavelet, an additional constraint comes directly from the balance condition .

The balance condition (3.21) can be translated to a constraint on γ as follows. We first use (3.22) and (3.23) to express the coefficients of $h_0(n)$ and $g_0(n)$ filters in terms of γ . Then using the quadrature mirror property (3.14) allows us to express $g_0(n)$ and $g_1(n)$ in terms of γ . Substituting $g_0(n)$ and $g_1(n)$ into the balance condition (3.21) yields a nonlinear equation that can be solved for γ . The resultant equation has two real roots, $\gamma = -1.59$ and $\gamma = -0.24$. We chose the first root $\gamma = -1.59$, because it leads to a more regular wavelet that has smoother and less noticeable compression errors [78]. Table 3.9 shows the values of the derived wavelet filter coefficients.

Table 3.10 compares the compression performance (overall PSNR and er-

ror inhomogeneity) between the original and the new balanced 9-7 wavelets. The balanced wavelet reduces the error inhomogeneity by a factor of two, on average. However, it also slightly reduced the overall PSNR performance (by approximately 0.1dB), due to fewer vanishing moments (3 vs. 4 for the original wavelet). The decrease in inhomogeneity can be observed visually (Figure 3.10) — sub-images 0 and 3 now look more similar to each other.

Table 3.10: PSNR performance and inhomogeneity (in dB) for original and new balanced 9-7 wavelets (0.50 bpp, SPIHT)

Image	Overall PSNR		Error inhomogeneity	
	original	balanced	original	balanced
AIRFIELD	28.57	28.49	1.22	0.62
BABOON	27.03	26.94	1.28	0.51
BARBARA	31.39	31.44	2.31	1.65
BRIDGE	27.27	27.16	1.19	0.41
CLOCK	47.36	47.26	1.30	0.77
COUPLE	32.45	32.33	1.49	0.40
CROWD	33.88	33.66	1.42	0.62
GIRL	37.46	37.33	1.46	0.76
GOLD	33.08	32.94	1.03	0.29
HARBOR	30.01	29.96	1.45	0.50
LAX	27.53	27.43	0.96	0.17
LENA	37.21	37.10	1.33	0.69
MAN	33.08	32.93	1.36	0.59
PAOLINA	37.13	37.07	0.65	0.14
PEPPERS	35.81	35.73	0.79	0.32
SAILBOAT	31.57	31.46	1.54	0.83
TIFFANY	37.25	37.22	1.29	0.63
WOMAN	42.54	42.47	1.36	0.72
ZEBRA	26.77	26.66	1.41	0.69
Average	33.55	33.45	1.31	0.61
Statistical P-value			10^{-12}	10^{-7}

Figure 3.10: Reconstructed sub-images (bitrate 0.20bpp): (a,b) original 9-7 wavelet, (c, d) balanced 9-7 wavelet.

3.5.2 Balancing 9-3 wavelet

The original 9-3 wavelet is obtained by factorizing LHBF for $K = 3$. Releasing one order of zero yields

$$\begin{aligned} H_0 &= t_0(Z+1)(Z^3 + AZ^2 + BZ + \gamma) \\ G_0 &= t_1(Z+1) \end{aligned} \quad (3.24)$$

According to the PR condition (3.15), the coefficients of Z^2 and Z^4 in $P(Z) = L'_K(Z) = H_0 G_0$ must be zero, leading to two constraints for unknown A , B and γ :

$$\begin{aligned} A &= -2 \\ B &= 1 - \frac{\gamma}{2} \end{aligned} \quad (3.25)$$

For the original 9-3 wavelet an additional constraint arises from requiring $(Z^3 + AZ^2 + BZ + \gamma)$ to have a zero at $Z = -1$, leading to $\gamma = 1.33$. For a balanced 9-3 wavelet, an additional constraint comes from the balance condition (3.21). Solving resultant nonlinear equation yields two roots, $\gamma = -0.11$ and $\gamma = 1.02$. We have chosen the second root $\gamma = 1.02$ due to its better regularity. The values of the derived wavelet filter coefficients are shown in Table 3.11. Table 3.12 shows that the designed balanced wavelet reduces the inhomogeneity by a factor of ten to only 0.19dB, on average. This comes at a cost of reducing the overall PSNR by 0.34dB.

Our experiments show that reduction in inhomogeneity is correlated with the decrease in overall performance, i.e. the larger inhomogeneity reduction leads to larger drop in overall performance. For applications that are sensitive

Table 3.11: Balanced 9-3 wavelet filter coefficients

Analysis filter h_0		Synthesis filter g_0	
$h_0(0)$	0.88742610988342	$g_0(0)$	0.70710678118655
$h_0(-1, 1)$	0.52678745248968	$g_0(-1, 1)$	0.35355339059327
$h_0(-2, 2)$	-0.17677669529664	—	—
$h_0(-3, 3)$	-0.17323406189640	—	—
$h_0(-4, 4)$	0.08661703094820	—	—

to performance degradation, it might be desirable to design a wavelet that gradually trades off the overall performance for reduction in error inhomogeneity. The details of such a design are provided in the next section.

3.6 Trade-off between Overall Performance and Error Inhomogeneity

To establish the relationship between the error inhomogeneity and the overall performance we model them as functions of a single free parameter γ that arises from releasing one order of zero in factorizing LHBF, see Sections 3.5.1 and 3.5.2. Error inhomogeneity depends on the synthesis filter coefficients via (3.8) or (3.13), and through them on γ . The dependency of the overall compression performance on the wavelet filter design can be captured by the coding gain CG [40, 41, 44, 46, 47]. According to the model (1.1) proposed by Katto and Yasuda [39],

$$CG = 10 \log_{10} \Pi_{k=0}^{N-1} (A_k B_k)^{-\alpha_k} \quad (3.26)$$

where A_k and B_k are defined in (1.3) and (1.4), respectively. Note that $h_k(n)$ and $g_k(n)$ are the equivalent filters for the pyramid-decomposition structure. In this chapter, we adopt a five-level pyramid-transform and set the signal inter-pixel correlation $\rho = 0.95$ as suggested by Lightstone *et al.* [40]. Expressing

Table 3.12: PSNR performance and inhomogeneity (in dB) for original and new balanced 9-3 wavelets (0.50 bpp, SPIHT)

Image	Overall PSNR		Error inhomogeneity	
	original	balanced	original	balanced
AIRFIELD	28.31	27.90	1.44	0.07
BABOON	26.56	26.12	1.97	0.28
BARBARA	30.26	30.00	3.40	1.15
BRIDGE	27.00	26.61	2.02	0.36
CLOCK	46.38	47.26	0.96	-0.34
COUPLE	32.04	31.66	2.48	0.77
CROWD	33.08	32.64	2.51	0.45
GIRL	36.82	36.56	1.74	-0.23
GOLD	32.79	32.41	1.84	0.08
HARBOR	29.78	29.45	2.01	-0.20
LAX	27.29	26.87	1.74	0.16
LENA	36.62	36.27	1.91	0.12
MAN	32.57	32.16	2.24	0.45
PAOLINA	37.05	36.82	0.91	-0.22
PEPPERS	35.63	35.35	1.11	-0.17
SAILBOAT	31.08	30.79	2.13	0.31
TIFFANY	36.80	36.42	2.03	0.25
WOMAN	42.05	41.78	1.71	-0.03
ZEBRA	25.98	25.70	2.40	0.57
Average	33.07	32.73	1.93	0.19
Statistical P-value			10^{-12}	0.016

wavelet filter coefficients in terms of γ and plugging them into (3.26) yields $CG = CG(\gamma)$.

Figure 3.11 shows EI and CG as functions of γ for the modified 9-7 wavelet. As expected, the maximum of CG occurs at $\gamma = -1.68$, which corresponds to the original 9-7 wavelet. For $\gamma > -1.68$ both the CG and EI are monotonically decreasing, which confirms our observation that the more we reduce the error inhomogeneity the larger the performance degrades.

Using estimated $CG(\gamma)$ and $EI(\gamma)$ one can gradually trade-off the compression performance for inhomogeneity reduction. Instead of simply minimizing

Figure 3.11: Plots of coding gain (CG) and error inhomogeneity (EI) of modified 9-7 wavelet as functions of γ .

inhomogeneity, we can design a wavelet that minimizes the following cost:

$$Cost = w_{EI}|EI| - w_{CG}CG \quad (3.27)$$

where $w_{CG} > 0$ and $w_{EI} > 0$ are constant weights that can vary based on application. For example, choosing larger w_{EI} puts more emphasis on inhomogeneity reduction while increasing w_{CG} favors compression performance preservation. The examples of trade-off performance for several values of w_{CG}/w_{EI} are shown in Table 3.13. As expected, varying the ratio w_{CG}/w_{EI} allows us to construct the wavelets with varying degrees of inhomogeneity and compression performance reduction.

3.7 Applications

Error inhomogeneity in alternating pixels, while difficult to notice in most images, can become visible in images that contain a lot of texture. For example, consider the missing texture on the right knee of Barbara in Figure 3.12(a).

Table 3.13: Trade-off between PSNR performance and error inhomogeneity — EI (in dB) by wavelet design (0.50 bpp and SPIHT)

w_{CG}/w_{BV}	9-7 wavelet			9-3 wavelet		
	γ	PSNR	EI	γ	PSNR	EI
0	-1.59	33.45	0.61	0.76	32.73	0.19
1	-	-	-	0.76	32.73	0.19
5	-1.59	33.45	0.61	0.89	32.99	1.22
10	-1.61	33.50	0.83	1.08	33.06	1.78
20	-1.64	33.53	1.06	-	-	-
∞	-1.69	33.54	1.31	1.17	33.07	1.93

Compressing the same image with a balanced wavelet brings back the missing texture, as shown in Figure 3.12(b).

Figure 3.12: Compression image using (a) original and (b) balanced 9-7 wavelets (bitrate = 0.25bpp, SPIHT)

The theory developed can also be useful in image resizing. Reducing the image size is usually done by downsampling, preceded by low-pass filtering to avoid aliasing. Interpolation is used when a new sample position doesn't coincide with any of the original pixels.

Consider an example of downsizing a wavelet compressed image by a factor of two. The sub-image 0 pixels have smaller errors compared to other pixels,

even after low-pass filtering², as follows from Table 3.14 and Figures 3.13 (c), (d). Taking this sub-image as the final result leads to visually more pleasing results compared to using the bilinear version of the Matlab function “imresize”, as follows from the comparison of Figures 3.13 (b) and (c).

Table 3.14: PSNR performance of compressed sub-images after low-pass filtering (0.50 bpp, JPEG2000, 9-7 wavelet)

Image	The PSNR of sub-images (dB)			
	0	1	2	3
AIRFIELD	33.60	33.42	33.39	33.22
BABOON	32.84	32.57	32.47	32.23
BARBARA	37.52	37.16	37.21	36.87
BRIDGE	32.74	32.47	32.42	32.16
CLOCK	53.40	53.15	53.23	52.99
COUPLE	37.62	37.38	37.29	37.06
CROWD	38.07	37.68	37.61	37.26
GIRL	42.57	42.13	42.38	41.96
GOLD	39.00	38.71	38.40	38.15
HARBOR	35.62	35.41	35.52	35.31
LAX	33.77	33.51	33.52	33.24
LENA	43.12	42.86	42.85	42.58
MAN	37.78	37.45	37.53	37.22
PAOLINA	44.36	44.20	44.07	43.92
PEPPERS	42.61	42.41	42.45	42.26
SAILBOAT	37.58	37.11	37.28	36.85
TIFFANY	43.20	42.96	42.97	42.76
WOMAN	48.37	48.09	48.06	47.79
ZEBRA	31.13	30.79	30.93	30.61
Average	39.21	38.92	38.93	38.65

As another example we consider image downsizing combined with nearest neighbor interpolation. If the downsizing ratio is close (but not equal) to two, the pixels in the upper left corner of the image will be consistently interpolated from the sub-image 3 pixels. On the other hand, the pixels in the lower right corner will be interpolated primarily from the sub-image 0. The resultant image will then exhibit a visually apparent quality inhomogeneity between image corners, that can be corrected using balanced wavelets, as shown in Figure 3.14.

²Here we used filter (0.01 − 0.00 − 0.04 0.00 0.29 0.50 0.29 0.00 − 0.04 − 0.00 0.01),

Downsampling by a factor of two using the bilinear version of the Matlab function “imresize”

(a) Original uncompressed image

(b) Wavelet compressed image

Sub-images after compression and low-pass filtering

(c) Sub-image 0

(d) Sub-image 3

Figure 3.13: The downsize example of compressed sub-images after low-pass filtering (bitrate = 0.25bpp, JPEG2000, 9-7 wavelet)

3.8 Conclusions

We have experimentally discovered that wavelet compressed images exhibit spatially inhomogeneous error, i.e. the error that depends on pixel location. The difference can be quite substantial (upto 3.4 dB in our experiments), and cannot be ignored for some applications. We show that the error inhomogeneity is due to the fact that different wavelet synthesis filter coefficients contribute to the er-

which is a default for Matlab function “imresize”.

Figure 3.14: Downsampling using “nearest neighbor” method of the Matlab function “imresize”

rors at even and odd signal locations. Nonuniform and correlated quantization errors also contribute to the error inhomogeneity, especially at lower bitrates. The developed model allows accurate prediction of error inhomogeneity when statistics of quantization error is known.

Error inhomogeneity can be reduced by a proper wavelet design. We have shown this by modifying the design of popular Daubechies 9-7 and 9-3 wavelets. The redesign is based on releasing one order of zero at $z = -1$ to satisfy a chosen balance condition. However, releasing the order of zero leads to fewer vanishing moments and hence to the reduction in the overall compression performance, which we found to be reasonably small. For applications that might be sensitive to performance degradation, we show how to design wavelets that gradually trade-off the compression performance for reduction in error inhomogeneity.

The theory developed can potentially be useful for image resizing, superresolution, and other applications that rely on detailed modeling of pixel intensity statistics.

Part II

Demosaicking

Chapter 4

Adaptive Filtering for Color Filter Array Demosaicking

4.1 Introduction

Good demosaicking is essential for obtaining full color images of high-quality from the Color Filter array (CFA) data. Currently, the demosaicking is often done inside the camera, which typically has limitations in terms of processing power. However, in RAW photography and cameras using an alternative processing chain, the demosaicking can be done on a computer, which has stirred up a lot of development in this area. Better demosaicking not only improves the visual image quality, but can also lead to a wider applicability of alternative DSC processing chain; see discussion in Chapter 2.

As we have mentioned in the thesis introduction, Section 1.3.1, demosaick-

ing is similar to a low-pass filtering and hence can degrade the high-frequency image content, i.e. oversmooth the edges. Based on an observation that the luminance is well separated from the chrominance in the spatial frequency domain of a Bayer CFA image, Alleysson *et al.* [7] has recently proposed an interesting frequency selection approach to preserving high-frequency information. Here luminance is defined as a spatial map of intensity and chrominance as a map of chromatic (color-difference) components. Hence an appropriately designed low-pass filter can obtain a full-resolution luminance from a CFA image (frequency selection [7]), which is then used as a reference to reconstruct the RGB components. However, the filter proposed in [7] unnecessarily eliminated high-frequency information along horizontal and vertical axes, which could be important for image quality, as we will demonstrate in Section 4.2.2. More details of Alleysson’s frequency selection (FS) demosaicking will be provided in Section 4.3.

Our frequency analysis of Bayer CFA samples (see Section 4.3) indicates that the important high-frequency information along the two axes is preserved, free of aliasing, in the sub-sampled green plane, and hence can be extracted by a properly designed filter. Our further analysis of the CFA spectrum explains why the FS demosaicking method [7] is reasonably good but not optimal.

Based on the conducted analysis, we propose in Section 4.4 an efficient low-pass filter for the green samples to better preserve the high-frequency information. The high-frequency information at red/blue pixels is then estimated by adaptive filtering of color-difference components (Section 4.5). Combining several techniques for preserving high-frequency information, our method outperforms existing state-of-the-art methods both visually and in terms of peak

signal-to-noise ratio (PSNR), as shown in Section 4.7. We also show that the proposed method has a lower computational complexity and rectify a fallacious assumption from FS demosaicking [7] in Section 4.8.

4.2 Key Observations about CFA Demosaicking

In this section we discuss two key observations that can help better preserve the high-frequency information in demosaicked images. The first observation is the similarity between high frequencies across color components (see Sections 1.3.3 and 4.2.1). This property suggests that any full-resolution color or luminance component can help reconstruct the high-frequency information for all other color components. The second observation is that preserving the high-frequency information along the horizontal and vertical axes is more important for the image quality than high-frequency preservation along the diagonal direction (Section 4.2.2).

4.2.1 Similarity of inter-color high-frequency content

In Section 1.3.3, we have shown that the high frequencies across three color components are similar to each other. This property has been well exploited in demosaicking. The most popular way of exploiting this property is through interpolation of the color-difference images. For example, this can be done by estimating a full-resolution green component first and then using it to predict the red and blue components from color-difference images [1, 13, 15, 49, 79–83].

The following lemma explains why this approach works.

Lemma 1. *Let $\mathcal{F}$ be a full-resolution reference color component. Then any other full-resolution color component $\mathcal{C} \in \{R, G, B\}$ can be predicted from its subsampled version $\mathcal{C}_s$ using*

$$\mathcal{C} \approx \mathcal{I}(\mathcal{C}_s - \mathcal{F}_s) + \mathcal{F} \quad (4.1)$$

where $\mathcal{F}_s$ is subsampled version of $\mathcal{F}$ and $\mathcal{I}$ denotes a proper interpolation process.

Proof: Let $\mathcal{F} = \mathcal{F}^l + \mathcal{F}^h$, where $\mathcal{F}^l$ and $\mathcal{F}^h$ are the low- and high-frequency components of $\mathcal{F}$, respectively. Let $\mathcal{C} = \mathcal{C}^l + \mathcal{C}^h$. Assuming the similarity of inter-color high-frequency components, we have $\mathcal{F}^h \approx \mathcal{C}^h$. Then the difference image $\mathcal{C} - \mathcal{F} \approx \mathcal{C}^l - \mathcal{F}^l$ contains mainly low-frequency information, and hence can be rather easily and accurately interpolated from the corresponding subsampled versions, i.e., $\mathcal{C}^l - \mathcal{F}^l \approx \mathcal{I}(\mathcal{C}_s^l - \mathcal{F}_s^l) \approx \mathcal{I}(\mathcal{C}_s - \mathcal{F}_s)$. It follows that

$$\mathcal{I}(\mathcal{C}_s - \mathcal{F}_s) + \mathcal{F} \approx \mathcal{C}^l - \mathcal{F}^l + \mathcal{F}^l + \mathcal{F}^h \approx \mathcal{C} \quad (4.2)$$

■

4.2.2 Importance of different high-frequency components

In natural images the energy spectrum is primarily concentrated in a low-frequency region, along the horizontal and vertical axes [52]. A typical example is shown in Figure 4.1. The following experiments further demonstrate that

(a) Original image

(b) Image spectrum

Figure 4.1: An image and its spectrum.

high-frequency components along the axes are more important for preserving image quality than those at the corners.

Shown in Figure 4.2 are the images which are obtained by removing some frequency components. We first removed the frequency coefficients outside the central circle [see Figure 4.2(a)], preserving 99.9%, 99.8% and 99.7% of the total energy; see results in Figure 4.2(b)-(d). Compare these with the images in Figure 4.2(f)-(h), which were obtained by removing the frequency coefficients outside a region that is more elongated along the two axes [see Figure 4.2(e)]. The latter set of images are more visually pleasing than the former, even though the total preserved energy is the same. For example the image in Figure 4.2(h) with 99.7% preserved energy appears similar to the image with 99.8% preserved energy in Figure 4.2(c).

This experiment implies that high-frequency information along the horizontal and vertical axes has larger impact on the reconstructed image quality than that at the corners. This happens because the human visual system is more sensitive to the information of these high frequencies [84–86].

As we will show in the next section, the high frequencies of the green com-

Figure 4.2: (b)-(d) images with removed frequency components outside centered circle, as shown in (a); (f)-(h) images with removed frequency components outside a region elongated along both axes, as shown in (e). Note that the ideal property of the filter leads to the ringing errors near the boundary of these images [6], which is not our concern.

ponent along the horizontal and vertical axes are free from aliasing in the Bayer CFA pattern, making it possible to better preserve this information in the demosaicked images.

4.3 Frequency Analysis of Bayer CFA Pattern

As shown in Figure 1.1(b)-(d), the Bayer CFA image can be decomposed into three sub-sampled color components. Mathematically, the sampling operations can be expressed via multiplication by a comb function $D_{\mathcal{C}}(m, n) = D(m - a_{\mathcal{C}}, n - b_{\mathcal{C}}) = \sum_{p=0}^{\frac{M-2}{2}} \sum_{q=0}^{\frac{N-2}{2}} \delta(m - a_{\mathcal{C}} - 2p, n - b_{\mathcal{C}} - 2q)$, where offsets $a_{\mathcal{C}}, b_{\mathcal{C}} \in \{0, 1\}$ depend on the color component $\mathcal{C} \in \{R, G, B\}$. It is convenient to rewrite

$D_{\mathcal{C}}(m, n)$ in terms of cosine modulation [7]. For example, for $a_{\mathcal{C}}, b_{\mathcal{C}} = 0$ we have

$$D(m, n) = \frac{[1 + \cos(\pi m)][1 + \cos(\pi n)]}{4} \quad (4.3)$$

The Fourier transform of $D(m, n)$ is

$$D(u, v) = \Delta(u)^T \begin{bmatrix} \frac{1}{16} & \frac{1}{8} & \frac{1}{16} \\ \frac{1}{8} & \frac{1}{4} & \frac{1}{8} \\ \frac{1}{16} & \frac{1}{8} & \frac{1}{16} \end{bmatrix} \Delta(v) \quad (4.4)$$

where $\Delta(u) = \left[\delta(u + \frac{1}{2}) \quad \delta(u) \quad \delta(u - \frac{1}{2}) \right]^T$ and T denotes the vector transposition. This expression can be easily extended to all three sampling functions $D_{\mathcal{C}}(u, v)$ for $\mathcal{C} \in \{R, G, B\}$. Let $\mathcal{C}_f(u, v)$ be the Fourier transform of the full-resolution color component $\mathcal{C}$. Then the Fourier transform of the subsampled color component, $\mathcal{C}_s(u, v)$, can be obtained by convolving $D_{\mathcal{C}}(u, v)$ with $\mathcal{C}_f(u, v)$:

$$R_s(u, v) = \Delta(u)^T \begin{bmatrix} -\frac{1}{16} & \frac{1}{8} & -\frac{1}{16} \\ -\frac{1}{8} & \frac{1}{4} & -\frac{1}{8} \\ -\frac{1}{16} & \frac{1}{8} & -\frac{1}{16} \end{bmatrix} \Delta(v) * R_f(u, v) \quad (4.5)$$

$$B_s(u, v) = \Delta(u)^T \begin{bmatrix} -\frac{1}{16} & -\frac{1}{8} & -\frac{1}{16} \\ \frac{1}{8} & \frac{1}{4} & \frac{1}{8} \\ -\frac{1}{16} & -\frac{1}{8} & -\frac{1}{16} \end{bmatrix} \Delta(v) * B_f(u, v) \quad (4.6)$$

$$G_s(u, v) = \Delta(u)^T \begin{bmatrix} \frac{1}{8} & 0 & \frac{1}{8} \\ 0 & \frac{1}{2} & 0 \\ \frac{1}{8} & 0 & \frac{1}{8} \end{bmatrix} \Delta(v) * G_f(u, v) \quad (4.7)$$

Here $*$ denotes the convolution operation. Note that the four zeros that appear in the middle term of the expression for $G_s(u, v)$ are due to cancellation of inverse phases in $D(m, n)$ and $D(m - 1, n - 1)$.

Expressions (4.5), (4.6) and (4.7) show that the Fourier transforms of the sampled components are scaled, periodic replications of the Fourier transforms of the full-resolution components. For the green component, the replication takes place only along the diagonal directions; for the red/blue components, the replication is also in the horizontal and vertical directions, as shown in Figure 4.3. This means that the high-frequency information along the horizontal and vertical axes in the subsampled green component will not be as much aliased as the other color components, making it an ideal choice to be used as a reference component for prediction (Lemma 1). This also helps shed light on why the green component (plane) is normally first fully interpolated and then used to estimate the other color components in many existing CFA demosaicking methods that exploit inter-color correlation.

Figure 4.3: Fourier transforms of (a) full-resolution color component, (b) quincuncial sampled green component, and (c) rectangular sampled red/blue component.

Furthermore, low-pass filtering of the whole CFA image instead of a single color component can better preserve the high-frequency information. Let $A(m, n)$ be the CFA image, which is a sum of three subsampled color components, $A(m, n) = R_s(m, n) + G_s(m, n) + B_s(m, n)$. It follows from (4.5), (4.6) and (4.7) that its Fourier transform is

$$A(u, v) = \mathbf{1}^T \begin{bmatrix} \frac{1}{16} [2G_f - R_f - B_f] (u + \frac{1}{2}, v + \frac{1}{2}) & \frac{1}{8} [R_f - B_f] (u + \frac{1}{2}, v) & \frac{1}{16} [2G_f - R_f - B_f] (u + \frac{1}{2}, v - \frac{1}{2}) \\ -\frac{1}{8} [R_f - B_f] (u, v + \frac{1}{2}) & \frac{1}{4} [2G_f + R_f + B_f] (u, v) & -\frac{1}{8} [R_f - B_f] (u, v - \frac{1}{2}) \\ \frac{1}{16} [2G_f - R_f - B_f] (u - \frac{1}{2}, v + \frac{1}{2}) & \frac{1}{8} [R_f - B_f] (u - \frac{1}{2}, v) & \frac{1}{16} [2G_f - R_f - B_f] (u - \frac{1}{2}, v - \frac{1}{2}) \end{bmatrix} \mathbf{1} \quad (4.8)$$

where $\mathbf{1} = [1, 1, 1]^T$.

Figure 4.4 shows the energy distribution in the CFA spectrum, following (4.8). Note that $L = \frac{1}{4}(2G + R + B)$ is the luminance component at the center of the spectrum, while $C1 = \frac{1}{8}(R - B)$ and $C2 = \frac{1}{16}(2G - R - B)$ are the color-difference (chrominance) components, shifted to the sides and corners, respectively. Since the color-difference components are smooth (see Section 4.2.1), they have a more compact spectrum that has less overlap with the luminance spectrum (Figure 4.4), as compared with the overlap between spectrum replicas in subsampled components (Figure 4.3).

Based on this property of CFA spectrum, Alleysson *et al.* [7] recently proposed a frequency selection (FS) approach for demosaicking, engaging an image-dependent low-pass filter to extract the full-resolution luminance L from the CFA image. The estimated luminance is then used as a reference to predict the RGB components (Lemma 1). Figure 4.5 shows an example of the filter used in [7], where the two stop bands $r1$ and $r2$ are used to remove $C1$ and $C2$, respectively. Comparing this filter with the low-pass filters used for single subsampled components (Figure 4.3), it is clear that the Alleysson's filter can

Figure 4.4: (a) Fourier transform and (b) a typical Fourier spectrum of a CFA image

better preserve high frequencies due to less overlap in the CFA spectrum.

Figure 4.5: (a) The passband and (b) Fourier transform of the low-pass filter in [7].

However, the Alleysson's filter falls short in several respects. First, since the filter depends on image content, it can be difficult to design when the original image is not available. Second, the filter discards more high-frequency information along the horizontal and vertical directions, when compared to the filter for the subsampled green component [see Figure 4.3(b)]. This can lead to degradation in image quality, as discussed in Section 4.2.2.

In the next section we propose a more efficient filter, which is inspired by the Alleysson's approach and capable of better preservation of the horizontal and vertical high frequencies.

4.4 Better Luminance Filtering at Green Samples

According to the CFA spectrum $A(u, v)$ in (4.8), the replicas of color-difference component $C1$ in the horizontal and vertical directions have the same amplitudes but inverse signs. If we subsample CFA just at green-pixel locations, the side components at two directions will be overlapped and canceled out each other,

$$A(u, v) * D_G(u, v) = G_s(u, v) \quad (4.9)$$

The resultant spectrum has no replicas at the sides, as shown in Figure 4.3(b). This suggests the following idea. If we remove the corner components $C2$ before subsampling the CFA image, the resultant spectrum will coincide with the luminance spectrum. Let $A'(u, v)$ be the spectrum with the corner components $C2$ removed, then

$$A'(u, v) * D_G(u, v) = \frac{1}{4}[2G + R + B](u, v) * D_G(u, v) = L(u, v) * D_G(u, v) \quad (4.10)$$

The advantage of this approach is that the low-pass filter removing corner components $C2$ has a quincuncial shape and hence preserves the important high-frequency information along the horizontal and vertical axes.

One example of such quincuncial filter is shown in Figure 4.3(b), which is

often used in bilinear interpolation. However, we can see that such a quincuncial filter yields poor performance because it also removes a part of component $C1$. This leads to much larger mean square error (MSE) of the estimated luminance, even compared to Alleysson's filter; see Table 4.1.

An alternative solution is to simply modify the Alleysson's filter with $r1 = 0$, as shown in Figure 4.8(a). This substantially reduces the MSE (see the third column in Table 4.1). Here, we iteratively computed the optimal parameters $r2$ to achieve the best MSE, as Alleysson *et al.* did [7]. Figure 4.6 shows that the optimal values of $r2$ are typical in a small range around $[0.08, 0.16]$. Moreover, the MSE degrades little around the optimal value; see Figure 4.7(a). This can be explained by the relatively little luminance energy in the spectrum's corners, and it effects less by adaptively adjusting the passband position in this direction. While a larger filter size can reduce more the MSE values, Figure 4.7(b) shows that the reduction becomes minimum when the filter size is larger than 5×5 . Hence, we fixed our filter size as 5×5 in order to reduce the computational cost. The spectrum and the pulse response of the proposed 5×5 filter are shown in Figure 4.8(b)(c).

Since the passband of the proposed filter is fixed, it does not depend on the image content. Despite this simplification, the performance of the redesigned filter only degrades slightly in terms of MSE (see the last column in Table 4.1). Computationally, it requires only 5 multiplications and 20 additions for each pixel, compared to 21 additions and 68 multiplications for the Alleysson's 11×11 filter.

However, the proposed low-pass filter is not optimal for estimating the luminance at red/blue pixel locations, because the side replicas of the $C1$ component

Figure 4.6: The optimal values of r_2 for 30 test images.

Table 4.1: MSE performance of the estimated green pixel luminance by using different filters.

Image No.	11×11 filter Quincuncial	11×11 filter Alleysson's	11×11 filter Alleysson's ($r_1 = 0$)	5×5 filter Proposed
1	24.3	9.04	1.37	1.86
2	6.15	2.62	0.98	0.85
3	4.61	1.64	0.88	0.88
4	5.36	2.37	1.07	1.09
5	27.15	4.85	2.05	2.54
6	18.45	7.14	0.90	1.25
7	5.04	1.99	0.72	0.77
8	38.83	21.73	2.06	2.59
9	5.61	2.26	0.65	0.69
10	6.85	1.88	0.69	0.73
Average	14.23	5.55	1.14	1.32

do not cancel out at these locations. To address this problem we propose using an adaptive filtering scheme, as described in the following section.

4.5 Luminance Filtering at Red/Blue Samples

According to Lemma 1, we can interpolate the estimated, subsampled version of luminance (at green-sample locations, see Section 4.4) to full resolution by

Figure 4.7: MSE of the estimated green pixel luminance *vs.* (a) r^2 (b) filter size.

using the difference images $Cr = L - R$ or $Cb = L - B$ [Figure 4.9(a)]. However, this is one type of a chicken-and-egg problem, as the references must have full resolution, which requires extension of subsampled red/blue components to full resolution. But, the red/blue components also require a full-resolution reference, such as the full-resolution luminance.

We propose resolving this problem iteratively. We start with bilinear interpolation of red and blue components, separately, to full resolution, and then use these estimated components to interpolate a full-resolution version of the luminance (using Lemma 1). Hence, we improve the first estimates of full-resolution red and blue components by using the estimated luminance as a reference. This process can be repeated several times, until no further changes occur or the changes are sufficiently small. Our experiments show that two iterations are sufficient in most cases.

Since the first step, bilinear interpolation of red/blue components may fail to preserve high-frequency information, $\mathcal{F}_I \approx \mathcal{F}^l$ for $\mathcal{F} \in \{R, B\}$, the high-

Figure 4.8: Proposed low-pass filter for green pixel luminance: (a) passband, (b) spectrum representation, and (c) impulse response.

frequency information will also be lost in the second step, bilinear interpolation of the luminance component. According to Lemma 1:

$$\hat{L} = \mathcal{I}_{bil}(L_s - \mathcal{F}_s^l) + \mathcal{F}^l \approx L^l \quad (4.11)$$

where $\mathcal{I}_{bil}$ designates bilinear interpolation. To better preserve the high-frequency information we propose using an anisotropic adaptive interpolation for the color-difference components.

Figure 4.9: (a) Luminance instead of green in CFA, (b) reference neighboring samples, and (c) the adaptive interpolation filter.

Figure 4.10: (a) color-difference image $Cr = L - R_T$ and its spectrum of regions with (b) diagonal edge, (c) horizontal edges, and (d) vertical edges.

We define the adaptive filter coefficients $w_k, k = 1 - 4$, shown in Figure 4.9(c), to be proportional to the spatial correlation along the corresponding directions, which can adapt to different neighborhood spatial patterns. For example, the stripes in Figure 4.10(c) are much smoother (higher spatial correlation) along the horizontal direction than along the vertical direction, which contains substantial high-frequency information. The corresponding adaptive filter, composed of high weights w_2, w_4 and low w_1, w_3 , has the passband as shown in Figure 4.11(b), which preserves high frequencies along the vertical direction. Similar effect can be achieved for edges along vertical or diagonal directions. For example, for the diagonal edge in Figure 4.10(b) the corresponding adaptive filter is shown in Figure 4.11(c).

In our adaptive filtering, the correlations (and hence the coefficients w_i) are estimated from the neighborhood pixel values. For red-sample locations shown

Figure 4.11: (a) Bilinear interpolation filter and (b,c) two examples of adaptive filter for horizontal and diagonal edges, respectively. The side or corner passband is due to the replication of spectrum.

in Figure 4.9(b), the coefficients are defined as

$$\begin{aligned} \frac{1}{w_1} &= 1 + |R(m, n) - R(m, n + 2)| + |L(m, n + 1) - L(m, n - 1)| \\ \frac{1}{w_2} &= 1 + |R(m, n) - R(m - 2, n)| + |L(m + 1, n) - L(m - 1, n)| \\ \frac{1}{w_3} &= 1 + |R(m, n) - R(m, n - 2)| + |L(m, n + 1) - L(m, n - 1)| \\ \frac{1}{w_4} &= 1 + |R(m, n) - R(m + 2, n)| + |L(m + 1, n) - L(m - 1, n)| \end{aligned} \quad (4.12)$$

Note that these expressions define a type of edge detector; the presence of edge makes the corresponding coefficients small. However, unlike traditional edge

Figure 4.12: Differences between original Cr component and the estimated ones by (a) bilinear interpolation, and (b) proposed adaptive filter. The dark pixels are with intensity levels larger than 8.

detectors, ours operates directly on the subsampled data. Moreover, it can also work on diagonal edges, as shown in Figure 4.11(c).

Similar weights can be used for interpolation at blue-sample locations. The luminance prediction results, shown in Figure 4.12, demonstrate that the proposed adaptive filter yields more accurate luminance component compared with bilinear interpolation.

The filtering performance is improved substantially from first to second iteration, but further improvements are minor; see Table 4.2. Overall, the proposed approach has a notably better performance (in terms of MSE) compared with the Alleysson's filter. The first column in the table also shows that the performance of previously used 5×5 filter (developed for green samples) is worse than Alleysson's filter when applied at red or blue-sample locations, as expected.

Table 4.2: MSE performance of the estimated red/blue pixel luminance by different filters

Image No.	Green samples 5×5 filter	11×11 filter Alleysson's	Adaptive filter with iteration		
			Once	Second	Third
1	30.37	13.97	24.74	9.34	7.41
2	9.30	11.55	5.99	3.97	4.03
3	5.89	5.87	4.39	2.19	2.47
4	6.29	5.17	6.42	2.90	3.18
5	23.97	15.66	20.52	6.94	7.53
6	24.52	10.43	19.72	8.45	7.11
7	6.69	6.11	4.38	2.10	2.53
8	62.05	30.21	35.28	14.95	12.79
9	7.82	6.34	5.20	2.36	2.44
10	7.14	4.82	5.13	2.28	2.33
Average	18.41	11.02	13.18	5.55	5.18

4.6 Summary of Proposed Method

The proposed adaptive filtering demosaicking can be summarized with the following three steps:

1. Apply the proposed filter shown in Figure 4.8 to CFA image to obtain the luminance values at green-sample locations.
2. Apply the adaptive filter with coefficients defined in (4.12) to estimate the luminance values at the red/blue-sample locations.
3. Use the estimated full-resolution luminance as a reference to interpolate the sub-sampled color components (red, green, and blue) of the CFA image to reconstruct the full color image based on Lemma 1.

Note that we use adaptive interpolation in Step 2 but bilinear interpolation in Step 3. Bilinear interpolation is sufficient in Step 3, because the well estimated luminance preserves the high-frequency information and leads to sufficiently smooth color-difference images. Applying the proposed adaptive filter in Step

Table 4.3: PSNR performance (in dB) comparison of bilinear and adaptive filter in Step 3.

Image		Bilinear	Adaptive	Image		Bilinear	Adaptive
1	R	36.68	37.09	6	R	37.70	37.94
	G	39.92	39.80		G	40.74	40.67
	B	36.90	37.20		B	36.85	37.13
2	R	37.71	37.77	7	R	42.27	42.54
	G	43.71	43.80		G	45.61	45.55
	B	40.10	40.85		B	40.93	41.22
3	R	42.19	42.40	8	R	34.54	34.66
	G	45.63	45.69		G	38.21	38.13
	B	41.41	41.96		B	34.53	34.65
4	R	37.16	37.81	9	R	41.37	41.87
	G	44.67	44.58		G	45.17	45.15
	B	42.63	42.90		B	43.07	42.75
5	R	37.51	37.75	10	R	41.17	41.53
	G	40.84	40.74		G	45.54	45.51
	B	36.86	37.11		B	42.06	41.99
				Average	R	38.83	39.14
					G	43.00	42.96
					B	39.53	39.78

3 will improve the performance by only 0.31 dB on average, as shown in Table 4.3, which might not justify the 70% increase in the computational complexity.

4.7 Experimental Results

To evaluate the proposed method we compare it with five state-of-the-art demosaicking techniques—Alternating Projections (AP) [2], Successive Approximation (SA) [83], Primary-Consistent Soft-Decision (PCSD) [49]¹, Frequency Selection (FS) [7], and Adaptive Homogeneity-Directed (AHD) demosaicking [50]—as well as bilinear interpolation. Among these methods, bilinear interpolation is the simplest and also most common reference for performance comparison in the demosaicking literature. The other five methods have only recently been pro-

¹The PCSD algorithm we used here includes the directional filtering that further improves the PSNR performance by an average of over 1.5 dB [49].

Table 4.4: PSNR performance (in dB) comparison (images 1-15). For each image the red, green and blue components are listed in the 1st, 2nd and 3rd rows. The best PSNR of each row is shown in bold and underlined.

Image	Bilinear	AP	PCSD	SA	FS	AHD	Ours
1	25.31	36.59	36.35	36.66	35.06	34.62	<u>36.68</u>
	29.58	40.49	38.85	<u>40.88</u>	39.64	36.28	39.92
	25.35	36.86	36.55	<u>37.64</u>	35.45	34.79	36.90
2	31.89	35.87	37.32	35.54	34.04	36.59	<u>37.71</u>
	36.26	40.81	42.59	40.06	39.70	41.56	<u>43.71</u>
	32.36	38.60	<u>40.78</u>	39.07	36.57	40.19	40.10
3	33.45	40.43	41.46	39.08	38.68	40.64	<u>42.19</u>
	37.17	43.48	44.67	42.21	42.20	43.74	<u>45.63</u>
	33.83	40.39	40.88	39.36	38.49	40.16	<u>41.41</u>
4	32.61	36.68	36.31	36.17	35.21	35.74	<u>37.16</u>
	36.53	43.33	43.31	42.64	42.99	41.74	<u>44.67</u>
	32.91	41.42	42.18	40.88	40.76	40.98	<u>42.63</u>
5	25.75	36.82	36.73	35.59	34.97	35.14	<u>37.51</u>
	29.32	39.81	39.80	38.27	38.47	37.46	<u>40.84</u>
	25.92	36.55	36.14	35.34	34.80	34.44	<u>36.86</u>
6	26.70	37.72	<u>39.32</u>	37.22	36.25	37.60	37.70
	31.05	<u>41.52</u>	41.49	41.40	40.81	39.13	40.74
	27.00	37.27	<u>37.99</u>	36.67	35.67	36.51	36.85
7	32.57	40.84	41.02	40.12	38.89	40.12	<u>42.27</u>
	36.47	43.62	43.76	42.67	42.44	42.71	<u>45.61</u>
	32.58	39.91	39.82	39.35	38.08	39.08	<u>40.93</u>
8	22.53	34.06	34.34	34.44	31.47	33.23	<u>34.54</u>
	27.4	38.59	37.48	<u>38.74</u>	36.63	35.36	38.21
	22.48	34.29	34.45	<u>35.00</u>	31.64	33.26	34.53
9	31.56	40.54	41.05	40.60	38.68	40.36	<u>41.37</u>
	35.72	43.26	44.21	43.33	42.34	43.06	<u>45.17</u>
	31.38	41.43	42.52	41.36	39.74	40.70	<u>43.07</u>
10	31.84	40.55	40.46	40.12	39.28	39.53	<u>41.17</u>
	35.37	43.89	44.46	43.86	43.55	42.92	<u>45.54</u>
	31.23	40.68	41.50	40.56	40.14	39.85	<u>42.06</u>
11	28.17	37.86	38.02	37.60	36.45	36.66	<u>38.28</u>
	32.22	41.64	40.96	41.70	41.05	39.03	<u>41.81</u>
	28.34	<u>39.06</u>	38.85	38.69	37.51	37.48	38.98
12	32.67	41.12	41.64	41.07	39.05	40.63	<u>41.75</u>
	36.82	44.90	45.41	44.71	43.62	44.01	<u>45.77</u>
	32.35	41.40	<u>42.10</u>	41.39	39.51	40.99	41.86
13	23.09	34.05	33.43	<u>34.70</u>	34.57	31.54	33.68
	26.49	36.91	34.95	<u>37.90</u>	37.72	32.41	35.98
	23.00	33.16	32.43	<u>33.35</u>	33.30	30.62	32.74
14	28.15	34.79	34.40	33.33	32.44	34.17	<u>36.10</u>
	32.01	38.37	38.32	36.74	36.88	38.02	<u>40.52</u>
	28.60	35.62	35.29	34.39	33.17	34.95	<u>36.22</u>
15	31.05	36.37	36.28	35.34	34.85	35.52	<u>37.19</u>
	35.36	41.49	41.84	40.58	40.55	40.69	<u>43.13</u>
	32.26	39.31	39.67	38.83	37.45	38.61	<u>40.24</u>

Table 4.5: PSNR performance (in dB) comparison (images 16-30). For each image the red, green and blue components are listed in the 1st, 2nd and 3rd rows. The best PSNR of each row is shown in bold and underlined.

Image	Bilinear	AP	PCSD	SA	FS	AHD	Ours
16	30.30	40.80	<u>43.05</u>	38.66	39.16	41.43	40.72
	34.73	44.88	<u>45.14</u>	43.40	43.99	42.91	43.83
	30.39	40.62	<u>42.06</u>	39.40	38.85	40.61	40.22
17	31.63	40.78	40.64	40.71	40.06	39.23	<u>40.96</u>
	34.54	42.92	42.56	43.13	43.06	40.68	<u>43.45</u>
	30.86	40.22	40.41	39.93	40.03	38.65	<u>40.69</u>
18	27.32	35.52	35.13	35.51	34.90	33.58	<u>35.57</u>
	30.54	38.64	37.64	38.61	38.37	35.56	<u>38.79</u>
	26.82	36.36	35.75	<u>36.43</u>	35.55	33.56	36.35
19	26.78	37.90	<u>38.72</u>	38.55	35.75	37.44	38.68
	31.74	41.85	41.41	<u>42.33</u>	40.46	39.39	42.30
	26.95	39.23	39.79	39.63	36.73	37.62	<u>39.99</u>
20	30.81	39.25	<u>40.21</u>	38.75	38.59	38.98	40.14
	34.58	42.72	42.37	41.74	41.97	40.71	<u>43.59</u>
	30.59	38.01	37.80	37.22	36.60	36.75	<u>38.39</u>
21	27.63	38.26	37.86	37.83	36.93	36.52	<u>38.34</u>
	31.54	<u>41.63</u>	40.00	41.61	40.98	37.89	41.18
	27.53	<u>37.55</u>	36.85	36.98	36.22	35.48	37.38
22	29.84	37.09	36.78	36.93	36.40	35.69	<u>37.81</u>
	33.34	39.58	39.52	39.63	39.28	38.47	<u>41.20</u>
	29.22	36.81	36.37	36.87	35.51	35.36	<u>37.42</u>
23	34.43	37.61	39.36	38.06	36.97	<u>40.05</u>	38.94
	37.98	41.64	44.11	41.83	41.27	44.15	<u>44.93</u>
	34.12	40.42	39.79	40.54	37.65	39.18	<u>40.75</u>
24	26.40	34.87	35.02	34.79	<u>35.29</u>	32.58	35.04
	29.38	37.33	37.12	37.59	<u>37.83</u>	34.96	37.51
	25.29	32.70	<u>33.00</u>	32.80	32.74	31.53	32.83
25	25.9	32.25	<u>33.34</u>	32.14	31.78	33.28	32.50
	30.28	36.71	<u>38.25</u>	36.48	36.60	38.05	37.44
	26.01	32.54	<u>33.48</u>	32.43	31.81	33.06	32.69
26	29.65	35.53	34.37	35.28	34.62	33.40	<u>35.69</u>
	32.72	39.08	37.27	<u>39.46</u>	39.06	35.68	39.09
	29.58	36.45	35.03	<u>36.51</u>	35.86	34.33	36.34
27	22.47	30.96	<u>31.93</u>	30.99	29.76	30.81	31.75
	26.10	33.36	34.40	33.32	33.03	32.92	<u>34.48</u>
	22.32	30.15	<u>30.90</u>	30.20	29.23	29.76	30.83
28	25.70	35.60	36.15	36.00	34.38	35.41	<u>36.19</u>
	29.76	38.70	38.83	39.11	38.22	37.73	<u>39.50</u>
	25.94	35.15	35.48	35.33	33.97	34.77	<u>35.84</u>
29	32.75	39.52	39.62	39.89	37.84	39.63	<u>40.38</u>
	36.34	42.80	43.41	43.14	41.64	43.87	<u>44.52</u>
	33.49	40.74	40.68	40.71	39.04	40.31	<u>41.63</u>
30	27.01	28.38	28.38	28.01	26.75	28.01	<u>29.21</u>
	29.46	30.72	31.01	30.23	30.03	31.42	<u>32.59</u>
	24.09	26.50	26.47	26.19	25.17	26.46	<u>27.13</u>
Average	28.87	36.95	37.29	36.66	35.64	36.27	<u>37.61</u>
	32.69	40.49	40.51	40.24	39.81	39.08	<u>41.37</u>
	28.76	37.31	37.50	37.10	35.91	36.33	<u>37.81</u>

posed and can obtain superior results. The test images used in the experiments are shown in Figure 2.16.

PSNR performance comparison results are summarized in Tables 4.4 and 4.5. Our method outperforms the other methods for the overwhelming majority of test images. On average, the improvement to bilinear is quite large, up to 8.8 dB. The improvement to the AP, PCSD and SA methods is around 0.5 dB - 1.0 dB, and that to the FS and AHD methods is around 1.5 dB - 2.3 dB.

Figure 4.13: Original and demosaicked results of a cropped region in Image 3.

Figure 4.14: Original and demosaicked results of a cropped region in Image 9.

Figure 4.15: Original and demosaicked results of a cropped region in Image 27.

Figures 4.13 – 4.16 show cropped regions of four test images and the corresponding demosaicked images. It can be observed that the AHD and our methods obtain the best perceptual results with sharper edges and fewer zipper artifacts. Bilinear interpolation yields blurred edges and many color artifacts. The AP, SA, and FS methods produce severe zipper artifacts around the edges due to the use of isotropic interpolation. The PCSD method can obtain better results since it uses an anisotropic interpolation technique, but the training criterion it uses for edge direction estimation may be inaccurate for some image regions, leading to some remaining artifacts.

As reported in Gunturk’s review paper [82], the AHD method can produce perceptually favorable images with fewer artifacts, which is also confirmed by the results in Figures 4.13 - 4.16. The approach uses a type of anisotropic filtering, based on some homogeneity criteria for neighborhood similarity. However, the interpolation is restricted to one of two directions (horizontal or vertical), which may not work well for textured images, where high-frequency components might

Figure 4.16: Original and demosaicked results of a cropped region in Image 27.

need to be preserved in both directions (see Figure 4.17). On the other hand, the same figure shows that our method can better preserve the textured structures due to the adopted adaptive filtering. Moreover, compared to the AHD method, our method yields better PSNR performance (see Tables 4.4 and 4.5), and has much lower computational cost, as discussed in the following section.

4.8 Discussion

To further show the superiority of the proposed algorithm, in this section we discuss the issues of computational complexity and an interesting phenomenon on modified CFA patterns observed earlier by Alleysson *et al.* [7].

4.8.1 Computational complexity

Let $M \times N$ be the size of the CFA image to be demosaicked. For the proposed method, the low-pass filtering at green pixels requires a total of $10MN$

Figure 4.17: Comparison of AHD and proposed demosaicking methods in texture regions

additions, $2.5MN$ multiplications, and $8MN$ bit-shift operations. Adaptive filtering at red and blue pixels needs $22.5MN$ additions, $8MN$ multiplications, $3MN$ absolute conversions, and $1MN$ bit-shift operations. Note that the coefficients of the adaptive filter are computed only once for the two iterations. The subsequent bilinear interpolation in the color-difference images requires $8MN$ additions and $2MN$ bit shifts. The computational complexity of the proposed method is summarized in Table 4.6.

Table 4.6 also shows the computational complexities of the five state-of-the-

Table 4.6: Comparison of demosaicking complexity.

Method	No. of computational operations			
	Addition	Absolute	Bit-shift	Multiplication
AP	$384MN$	-	-	$384MN$
SA	$32-48MN$	$2MN$	$3.5MN$	$12.75-21.25MN$
PCSD	$149MN$	$60MN$	$5MN$	$8MN$
FS	$76MN$	-	$48MN$	$21MN$
AHD ²	$184MN$	$12MN$	-	$49MN$
Proposed	$40.5MN$	$3MN$	$11MN$	$10.5MN$

art methods. The complexity figures of the AP [2] and AHD [50] methods are taken directly from the corresponding papers. Although Li [83] also analyzed the complexity of his proposed SA method, his analysis did not include the computational cost for region separation (which requires $8MN$ additions and $2MN$ bit-shifts) and the stop criterion (which requires about $18-30MN$ additions and $9-15MN$ multiplications for 3-5 iterations). For the PCSD and FS methods we performed our own complexity analysis based on the algorithms described in the respective papers [7, 49].

It also follows from Table 4.6 that our proposed method is the least computationally complex. Although the PCSD method has fewer multiplications, it requires much more additions and absolute operations; moreover, the computation reported here does not include the training process. The SA method requires slightly fewer additions, but more multiplications. The AHD method, which is most comparable to the proposed method in the visual quality sense, requires nearly five times the number of additions and multiplications.

²Here we consider the comparison operation as addition one, and count two iterations (as suggested by authors) for median filtering.

4.8.2 Modified CFA patterns

Based on the frequency selection (FS) demosaicking method as described in Section 4.3, Alleysson *et al.* have specified that the current Bayer CFA pattern is not optimal and could be improved by exchanging the color components (see the modified patterns shown in Figure 4.18). The improved PSNR performance is 0.5 dB on average, as shown in the left columns of Table 4.7, which is the same as Table I in [7]. The explanation given in [7] is that CFA side spectrum component $C1$ has a narrower spectrum in the modified patterns, where $C1 = \frac{1}{8}(R - G)$ or $C1 = \frac{1}{8}(B - G)$. This can be explained by typically higher correlation between the red/blue and green components, resulting in better luminance estimation and demosaicking performance.

Figure 4.18: Modified Bayer CFA patterns with (a) red and (b) blue in quincunx lattice.

However, the visual quality results on the modified patterns do not show much improvement, and sometimes are even worse than those on the original pattern [87]. Actually the higher red-green and blue-green correlations suggest that green samples share more information with red/blue samples, and hence are given more samples in the original Bayer CFA. We have found that the demonstrated PSNR improvement on modified patterns is mainly due to the filter used by FS demosaicking. The filter removes the side components of $C1$

Table 4.7: Comparison of PSNR performance (in dB) by applying FS approach [7] and the proposed method on original Bayer CFA pattern and two modified ones. The best PSNR in the comparison is shown in bold and underlined.

Image	FS [7]			Proposed method		
	Bayer CFA	$R \Leftrightarrow G$	$B \Leftrightarrow G$	Bayer CFA	$R \Leftrightarrow G$	$B \Leftrightarrow G$
Lighthouse	33.88	<u>34.43</u>	34.34	40.10	<u>40.26</u>	39.72
Sails	35.72	36.28	<u>36.59</u>	42.92	<u>43.01</u>	42.40
Statue	37.70	<u>37.92</u>	37.86	<u>41.54</u>	41.30	41.36
Window	35.27	35.47	<u>35.87</u>	42.67	42.53	<u>42.81</u>
Average	35.64	36.03	<u>36.17</u>	<u>41.81</u>	41.77	41.57

(see Figure 4.5), but also discards some high frequencies along the horizontal and vertical axes, which are preserved in the Bayer CFA, as we discussed in Section 4.3.

For our method, the proposed filter does not remove $C1$ and then preserves the important high frequencies. Hence the spectrum bandwidth of $C1$ does not affect the luminance estimation and the PSNR performance. The right columns of Table 4.7 show that by using the proposed method the PSNR values on three patterns are quite close to each other and the original Bayer pattern yields best PSNR results on average.

4.9 Conclusions

In this chapter, we have discussed two important observations for preserving high-frequency information in CFA demosaicking. First, we have shown that due to the similarity of high-frequency information across three color components, one full-resolution component can be used to estimate the high-frequency information of all other components. We have also shown that high-frequency information along horizontal and vertical axes is more important for image qual-

ity than that in the spectrum's corners. Our frequency analysis explains why the Bayer CFA pattern can preserve the vital high-frequency information.

Based on these observations, we propose an adaptive filtering demosaicking method for better preservation of high-frequency information. We designed an efficient low-pass filter on CFA image to estimate the luminance at green-sample locations. The proposed filter can preserve more high frequencies than the existing filters. Next, an adaptive filtering is used to estimate the luminance at red and blue samples. Our frequency analysis indicates that the proposed adaptive filter is well suited for varying image content and can better preserve high frequencies. Experimental results confirm that the proposed method outperforms the existing state-of-the-art methods both visually and in terms of peak signal-to-noise ratio (PSNR), at a notably lower computational cost.

Part III

Denoising

Chapter 5

Wavelet and Min-Cut Analysis for Image Denoising

5.1 Introduction

As discussed in Chapter 1 (Section 1.3.2), wavelet based approach can achieve superior edge preservation in image denoising. The main idea of wavelet based denoising is to separate wavelet coefficients into two clusters, corresponding to edges and smooth image areas, and then apply different denoising rules to each cluster. The performance of this denoising approach depends on the accuracy of classification. Clustering can be done via simple thresholding of individual wavelet coefficients, but can be significantly improved by exploiting inter-scale correlations [62]. For example, the uHMT algorithm uses tree structure to constrain coefficients between neighboring scales [4].

Intra-scale dependencies are rarely used in denoising due to their anisotropic properties near image edges. However, since the high-frequency information across three color components is similar to each other (Section 1.3.3), the color-difference components, e.g., Cb and Cr in YCbCr space¹ are generally smooth and hence have stronger and more isotropic intra-scale correlations that can be easily exploited. We propose in this chapter an algorithm that achieves superior denoising of Cb and Cr components by incorporating intra-scale correlations.

For the main implementation we adopt a minimum cut (min-cut) algorithm [88]. This is a popular graph partitioning technique that has been applied to many problems in stereo matching, image segmentation, and binary image denoising [89], and but not yet to wavelet denoising. While the uHMT algorithm exploits only inter-scale correlations, the min-cut algorithm can potentially exploit both intra-scale and inter-color correlations. Moreover, compared with some Markov random field (MRF) based algorithms [56], the min-cut algorithm converges to an optimal solution in polynomial time.

5.2 Min-Cut in Wavelet Denoising

Let x_i be the wavelet coefficient, $\tilde{x}_i$ its noisy version, and n_i the noise component; that is

$$\tilde{x}_i = x_i + n_i. \quad (5.1)$$

Let s_i be the state of x_i , where $s_i = \mathcal{S}$ corresponds to small coefficients and $s_i = \mathcal{L}$ to large coefficients. We model states $s = (s_1, s_2, \dots, s_N)$ of the wavelet coefficients $\tilde{x} = (\tilde{x}_1, \tilde{x}_2, \dots, \tilde{x}_N)$, as a Markov random field (MRF), and obtain

¹Denoising in YCbCr space achieves better performance than that in RGB space, as shown in Table 6.7 of Chapter 6.

the maximum a posteriori (MAP) estimates of the states by maximizing $p(s|\tilde{x})$. Alternatively, we can minimize its negative logarithm version, defined as the cost function

$$J(s) = -\log p(s|\tilde{x}) \propto -\sum_{i=1}^N \log f(\tilde{x}_i|s_i) - \log p(s),$$

where $f(\cdot|q)$ denotes the probability density function of the noisy coefficient $\tilde{x}_i$ when state $s_i = q$ and $q \in \{\mathcal{S}, \mathcal{L}\}$. Since s is a MRF, conditional probability of the state s_i given all other states is equal to the conditional probability of the state s_i given only its neighborhood δs_i :

$$p(s_i|s) = p(s_i|\delta s_i) \quad (5.3)$$

A clique c is a set of nodes that are all neighbors of each other. Let $\mathcal{C}$ be a set of all cliques. Then according to the Markov Random Theory [90], there exists a set of potential functions $V_c(s_c)$, such that,

$$-\log p(s) = \log Z + \sum_{c \in \mathcal{C}} V_c(s_c) \quad (5.4)$$

where Z is the normalizing constant. If the chosen neighborhood allows only single and pairwise cliques, then $\sum_c V_c(s_c) = \sum_i V_i(s_i) + \sum_{i,j} V_{i,j}(s_i, s_j)$. Substituting (5.4) into (5.2) we obtain

$$J(s) = -\sum_{i=1}^N \log f(\tilde{x}_i|s_i) + \log Z + \sum_i V_i(s_i) + \sum_{i,j} V_{i,j}(s_i, s_j) \quad (5.5)$$

Similar expressions can be obtained for larger neighborhoods. For the case of pairwise and triple cliques, a global minimum of (5.5) can be found in polynomial

time using a graph-cut approach [91].

A graph cut is a set of graph edges that, if removed, separate the vertices of the graph into two disjoint components. Minimum cut is a problem of finding a cut with a minimum cost (sum of weights of the removed edges) that separates the graph into two subsets of vertices. We define a directed graph where each vertex corresponds to a wavelet coefficient, and the edges in the graph connect vertices that correspond to neighboring coefficients. In addition we introduce two terminal vertices, referred to as the source **S** and the sink **T**, and connect them to all other vertices. The two terminal vertices represent the states, i.e. Small (**S**) and Large (**T**); see Figure 5.1. By proper assignment of edge weights, minimum cut solves the problem of minimizing (5.5).

A graph that exploits only inter-scale correlations is acyclic and hence will have only pairwise cliques, as shown Figure 5.1(a). Exploiting inter and intra-scale ones leads to a graph with triple cliques, as shown in Figure 5.1(b). Both of these cases can be solved using minimum cut [91]. In what follows, we show that how to define the edge weights for an acyclic graph based on the local statistics of wavelet coefficients.

Consider a MRF $\mathcal{G}$, where a node a has K neighbors $b_1, b_2, \dots, b_K$, which are not neighbors of each other (Figure 5.1(a)). Then

$$\log p(s_{b_1} | s_a, s_{b_2}, \dots, s_{b_K}) = \log p(s_{b_1} | s_a). \quad (5.6)$$

From (5.6) we can obtain

$$-\log p(s_a, s_{b_1}, s_{b_2}, \dots, s_{b_K}) = -\sum_{i \in \delta a} \log p(s_i, s_a) + (K-1) \log p(s_a), \quad (5.7)$$

where δa is the neighbor set of a . Let $s = (s_a, s_{b_1}, s_{b_2}, \dots, s_{b_K})$. Substituting s into (5.2) and combining it with (5.7) yields

$$J(s) = - \sum_{i \in \mathcal{G}}^N \log f(\tilde{x}_i | s_i) - \sum_{i \in \delta a} \log p(s_i, s_a) + (K - 1) \log p(s_a). \quad (5.8)$$

The functional $J(s)$ can be efficiently minimized by finding a minimum cut on the graph, where the edge weights are given in the following Lemma.

Lemma 2. *For an acyclic graph as shown in Figure 5.1(a), let*

$$w_i(q) = -\log f(\tilde{x}_i | q) - \sum_{j \in \delta i} \log \sqrt{p_{i,j}(q, q)} + (k_i - 1) \log p_i(q) \quad (5.9)$$

be the weights connecting vertex i to the source ($q = \mathcal{S}$) and the sink ($q = \mathcal{L}$), where k_i is the number of the vertices connected to vertex i . Let

$$w_{i,j} = -\log \frac{p(s_i = \mathcal{S}, s_j = \mathcal{L})}{\sqrt{p_{i,j}(\mathcal{L}, \mathcal{L})p_{i,j}(\mathcal{S}, \mathcal{S})}} \quad (5.10)$$

be the weight of a directed edge that starts at vertex i and ends at vertex j . Then the minimum cut on so defined graph minimizes $J(s)$ in (5.8).

Proof: Any cut that separates the source and the sink assigns the remaining vertices to two disjoint sets, w_0 and w_1 . Vertices in w_0 remain connected to the source, while vertices in w_1 remain connected to the sink. The weights of the edges removed between the vertices in w_0 and the sink, and between the vertices in w_1 and the source, are given in (5.9). In addition, edges connecting vertices in w_0 with vertices in w_1 are removed. Their weights are given in (5.10). The sum of the weights of all the removed edges is equal to $J(s)$ if we assign $s_i = \mathcal{L}$ when $i \in w_0$, and $s_i = \mathcal{S}$ when $i \in w_1$. ■

Figure 5.1: Graphs exploit (a) inter-scale correlation only or (b) both inter and intra-scale correlations.

Corollary 1. *The same weight assignment is true for any acyclic graph.*

The acyclic graph analysis cannot be straightforwardly extended to the graphs with triple cliques. Solving the problem in the latter case will require modelling of the potential functions. Alternatively, one can simply use the weight assignment (5.9), (5.10) in a triple clique graph to obtain a suboptimal solution.

For the probability density function of wavelet coefficients we adopt a Gaussian mixture model [4], where

$$f(x|q) = \frac{1}{\sqrt{2\pi\sigma_q^2}} \exp -\frac{x^2}{2\sigma_q^2}. \quad (5.11)$$

Furthermore, the joint probabilities $p(s_a, s_b)$ are derived using the universal model of the uHMT algorithm [4].

5.3 Incorporating Intra-Scale Correlations

A straightforward denoising approach would be to simply plug in the state estimates produced by the min-cut into the MMSE estimate (1.7). However, we propose to also exploit the intra-scale correlations and use the state estimates to compute the local prior state probabilities:

$$\begin{aligned} p_i(\mathcal{L}) &= \frac{\sum_{j \in B(i)} 1_{s_j = \mathcal{L}}}{N_{B(i)}} \\ p_i(\mathcal{S}) &= 1 - p_i(\mathcal{L}) \end{aligned} \quad (5.12)$$

where $B(i)$ is the set of the neighbors of vertex i in the same scale, and $N_{B(i)}$ is the size of $B(i)$. The posterior probability given the noisy coefficient $\tilde{x}_i$ is thus

$$p(q|\tilde{x}_i) = \frac{p(q)f(\tilde{x}_i|q)}{p(\tilde{x}_i)} = \frac{p(q)f(\tilde{x}_i|q)}{\sum_{s \in \{\mathcal{S}, \mathcal{L}\}} p(s)f(\tilde{x}_i|s)} \quad (5.13)$$

The newly calculated posterior probabilities can be used for denoising in one of the following two ways. We can recompute the states by comparing $p(\mathcal{S}|\tilde{x}_i)$ and $p(\mathcal{L}|\tilde{x}_i)$ and then plug the new states into equation (1.6). Instead, we obtain better performance by linearly combining the denoised results with the use of the posterior probabilities as weights, the conditional MMSE estimate (1.6).

We refer to this novel denoising method, which applies wavelets and min-cut analysis with intra-scale processing in YCbCr space, as the WMCA method. Note that the WMCA method doesn't require the universal model for its min-cut stage. For example, we can adopt the signal-noise model by Donoho [53]. Model change will simply lead to different probability density function $f(\tilde{x}|q)$ and the denoising method (1.6).

5.4 Experimental Results

We apply the proposed WMCA method to 24 test color images in Figure 5.2 and compare its performance with that of the uHMT algorithm and a locally adaptive, variance-based filtering method [24] which is implemented as the MATLAB function *wiener2*. The popular Daubechies wavelet (Daubechies-4²), used in uHMT [4], is adopted in WMCA. The noise is added in RGB space, but the denoising is performed in RGB and YCbCr spaces, respectively. We report the results for two combinations of standard deviations of noise in RGB color components: $\{0.1, 0.1, 0.1\}$ and $\{0.15, 0.10, 0.05\}$. In YCbCr space, they correspond to standard deviations $\{0.067, 0.062, 0.066\}$ and $\{0.074, 0.049, 0.086\}$, respectively.

Figure 5.2: Test images (Image 1 to Image 24, enumerated from left-to-right and top-to-bottom).

The denoising results for each color component are shown in Tables 5.1 and

²Its mother wavelet filter is $(0.4830, 0.8365, 0.2241, -0.1294)$.

5.2. In terms of the average peak signal-to-noise ratio (PSNR), the proposed WMCA performs much better than the *wiener2* method. Compared with the uHMT method, the WMCA method performs marginally better on the Y component, and up to 5-dB better on the Cb and Cr components. We attribute this notable improvement to the WMCA being able to exploit the intra-scale correlations, as the Cb and Cr color-difference components are much smoother than the Y component, leading to stronger intra-scale correlations. On the other hand, inter-scale correlations are sufficient to denoise the Y component.

The performance gain can be confirmed by visual comparison. For example, Figure 5.3 and Figure 5.4 show that the images denoised by using the proposed WMCA method have fewer color artifacts than that obtained by the uHMT and *wiener2* methods.

5.5 Conclusions

We presented a novel method that applies min-cut, a popular graph partitioning technique, to wavelet-based image denoising. The min-cut exploits the inter-scale correlations and classifies wavelet coefficients into two categories. Then intra-scale correlations are exploited by updating local state probabilities using the states of the neighboring coefficients.

Superiority of the proposed method is confirmed by applying it to denoising of color images. Experimental results suggest substantial PSNR improvement in the denoised Cb and Cr color components. This improvement translates into fewer color artifacts in the resultant images.

Table 5.1: Performance comparison of the *wiener2*, uHMT, and WMCA methods in YCbCr space for $\sigma_n = \{0.1, 0.1, 0.1\}$. The PSNR results (in dB) of the Y, Cb and Cr components are listed in the 1st, 2nd and 3rd rows, respectively, of each image.

Image	<i>wiener2</i>	uHMT	WMCA	Image	<i>wiener2</i>	uHMT	WMCA	
1	26.34	26.69	26.97	13	25.73	25.40	25.91	
	31.63	37.46	41.11		31.64	36.50	37.90	
	31.08	36.68	38.33		31.13	37.24	40.65	
2	29.89	30.96	30.85	14	28.79	28.54	28.59	
	32.08	36.90	38.87		31.78	35.82	36.40	
	31.09	35.56	36.39		31.21	35.37	35.80	
3	29.90	30.96	30.78	15	29.63	30.21	30.22	
	31.68	36.43	37.78		32.05	37.32	39.69	
	31.08	36.96	39.43		31.27	35.79	37.00	
4	29.90	30.73	30.56	16	29.47	30.57	30.57	
	31.71	37.69	41.77		31.64	37.74	41.35	
	31.11	36.14	37.29		31.11	37.70	43.39	
5	27.13	26.80	27.04	17	29.88	30.12	30.11	
	31.76	35.85	36.41		31.78	36.92	39.65	
	31.23	36.06	36.86		31.33	37.10	40.96	
6	26.94	27.57	27.73	18	28.69	28.07	28.14	
	31.37	35.99	37.63		31.80	36.40	37.46	
	31.20	36.65	39.14		31.25	36.68	38.04	
7	29.88	29.94	29.82	19	27.42	27.91	28.44	
	31.60	36.67	37.94		31.62	37.02	39.16	
	31.09	36.78	38.44		31.14	36.93	39.51	
8	25.49	26.48	27.09	20	28.38	28.13	28.22	
	31.59	36.52	37.83		31.74	35.27	36.44	
	31.20	35.79	36.79		31.58	35.11	36.41	
9	29.57	30.19	30.17	21	28.16	28.36	28.33	
	31.53	37.04	39.14		31.53	36.95	38.91	
	31.13	37.34	39.93		31.09	36.97	39.40	
10	30.52	32.11	32.17	22	29.10	29.07	29.13	
	31.51	37.55	40.68		31.54	36.62	37.92	
	31.07	37.51	41.99		30.98	35.93	37.14	
11	28.59	28.85	28.87	23	30.73	32.44	32.63	
	31.70	37.13	39.59		31.56	36.38	37.84	
	31.15	36.58	38.12		31.04	36.26	37.85	
12	30.03	31.11	31.26	24	27.77	27.36	27.61	
	31.57	37.24	40.05		31.38	35.37	36.15	
	31.11	36.79	38.92		31.03	36.13	37.97	
				Average	28.66	29.11	29.22	
					31.66	36.70	38.65	
					31.15	36.50	38.57	

Table 5.2: Performance comparison of the *wiener2*, uHMT, and WMCA methods in YCbCr space for $\sigma_n = \{0.15, 0.10, 0.05\}$. The PSNR results (in dB) of the Y, Cb and Cr components are listed in the 1st, 2nd and 3rd rows, respectively, of each image.

Image	<i>wiener2</i>	uHMT	WMCA	Image	<i>wiener2</i>	uHMT	WMCA
1	25.98	26.16	26.45	13	25.39	24.98	25.48
	33.67	38.39	41.73		33.56	37.12	38.27
	28.82	35.69	37.51		28.95	36.09	39.84
2	29.17	30.59	30.54	14	28.24	28.05	28.09
	33.82	37.99	40.06		33.77	36.74	37.25
	28.80	34.55	35.39		28.99	34.16	34.58
3	29.16	30.47	30.31	15	28.84	29.62	29.64
	33.73	37.81	39.25		33.61	37.22	38.89
	28.90	36.09	38.57		29.19	34.90	36.37
4	29.16	30.37	30.17	16	28.84	30.12	30.12
	33.84	38.85	42.78		33.74	38.55	41.98
	28.85	35.16	36.27		28.95	36.60	42.71
5	26.71	26.18	26.39	17	29.08	29.51	29.50
	33.73	36.78	37.25		33.75	37.64	39.84
	29.12	35.08	35.97		29.09	35.91	39.84
6	26.45	26.95	27.10	18	28.15	27.60	27.67
	33.09	36.15	37.31		33.76	37.25	38.20
	28.96	35.11	37.45		29.07	35.25	36.68
7	29.14	29.40	29.30	19	26.93	27.36	27.94
	33.74	37.83	39.18		33.68	37.84	39.78
	28.88	35.80	37.49		28.87	35.65	38.19
8	25.14	25.42	26.40	20	27.55	27.37	27.46
	33.60	37.16	38.26		32.56	34.46	35.15
	28.92	34.48	35.65		29.11	32.63	33.56
9	28.90	29.62	29.60	21	27.62	27.82	27.83
	33.72	38.06	40.06		33.55	37.67	39.28
	28.88	36.25	38.99		28.83	35.72	38.15
10	29.70	31.59	31.69	22	28.49	28.61	28.70
	33.67	38.35	41.30		33.58	37.29	38.52
	28.77	36.41	41.26		28.68	34.63	35.92
11	28.02	28.30	28.30	23	29.88	31.84	32.01
	33.79	38.07	40.38		33.70	37.54	38.98
	28.98	35.68	37.46		28.80	34.88	36.34
12	29.26	30.57	30.75	24	27.19	26.71	26.99
	33.54	37.72	40.01		33.39	36.16	36.70
	28.93	36.01	38.13		28.81	35.00	36.99
				Average	28.04	28.55	28.68
					33.61	37.44	39.18
					28.92	35.32	37.47

(a) Origin image

(b) Noisy image

(c) *wiener2* on RGB (d) *wiener2* on YCbCr

(e) uHMT on RGB

(f) uHMT on YCbCr

(g) WMCA

Figure 5.3: Denoised images obtained by the *wiener2*, uHMT and WMCA methods. The standard deviations of the noise are $\{0.1, 0.1, 0.1\}$, added to the R, G, and B components, respectively.

(a) Origin image

(b) Noisy image

(c) *wiener2* on RGB (d) *wiener2* on YCbCr

(e) uHMT on RGB

(f) uHMT on YCbCr

(g) WMCA

Figure 5.4: Denoised images obtained by the *wiener2*, uHMT and WMCA methods. The standard deviations of the noise are $\{0.15, 0.10, 0.05\}$ added to the R, G, and B components, respectively.

Chapter 6

Optimal Color Space Projection for Image Denoising

6.1 Introduction

We have shown in the thesis introduction (Section 1.3.3), that the high-frequency wavelet coefficients of RGB color components are similar to each other. As a result, when plotted in a 3D space, the wavelet coefficients form a compact cluster that extends along the direction of vector $\mathbf{1} = \{1, 1, 1\}^T$, as shown in Figure 1.7. Hence the color-difference vectors $\mathbf{d}_i$ of typical luminance/color-difference (YCbCr) space [5] are orthogonal to the signal vector $\mathbf{1} = \{1, 1, 1\}^T$, as shown in Figure 1.7. This explains why denoising in YCbCr space improves the performance: the Cb and Cr components become smooth and easy to denoise (see also Section 1.3.3). The improvements can be

substantial, up to 1.1 dB compared to denoising in RGB space, in Table 6.7.

Figure 1.7 also shows that typical luminance vector $\mathbf{v}$ may not be aligned with the signal vector $\mathbf{1}$. The reason is that the luminance Y is chosen to approximate the grayscale image intensity. This makes YCbCr space backward compatible with the older black-and-white analog television sets [5]. However, there is a large family of luminance/color-difference spaces, all having basis vectors $\mathbf{d}_1$ and $\mathbf{d}_2$ orthogonal to $\mathbf{1}$, but using different definitions of Y. For example, orthonormal double-opponent color-difference basis [92] has vector $\mathbf{v}$ aligned with $\mathbf{1}$. This, however, still doesn't guarantee optimality. The scatter plots of wavelet coefficients are clustered along $\mathbf{1}$ only in the noiseless case. A good basis for description of a noiseless image is not necessarily optimal for image denoising. In this chapter we propose a way to estimate an optimal projection space for image denoising from the image noise statistics.

Specifically,

- We derive an optimal luminance/color-difference space projection that minimizes the noise variance in the luminance image, leading to better edge preservation. Color-difference images comprise a larger portion of noise but are smooth and hence can be efficiently denoised without sacrificing the quality.
- We propose a method to estimate the image and noise statistics, which are necessary to derive the optimal color space projection, directly from the noisy image data. The estimation approach uses similarity of detail wavelet coefficients to derive an overconstrained system of linear equations. We then solve this system of equations for the least squares estimates of the signal variance and the covariance matrix of the noise.

The remainder of the chapter is organized as follows. Section 6.2 provides a necessary background on wavelet-based denoising. Section 6.3 proposes an optimal luminance/color-difference projection for image denoising. In Sections 6.4 and 6.5, we describe the noise estimation procedure and a modified denoising method for color-difference images. A summary of the proposed denoising method is given in Section 6.6. Experimental results are presented in Section 6.7 to demonstrate the superiority of the proposed method over existing solutions, both in PSNR and in visual quality. Section 6.8 concludes the chapter and discusses future work.

6.2 Estimate Error of Wavelet-Based Denoising

In order to choose an optimal projection space we need to understand the link between the signal/noise energy and denoising performance. We do this on the example of conditional MMSE estimation approach [see Equation (1.7)], used in uHMT [4] and [61]. We first show that in this estimation approach smaller noise always leads to smaller estimation error. This is ensured by the following Lemma.

Lemma 3. *For a fixed signal variance $\sigma_{x|q}^2$, the error of conditional MMSE estimate (1.7) is a non-decreasing function of error variance σ_n^2 .*

Proof: From (1.7) the estimation error variance is

$$\varepsilon^2 = E[(x - \hat{x})^2] = E\left[\left(\frac{\sigma_n^2}{\sigma_{x|q}^2 + \sigma_n^2}x - \frac{\sigma_{x|q}^2}{\sigma_{x|q}^2 + \sigma_n^2}n\right)^2\right] = \frac{\sigma_{x|q}^2 \sigma_n^2}{\sigma_{x|q}^2 + \sigma_n^2} = \frac{\sigma_{x|q}^2}{\frac{\sigma_{x|q}^2}{\sigma_n^2} + 1} \quad (6.1)$$

which is a non-decreasing function of σ_n^2 for a fixed $\sigma_{x|q}^2$, as shown in Figure 6.1.

■

Corollary 2. *The estimation error variance is always smaller than the signal variance and the noise variance; i.e.,*

$$\forall \sigma_n^2 \rightarrow \varepsilon^2 \leq \sigma_{x|q}^2, \quad \varepsilon^2 \leq \sigma_n^2$$

Corollary 2 explains why it is easier to denoise color differences (Cb and Cr), where suppression of high frequencies leads to small detail wavelet coefficients (small $\sigma_{x|q}^2$). Therefore, the estimation error variance is always small, regardless of how large the noise is, as depicted in Figure 6.1. This suggests that the denoising performance can be improved by transforming the image to a color space in which the signal energy is concentrated in one of the transformed color components.

The results of Lemma 3 and its corollary are likely to hold even for techniques that don't use conditional MMSE estimates. Without loss of generality, the experimental results in this chapter are based on uHMT denoising approach [4].

Figure 6.1: Estimation error variance as a function of noise variance.

6.3 The Optimal Color Space Projection

In Chapter 1 we showed that detail wavelet coefficients are strongly correlated across image color components, with the correlation coefficient close to 1. For the sake of simplicity in our following analysis, we assume perfect correlation, namely

$$E(x_i x_j) = \sqrt{E(x_i^2)E(x_j^2)} = \sqrt{s_i^2 s_j^2}, \quad \forall i, j \quad (6.2)$$

where x_i 's are the detail wavelet coefficients in the i -th color component of the noiseless image, and $s_i^2 \equiv E(x_i^2)$. This assumption may not be satisfied exactly in practice and can be relaxed (see Appendix B for the treatment of the general case of an arbitrary signal covariance matrix).

For a typical color image we have $s_1^2 \simeq s_2^2 \simeq s_3^2 = \bar{s}^2$ (see discussion in Chapter 1). However, the results derived in this section are applicable to arbitrary signal variances.

Let vector $\mathbf{s} = (s_1, s_2, s_3)^T$. We consider a family of luminance/color-

difference spaces with basis vectors $(\mathbf{v}, \mathbf{d}_1, \mathbf{d}_2)$, where $\mathbf{d}_1$ and $\mathbf{d}_2$ are orthogonal to $\mathbf{s}$. Our goal is to determine a basis that yields the best denoising performance. Let $x_v = \mathbf{v}^T \mathbf{x}$ be the luminance projection, and $x_{d_i} = \mathbf{d}_i^T \mathbf{x}, i = 1, 2$, be the color-difference projections. Varying the direction of $\mathbf{d}_i$ changes the noise components of x_{d_i} only; the signal component of x_{d_i} , and hence its variance $\sigma_{x|q}^2$, remain small. This means good denoising performance, as the estimation error variance ε^2 is always smaller than $\sigma_{x|q}^2$ (see Corollary 2). On the other hand, the luminance x_v has a large signal component. Hence, according to Lemma 3, its denoising performance can be improved by reducing the projected noise variance.

This observation is illustrated in Table 6.1, where we apply uHMT denoising to color components in YCbCr space. Denoising Cb and Cr leads to 6-7 dB better PSNR than denoising Y. Hence, the performance of luminance denoising, and not that of the color differences, is the key factor in the overall denoising performance.

Table 6.1: PSNR performance (in dB) of Y, Cb and Cr using the uHMT method in the case of uncorrelated noise. For uniform noise, the noise standard deviations added to {R, G, B} are equal to $\sigma_n = \{0.1, 0.1, 0.1\}$; for nonuniform noise, $\sigma_n = \{0.15, 0.10, 0.05\}$. These two types of uncorrelated noise remain the same for all other experiments in the chapter.

Color Image	Uniform noise			Nonuniform noise		
	Y	Cb	Cr	Y	Cb	Cr
1	26.69	37.46	36.68	26.16	38.39	35.69
2	30.96	36.90	35.56	30.59	37.99	34.55
3	30.96	36.43	36.96	30.47	37.81	36.09
4	30.73	37.69	36.14	30.37	38.85	35.16
5	26.80	35.85	36.06	26.18	36.78	35.08
6	27.57	35.99	36.65	26.95	36.15	35.11
7	29.94	36.67	36.78	29.40	37.83	35.80
8	26.48	36.52	35.79	25.42	37.16	34.48
9	30.19	37.04	37.34	29.62	38.06	36.25
10	32.11	37.55	37.51	31.59	38.35	36.41
Average	29.24	36.81	36.55	28.68	37.74	35.46

Our goal is to find a basis $(\mathbf{v}, \mathbf{d}_1, \mathbf{d}_2)$ with optimal luminance projection for image denoising purposes. Note that the basis vectors are not orthogonal in general because $\mathbf{d}_1$ and $\mathbf{d}_2$ can not be simultaneously orthogonal to $\mathbf{v}$ and $\mathbf{s}$. To simplify the analysis we normalize the coordinates of $\mathbf{v} = (v_1, v_2, v_3)^T$ so that the variance of the signal projection onto $\mathbf{v}$ is fixed. This is the topic of the following lemma.

Lemma 4. *Let $\mathbf{x}$ be a random vector that satisfies assumption (6.2). If we normalize $\mathbf{v}$ so that $\mathbf{v}^T \mathbf{s} = 1$, then the expected squared norm of the projection of $\mathbf{x}$ onto $\mathbf{v}$ is always equal to 1 and hence doesn't depend on $\mathbf{v}$.*

Proof: Let $x_v = \mathbf{v}^T \mathbf{x}$ be the projection of $\mathbf{x}$ onto $\mathbf{v}$. Then $E(x_v^2) = E(\mathbf{v}^T \mathbf{x} \mathbf{x}^T \mathbf{v}) = \mathbf{v}^T E(\mathbf{x} \mathbf{x}^T) \mathbf{v}$. Since the covariance matrix of the signal is $E(\mathbf{x} \mathbf{x}^T) = \mathbf{s} \mathbf{s}^T$, we have $E(x_v^2) = \mathbf{v}^T \mathbf{s} \mathbf{s}^T \mathbf{v} = 1$. ■

If the variance of x_v is fixed, then choosing the direction of vector $\mathbf{v}$ that minimizes the noise projection will minimize the estimation error of the luminance, as follows from Lemma 3. Yet, this doesn't guarantee the optimality of $\mathbf{v}$, since the contribution of the luminance component in the inverse formula might depend on $\mathbf{v}$. However, in Lemma 5 we prove that the constraint $\mathbf{v}^T \mathbf{s} = 1$ also fixes the contribution of x_v in the inverse projection.

Lemma 5. *Let $\mathbf{x} = (x_1, x_2, x_3)^T$ be a random vector that satisfies assumption (6.2). Let $\mathbf{T}$ be an invertible color projection, such that*

$$\begin{pmatrix} x_v \\ x_{d_1} \\ x_{d_2} \end{pmatrix} = \mathbf{T} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} \mathbf{v}^T \\ \mathbf{d}_1^T \\ \mathbf{d}_2^T \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} \quad (6.3)$$

where $\mathbf{v}^T \mathbf{s} = 1$ and $\mathbf{d}_1^T \mathbf{s} = \mathbf{d}_2^T \mathbf{s} = 0$. Then, the corresponding inverse projection

is

$$\begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} s_1 \\ s_2 \\ s_3 \end{pmatrix} x_v + \mathbf{R} \begin{pmatrix} x_{d_1} \\ x_{d_2} \end{pmatrix} \quad (6.4)$$

where $\mathbf{R}$ is a 3×2 matrix.

Proof: Let $\mathbf{T}^{-1} = (\mathbf{r}, \mathbf{R})$, where $\mathbf{r}$ is the first column of $\mathbf{T}^{-1}$. From $\mathbf{T}\mathbf{T}^{-1} = \mathbf{I}$ it follows that $\mathbf{v}^T \mathbf{r} = 1$ and $\mathbf{d}_1^T \mathbf{r} = \mathbf{d}_2^T \mathbf{r} = 0$. Hence, clearly $\mathbf{r} = \mathbf{s}$ and we obtain (6.4). ■

To simplify the inverse projection, we use four basis vectors (one luminance and three color differences) instead of three:

$$\begin{aligned} x_v &= \mathbf{v}^T \mathbf{x} \\ x_{d_i} &= x_i - s_i x_v = x_i - s_i \mathbf{v}^T \mathbf{x} \end{aligned} \quad (6.5)$$

One can easily check that the color-difference vectors so defined are orthogonal to $\mathbf{s}$. Although using three color differences instead of typical two, the projection still satisfies Lemma 5 because the new constraint $\mathbf{d}_3^T \mathbf{s} = 0$ ensures $\mathbf{r} = \mathbf{s}$ when $\mathbf{d}_3^T \mathbf{r} = 0$. The inverse projection is then

$$x_i = s_i x_v + x_{d_i} \quad (6.6)$$

Let $\hat{x}_i$ be a denoised version of x_i . From (6.6) it follows that the quality of estimate $\hat{x}_i = s_i \hat{x}_v + \hat{x}_{d_i}$ depends on $\mathbf{v}$ indirectly through the estimation error of $\hat{x}_v$ and $\hat{x}_{d_i}$. The constraint $\mathbf{v}^T \mathbf{s} = 1$ fixes the signal variance of x_v according to Lemma 4, and fixes the contribution of x_v in the inverse projection according to Lemma 5. The estimation error of color differences is comparatively small

and can be neglected. Hence, choosing $\mathbf{v}$ that minimizes the variance of the luminance noise projection leads to optimal denoising. Lemma 6 relates optimal $\mathbf{v}$ to the signal and noise statistics.

Lemma 6. *Let $\mathbf{n} = (n_1, n_2, n_3)$ be the noise vector in RGB space and Σ_n its covariance matrix. Then the optimal vector $\mathbf{v}$ that yields the smallest projected noise variance is*

$$\mathbf{v}_{opt} = \arg \min_{\mathbf{v}^T \mathbf{s} = 1} \mathbf{v}^T \Sigma_n \mathbf{v} = \frac{\Sigma_n^{-1} \mathbf{s}}{\mathbf{s}^T \Sigma_n^{-1} \mathbf{s}} \quad (6.7)$$

Proof: Let $n_v = \mathbf{v}^T \mathbf{n}$ be the projection of $\mathbf{n}$ onto $\mathbf{v}$. The variance of n_v is $E\{n_v^T n_v\} = \mathbf{v}^T \Sigma_n \mathbf{v}$. According to Lagrange multiplier method [93], the minimization of $\mathbf{v}^T \Sigma_n \mathbf{v}$ under the constraint $\mathbf{v}^T \mathbf{s} = 1$ is equivalent to minimization of functional $\mathbf{v}^T \Sigma_n \mathbf{v} - 2\lambda \mathbf{v}^T \mathbf{s}$, where 2λ is the Lagrange parameter. Setting the derivative of the functional to zero yields

$$\frac{d\{\mathbf{v}^T \Sigma_n \mathbf{v} - 2\lambda \mathbf{v}^T \mathbf{s}\}}{d\mathbf{v}} = 2\Sigma_n \mathbf{v} - 2\lambda \mathbf{s} = 0$$

Solving this equation for $\mathbf{v}$ and normalizing the result yields (6.7). ■

Figure 6.2 shows the effect of optimal luminance projection on the denoising performance. It can be easily seen that using $\mathbf{v}_{opt}$ instead of $\mathbf{v} = (0.299, 0.587, 0.114)^T$ from YCbCr leads to better edge preservation and overall image quality.

To obtain $\mathbf{v}_{opt}$ we need the noise covariance matrix $\Sigma_n = E(\mathbf{n}\mathbf{n}^T)$, which is typically unknown beforehand. In Section 6.4 we discuss how to estimate $\Sigma_n = E(\mathbf{n}\mathbf{n}^T)$ directly from the noisy image data.

Figure 6.2: (a) Noiseless luminance. (b) Denoised luminance using $\mathbf{v} = [0.299, 0.587, 0.114]^T$. (c) Denoised luminance using $\mathbf{v}_{opt}$.

6.4 Estimation of Noise Covariances

Wavelet coefficients in the HH subband of the finest scale are typically smaller than the coefficients in other subbands, making it more convenient for noise estimation. For example, in a popular Donoho's estimator [53], the standard deviation of the noise is computed as

$$\hat{\sigma}_n = \frac{\text{Median}(|\tilde{x}_i|)}{0.6745} \quad (6.8)$$

where $\tilde{x}_i$ are the noisy HH wavelet coefficients. However, in images with fine structures, such as textures, the HH coefficients are generally larger, leading to

potentially inaccurate estimation [60].

Donoho's estimator can be straightforwardly extended to color images by applying it to each color component independently. However, this will only produce the estimates of the noise variances and not the noise correlations.

We propose a novel approach capable of estimating the full covariance matrix of the noise by exploiting the similarity property of detail wavelet coefficients. Unlike Donoho's approach, our estimates are derived simultaneously for all color components by solving a system of linear equations. We first discuss the case of uncorrelated noise, i.e., $E(n_i n_j) = 0$ for $i \neq j$, in Section 6.4.1. A more general case of correlated noise is considered in Section 6.4.2.

6.4.1 Estimation of uncorrelated noise variances

Similar to Donoho's estimator, we use the wavelet coefficients in the HH subband of the finest scale. Table 1.1 and Figure 1.6 show that the HH subband coefficients exhibit more similarity than other subbands. We use perfect correlation assumption (6.2) and in addition assume that the expected squared norm of the signals is the same for all color components: $E(x_i x_j) = s_1^2 = s_2^2 = s_3^2 = \bar{s}^2$. Then the covariance matrix of a noisy signal $\tilde{\mathbf{x}} = \mathbf{x} + \mathbf{n}$ is

$$\Sigma_{\tilde{\mathbf{x}}}(i, j) = E\{\tilde{x}_i \tilde{x}_j\} = \begin{cases} \bar{s}^2 & i \neq j \\ \bar{s}^2 + \sigma_{n_i}^2 & i = j \end{cases} \quad (6.9)$$

where $i, j = \{1, 2, 3\}$. Let $\boldsymbol{\theta} = (\bar{s}^2, \sigma_{n_1}^2, \sigma_{n_2}^2, \sigma_{n_3}^2)^T$ be the vector of unknown parameters, and $\mathbf{y} = (\Sigma_{\tilde{\mathbf{x}}}(1, 1), \Sigma_{\tilde{\mathbf{x}}}(2, 2), \Sigma_{\tilde{\mathbf{x}}}(3, 3), \Sigma_{\tilde{\mathbf{x}}}(1, 2), \Sigma_{\tilde{\mathbf{x}}}(1, 3), \Sigma_{\tilde{\mathbf{x}}}(2, 3))^T$ be

Table 6.2: Comparison of normalized errors of Donoho's and the proposed noise estimators. The two types of uncorrelated noise are the same as that in Table 6.1. For correlated noise, the noise standard deviations added to $\{R, G, B\}$ and the correlation coefficients for $\{R\&G, R\&B, G\&B\}$ are equal to $\sigma_n = \{0.1, 0.1, 0.1\}$ and $\{0.4, 0.2, 0.3\}$, respectively, in the case of uniform noise, and equal to $\sigma_n = \{0.05, 0.15, 0.10\}$ and $\{0.3, 0.3, 0.3\}$ in the case of nonuniform noise. These two types of correlated noise remain the same for all other experiments in the chapter.

	Uncorrelated noise			
	Uniform		Nonuniform	
Covariance matrix	Diag.	Off-diag.	Diag.	Off-diag.
Donoho's	0.059	-	0.090	-
Proposed uncorrelated (6.10)	0.012	-	0.020	-
Proposed correlated (6.14)	0.050	0.049	0.089	0.063
	Correlated noise			
	Uniform		Nonuniform	
Covariance matrix	Diag.	Off-diag.	Diag.	Off-diag.
Donoho's	0.059	-	0.089	-
Proposed uncorrelated (6.10)	-	-	-	-
Proposed correlated (6.14)	0.049	0.033	0.087	0.037

the data vector. Rewriting (6.9) in a matrix form

$$\mathbf{y} = \mathbf{B}_0 \boldsymbol{\theta},$$

we obtain an overconstrained system of 6 equations and 4 unknowns. A least squares estimate (LSE) of $\boldsymbol{\theta}$ can be obtained by using the pseudoinverse of $\mathbf{B}_0$ [94]:

$$\hat{\boldsymbol{\theta}} = (\mathbf{B}_0^T \mathbf{B}_0)^{-1} \mathbf{B}_0 \mathbf{y} \quad (6.10)$$

We applied the Donoho's estimator and the proposed algorithm to our test images in Figure 5.2. Table 6.2 shows that the normalized average estimation error $\frac{|\hat{\sigma}_n - \sigma_n|}{\sigma_n}$ of the proposed algorithm is 4-5 times smaller than the error of Donoho's estimator for the case of uncorrelated noise.

6.4.2 Estimation of correlated noise covariances

In general, the noise in different color planes may be correlated:

$$\Sigma_n(i, j) = \sigma_{n_{i,j}}^2 = E \{n_i n_j\} \neq 0, \quad i \neq j$$

For example, a high temperature can cause excessive thermal noise on a color sensor. The noise is temperature dependent and hence is likely to be similar in the local neighborhood of the color sensors [95]. In this case, assuming uncorrelated noise and using estimate (6.10) can lead to poor estimates of the optimal luminance basis vector $\mathbf{v}_{opt}$ and hence poor denoising performance.

For correlated noise we can rewrite (6.9) as

$$\Sigma_{\tilde{x}}(i, j) = \begin{cases} \bar{s}^2 + \sigma_{n_{i,j}}^2 & i \neq j \\ \bar{s}^2 + \sigma_{n_i}^2 & i = j \end{cases} \quad (6.11)$$

where $i, j = \{1, 2, 3\}$. Let $\boldsymbol{\theta}$ be the vector of seven unknown parameters (signal variance, three noise variances, and three noise covariances). Rewriting (6.11) in a matrix form yields

$$\mathbf{y} = \mathbf{B}_1 \boldsymbol{\theta}$$

This is an underconstrained system with 6 equations and 7 unknowns. To provide additional constraints we use Donoho's estimator (6.8) for $\sigma_{n_i}^2$'s, thus adding three more equations to (6.11). Let $\mathbf{y}_d$ be the vector of Donoho's estimated noise variances:

$$\mathbf{y}_d = \mathbf{A} \boldsymbol{\theta} \quad (6.12)$$

Since Donoho's estimator is not as accurate as (6.11), we take (6.11) as a con-

straint and obtain a constrained LSE estimate of (6.12):

$$\hat{\boldsymbol{\theta}} = \arg \min_{\mathbf{y}=\mathbf{B}_1\boldsymbol{\theta}} (\mathbf{y}_d - \mathbf{A}\boldsymbol{\theta})^T (\mathbf{y}_d - \mathbf{A}\boldsymbol{\theta}) \quad (6.13)$$

According to Theorem 19.2.1 in [96], $\hat{\boldsymbol{\theta}}$ is the first part of the solution to the following system of linear equations:

$$\begin{pmatrix} \mathbf{A} & \mathbf{B}_1^T \\ \mathbf{B}_1 & \mathbf{0} \end{pmatrix} \begin{pmatrix} \boldsymbol{\theta} \\ \mathbf{r} \end{pmatrix} = \begin{pmatrix} \mathbf{A}^T \mathbf{y}_d \\ \mathbf{y} \end{pmatrix}$$

Hence,

$$\begin{pmatrix} \hat{\boldsymbol{\theta}} \\ \mathbf{r} \end{pmatrix} = \begin{pmatrix} \mathbf{A} & \mathbf{B}_1^T \\ \mathbf{B}_1 & \mathbf{0} \end{pmatrix}^{-1} \begin{pmatrix} \mathbf{A}^T \mathbf{y}_d \\ \mathbf{y} \end{pmatrix} \quad (6.14)$$

Table 6.2 shows the performance of the proposed estimator (6.14) on the diagonal and off-diagonal entries of the noise covariance matrix. Performance on the diagonal entries is similar to that of the Donoho's estimator. The non-diagonal entries are estimated equally well. Hence the estimator (6.14) can be used in the case when it is unknown a priori whether the noise is correlated or not.

Even though the assumptions of perfect signal correlation and uniform variances may not be satisfied in real life, experimental results suggest that estimator (6.14) is sufficiently precise for the optimal luminance projection-based wavelet denoising, see Table 6.5. The PSNR performance is only 0.15-dB worse compared to that of using the exact noise covariance matrix.

6.5 Denoising of Color-Difference Images

Statistical description of the optimal luminance projection resembles that of grayscale images. Hence the standard grayscale image denoising approaches can be straightforwardly applied to luminance denoising. On the other hand, the color-difference components are much smoother, and might require some modifications to the existing grayscale approaches. The modifications are likely to be algorithm dependent and should be decided on a case by case basis. We consider this problem on the example of uHMT denoising approach.

Compared to grayscale images, the edges of color-difference images are smaller in amplitude, leading to smaller detail wavelet coefficients in the fine scales. For a sufficiently noisy image, the signal variances in these scales are much smaller than the noise variances, $\sigma_{x|\mathcal{S}}^2 < \sigma_{x|\mathcal{L}}^2 \ll \sigma_n^2$. This leads to the following two observations:

1. The variances of noisy coefficients in the two states are similar to each other, i.e. $\sigma_{q|\mathcal{S}}^2 + \sigma_n^2 \approx \sigma_{x|\mathcal{L}}^2 + \sigma_n^2$, making it difficult to distinguish these coefficients
2. The shrinkage parameters corresponding to conditional MMSE estimates are close to zero:

$$\frac{\sigma_{x|\mathcal{S}}^2}{\sigma_{x|\mathcal{S}}^2 + \sigma_n^2} \approx 0, \quad \frac{\sigma_{x|\mathcal{L}}^2}{\sigma_{x|\mathcal{L}}^2 + \sigma_n^2} \approx 0 \quad (6.15)$$

The first observations is likely to lead to misclassification of fine scale coefficients. Since the conditional distributions of large and small coefficients are close to each other, the prior state probabilities $p(\mathcal{S})$, $p(\mathcal{L})$ will be the deciding

factor in computing conditional state probabilities $p(q|\tilde{x})$. Since for a typical image $p(\mathcal{S}) \gg p(\mathcal{L})$, hence $p(\mathcal{S}|\tilde{x}) \gg p(\mathcal{L}|\tilde{x})$ for a large range of $\tilde{x}$. This observation is supported by experiment in Figure 6.3(b): classifying coefficients according to conditional state probabilities puts an overwhelming majority of fine scale coefficients into the Small state.

On the other hand, the detail wavelet coefficients in the coarser scales are still large and should allow for a good separation of large and small coefficients. However, contrary to our expectations, an overwhelming majority of the coarse-scale coefficients are also classified as small, as shown in Figure 6.3(b). This can be explained by a tie between the neighboring scales, which is introduced to exploit the inter-scale correlations [55,60]. The original idea of scale connections was primarily for using coarser scale coefficients to help classification of the finer scale ones. In color-difference images we obtain an opposite effect — wrongly classified fine scale coefficients adversely affect the classification in coarser scales.

We propose to resolve this problem by separating the fine and coarse scales using the following simple test:

$$\begin{cases} \text{Fine} & \hat{\sigma}_{d_i}^2 \leq t\hat{\sigma}_n^2 \\ \text{Coarse} & \hat{\sigma}_{d_i}^2 > t\hat{\sigma}_n^2 \end{cases} \quad (6.16)$$

where $\hat{\sigma}_{d_i}^2$ and $\hat{\sigma}_n^2$ are signal and noise variances, estimated from noisy image data, and t is a tolerance parameter chosen empirically. If the noise dominates the signal, we set the scale to Fine; otherwise we set it to Coarse.

For simplicity, we treat all fine scale coefficients as small, since the conditional denoising rules are similar to each other according to (6.15). The denoising in coarse scales is done the same way as before. Scale separation

yields much better classification and denoising performance, as shown in Figure 6.3(c)(d) and Table 6.3. In particular, the proposed modification yields an average PSNR gain of 3 dB in color-difference images.

Figure 6.3: (a) Color-difference image. The states of wavelet coefficients (b) after uHMT classification, (c) after uHMT classification with Fine/Coarse-Scale separation, and (d) in coarse scale after uHMT classification with Fine/Coarse-Scale separation; Gray and white pixels correspond to small and large coefficient states, respectively.

Table 6.3: PSNR performance (in dB) of uHMT on color-difference images with and without Fine/Coarse-scale separation in the case of uncorrelated noise

Color Image		Uniform noise		Nonuniform noise	
		Without F/C	F/C	Without F/C	F/C
1	Cb	37.46	41.74	38.39	42.53
	Cr	36.68	39.56	35.69	38.66
2	Cb	36.90	39.35	37.99	40.71
	Cr	35.56	36.59	34.55	34.75
3	Cb	36.43	38.66	37.81	40.06
	Cr	36.96	41.05	36.09	39.78
4	Cb	37.69	43.34	38.85	44.38
	Cr	36.14	37.84	35.16	36.88
5	Cb	35.85	36.86	36.78	37.60
	Cr	36.06	37.43	35.08	36.48
Average	Cb	36.87	39.99	37.96	41.06
	Cr	36.28	38.49	35.32	37.31

6.6 Summary of the Proposed Denoising Method

The proposed optimal color projection (OCP) based denoising method consists of the following four key steps:

1. For a given RGB image, use (6.14) to estimate the covariance matrix of the noise Σ_n . If the noise is known to be uncorrelated, then estimator (6.10) can be used to achieve slightly better performance.
2. Use $\hat{\Sigma}_n$, $\mathbf{s} = [1, 1, 1]^T$, and (6.7) to derive the optimal color projection. Applying it to the original RGB image yields the optimal projection of luminance $\tilde{x}_{opt}$ and color differences $\tilde{x}_{d_i}$, as defined in (6.5).
3. Apply the uHMT (or any other grayscale denoising approach) to new color components. If necessary, modify the algorithm to suit the properties of color differences, as we did with uHMT by Fine/Coarse-scale separation.
4. Reconstruct the RGB color components using the inverse color projection

(6.6).

Note that the proposed scheme is not constrained by the uHMT denoising approach. Combining optimal projection with other popular wavelet-based denoising approaches leads to similar performance improvement, as shown in Section 6.7.2.

The complexity of the proposed denoising algorithm primarily depends on the adopted denoising method, and hence is increased by a factor of 3 or 4 (depending on the number of color-difference components) compared to grayscale image denoising. The complexities of the noise estimation and color projection steps are relatively small.

6.7 Experimental Results and Discussion

In our experiments we use 24 test color images shown in Figure 5.2. As a performance measure we adopt the color peak signal-to-noise ratio (CPSNR), defined in (2.1).

6.7.1 Performance contributions of proposed algorithm components

Our proposed algorithm, when coupled with uHMT, consists of three main components (Section 6.6): optimal color projection (OCP), noise estimator, and denoising of color-difference images with Fine/Coarse scale (F/C) separation. In this section we show the effect of each individual component on the resultant denoising performance.

Table 6.4: Comparison of CPSNR performance (in dB) of uHMT in RGB, YCbCr, and the proposed OCP color spaces

	Uncorrelated noise		Correlated noise	
	Uniform	Nonuniform	Uniform	Nonuniform
RGB	27.11	27.19	27.17	27.25
YCbCr	28.22	27.78	27.63	26.91
OCP	28.79	29.51	27.94	29.02

Table 6.5: Comparison of CPSNR performance (in dB) of Donoho's and the proposed noise estimators

	Uncorrelated noise		Correlated noise	
	Uniform	Nonuniform	Uniform	Nonuniform
Donoho	28.92	29.67	27.53	28.92
Proposed	29.18	29.81	28.21	29.31
exact	29.19	29.89	28.28	29.51

To demonstrate the effect of color projection we compare the average CPSNR performance of applying uHMT in RGB, YCbCr and the proposed OCP color spaces in Table 6.4. Denoising in OCP substantially outperforms denoising in RGB by an average of 0.8-2.3 dB depending on the relative noise distribution. The difference between denoising in OCP and YCbCr is more subtle. For the uniform noise, the average OCP gain over YCbCr is only 0.3-0.6 dB. This can be explained by the similarity of the optimum luminance projection vector and vector $Y = (0.299, 0.587, 0.114)$ from YCbCr under the uniform noise. However, the performance gain increases to 1.6-2.1 dB for the case of nonuniform noise.

The proposed noise estimation technique also has substantial effect on the overall denoising performance, especially for the correlated noise. According to

Table 6.6: Comparison of CPSNR performance (in dB) with or without separating Fine/Coarse scales (F/C) in denoising color-difference images.

	Uncorrelated noise		Correlated noise	
	Uniform	Nonuniform	Uniform	Nonuniform
without F/C	28.79	29.51	27.94	29.02
with F/C	29.18	29.81	28.21	29.31

results in Table 6.5, our method outperforms the Donoho's by 0.15-0.3 dB for the uncorrelated noise and by 0.4-0.7 dB for the correlated noise. Moreover, this performance is only about 0.15 dB worse compared with the case when exact noise covariance matrix is known.

The proposed separation of Fine/Coarse scales (F/C) in uHMT substantially improves the denoising performance of color-difference images. However, the color-difference images contribute less to the overall denoising performance compared to the luminance. The overall improvement is 0.3-0.4 dB on average, as shown in Table 6.6.

In summary, color projection is the main factor in the overall denoising performance improvement, followed by the noise estimation and the adaption of denoising to color differences. The improvements are already noticeable for images with uniform uncorrelated noise and become quite substantial in image with correlated and/or nonuniform noise.

6.7.2 OCP with other grayscale wavelet-based denoising approaches

The proposed OCP can be combined with any grayscale wavelet-based denoising approach, including the one developed in Chapter 5; see Table 6.7. Here, Hard-Thr and Soft-Thr are hard and soft thresholding of wavelet coefficients proposed in [3], GSM stands for denoising based on Gaussian Scale Mixtures model, proposed by Portilla *et al.* [61], the grayscale version of the LSAI algorithm [67]¹, and WMCA is our approach proposed in Chapter 5. We used redundant wavelet transform in LSAI, and orthogonal wavelet transform in Hard-Thr, Soft-Thr,

Table 6.7: CPSNR performance (in dB) by applying OCP in different approaches for grayscale images.

		Uncorrelated noise		Correlated noise	
		Uniform	Nonuniform	Uniform	Nonuniform
Hard-Thr	RGB	25.16	25.18	25.16	25.24
	YCbCr	26.16	25.78	25.64	24.94
	OCP	26.64	27.22	25.93	26.95
Soft-Thr	RGB	24.13	24.14	24.12	24.20
	YCbCr	25.03	24.68	24.59	23.95
	OCP	25.31	25.82	24.70	25.80
LSAI (grayscale version)	RGB	28.41	28.29	28.40	28.30
	YCbCr	29.82	29.33	29.15	28.34
	OCP	30.28	30.93	29.42	30.54
GSM	RGB	27.80	27.72	27.67	27.60
	YCbCr	28.90	28.46	28.31	27.52
	OCP	29.72	30.44	28.62	29.74
WMCA	RGB	27.38	27.30	27.38	27.34
	YCbCr	28.58	28.16	27.95	27.21
	OCP	29.19	29.90	28.23	29.36

GSM and WMCA. The performance gains obtained are similar to the ones we observed in uHMT (Table 6.4). For WMCA, the OCP further improves its denoising performance up to 2.1dB. Figure 6.4 shows an example for the improvement in Parrot image. The denoising in RGB space oversmooths the cheek area but reveals some color artifacts in the area above the eye. The YCbCr transformation can remove the color artifacts and the proposed OCP further preserves the texture around the parrot’s cheek region.

6.7.3 Comparison to existing state-of-the-art methods

We also compared the proposed OCP denoising with three state-of-the-art methods, wiener filtering *wiener2*², and two recently proposed wavelet-based de-

¹ There are two version of LSAI algorithm proposed in [67], one for grayscale image denoising and the other for color image denoising. Both can be applied to color images; the former denoises color components independently, while the latter uses joint color processing.

²The spatial filtering, the function “wiener2” in MATLAB is used.

(a) Origin image

(b) Noisy image

(c) RGB space

(d) YCbCr space

(e) OCP space

Figure 6.4: Comparison of LSAI denoising in different color spaces on the Parrot image corrupted by uncorrelated, nonuniform noise.

Table 6.8: CPSNR performance (in dB) of *wiener2*, LSAI, MML, and the proposed OCP methods for uncorrelated noise.

Color Image	Uniform				Nonuniform			
	<i>wiener2</i>	LSAI	MML	Ours	<i>wiener2</i>	LSAI	MML	Ours
1	24.84	25.25	26.67	27.42	24.43	25.26	27.42	28.73
2	27.19	29.17	29.20	30.22	26.43	29.07	29.20	30.48
3	27.05	29.34	29.43	30.54	26.47	29.19	29.66	30.76
4	27.04	29.10	29.34	30.41	26.44	29.08	29.56	30.85
5	25.45	24.98	26.47	27.11	25.04	24.94	27.02	28.11
6	25.31	25.89	27.21	28.01	24.82	25.74	27.37	28.66
7	26.97	27.93	28.51	29.74	26.41	27.96	28.95	30.46
8	24.23	25.00	26.40	27.04	23.85	24.88	26.93	28.13
9	26.82	28.23	28.98	30.24	26.29	28.10	29.32	30.72
10	27.28	30.55	30.28	32.08	26.67	30.24	30.30	32.63
11	26.30	26.99	27.95	29.02	25.83	26.90	28.41	29.79
12	27.08	29.67	29.58	31.04	26.51	29.61	29.70	31.59
13	24.40	24.10	26.08	26.52	24.03	24.22	26.80	27.80
14	26.47	26.72	27.56	28.23	25.96	26.59	27.78	28.50
15	27.09	28.78	28.86	29.96	26.45	28.62	28.80	30.16
16	26.81	29.05	29.53	30.83	26.30	28.96	29.73	31.70
17	27.08	28.37	28.84	30.23	26.50	28.32	29.11	31.00
18	26.40	26.38	27.49	28.19	25.92	26.33	27.76	28.93
19	25.57	26.53	27.73	28.39	25.08	26.31	27.99	29.16
20	26.48	27.21	27.39	28.45	25.67	26.63	26.81	28.03
21	26.02	26.55	27.86	28.76	25.50	26.50	28.32	29.66
22	26.53	27.49	28.09	28.90	25.97	27.40	28.25	29.54
23	27.40	30.25	30.13	31.40	26.77	29.93	30.06	31.46
24	25.63	25.73	26.82	27.52	25.16	25.77	27.19	28.65
Average	26.31	27.47	28.18	29.18	25.77	27.36	28.44	29.81

noising methods, color version of LSAI^{1,3} [67] and MML [68]⁴. For fairness, we adopt an orthogonal wavelet transform for all wavelet-based tested approaches. As shown in Tables 6.8 and 6.9, the proposed approach outperforms all others on every test image.

In particular, for uniform noise, the average CPSNR gains of RGB are 3 dB, 1.7 dB, and 1 dB over *wiener2*, LSAI and MML, respectively, as shown in

³We used the software provided by the authors with a minor modification to allow for different noise variances in the R, G and B components. Without this modification, the algorithm performs poorly in the presence of nonuniform noise.

⁴The paper mentions two alternative parameter estimation schemes: MAP and ML framework. We adopted the latter one since it yields better CPSNR.

Table 6.10. Even larger gains are achieved for nonuniform noise, where average PSNR gains for the noisiest component becomes 5 dB, 2.7 dB, and 1.9 dB, respectively.

Table 6.9: CPSNR performance (in dB) of *wiener2*, LSAI, MML, and the proposed OCP methods for correlated noise.

Color Image	Uniform				Nonuniform			
	<i>wiener2</i>	LSAI	MML	Ours	<i>wiener2</i>	LSAI	MML	Ours
1	24.80	25.09	25.90	26.07	24.32	25.11	26.93	27.48
2	27.18	29.15	28.94	29.68	26.67	29.16	29.50	30.14
3	27.03	29.17	29.03	29.86	26.36	29.14	29.60	30.97
4	27.04	29.03	29.06	29.77	26.41	29.04	29.41	30.39
5	25.40	24.85	25.73	25.86	24.87	24.86	26.62	27.34
6	25.27	25.67	26.53	26.94	24.71	25.76	27.41	28.59
7	26.97	27.79	28.04	28.86	26.23	27.75	28.73	30.01
8	24.20	24.79	25.61	25.62	23.76	24.77	26.71	27.39
9	26.83	28.05	28.45	29.25	26.11	27.97	29.22	30.77
10	27.31	30.29	29.97	31.23	26.52	30.23	30.46	32.56
11	26.29	26.88	27.46	28.05	25.57	26.85	28.11	29.27
12	27.13	29.56	29.23	30.35	26.40	29.18	29.32	30.89
13	24.36	23.98	25.10	24.81	23.88	24.10	26.06	26.28
14	26.46	26.60	27.05	27.33	25.81	26.66	27.75	28.42
15	27.08	28.72	28.45	29.34	26.35	28.37	28.34	29.48
16	26.80	28.92	29.13	29.97	26.10	28.90	29.73	31.52
17	27.10	28.27	28.36	29.35	26.24	27.96	28.38	30.24
18	26.43	26.30	26.97	27.21	25.81	26.33	27.73	28.53
19	25.57	26.33	27.15	27.21	24.96	26.35	28.15	29.02
20	26.55	26.94	26.47	27.41	25.81	26.80	26.16	27.37
21	26.00	26.38	27.29	27.59	25.35	26.43	28.25	28.70
22	26.54	27.34	27.63	28.07	25.93	27.42	28.35	29.05
23	27.41	29.85	29.87	30.74	26.66	29.75	30.23	31.32
24	25.64	25.48	26.28	26.43	25.00	25.50	26.97	27.57
Average	26.31	27.31	27.65	28.21	25.66	27.27	28.25	29.31

We provide here a simple intuition explaining the outstanding performance of the proposed OCP denoising, especially on images with nonuniform noise. Existing denoising techniques, such as *wiener2*, LSAI or grayscale approaches in YCbCr assign approximately equal weight to each color component. In the proposed OCP denoising, less noisy channels have larger weight in the final reconstruction. This results in better denoising, since smaller noise leads to

Table 6.10: Average PSNR performance (in dB) of *wiener2*, LSAI, MML, and the proposed OCP methods.

	Uncorrelated noise							
	Uniform				Nonuniform			
	<i>wiener2</i>	LSAI	MML	Ours	<i>wiener2</i>	LSAI	MML	Ours
R	26.11	27.59	28.20	29.23	23.23	25.88	26.72	28.63
G	26.15	27.60	28.34	29.43	26.15	27.61	29.00	30.45
B	26.22	27.86	28.49	29.59	30.30	30.41	31.58	31.82
	Correlated noise							
	Uniform				Nonuniform			
	<i>wiener2</i>	LSAI	MML	Ours	<i>wiener2</i>	LSAI	MML	Ours
R	26.30	27.20	27.55	28.12	28.83	29.76	30.75	30.67
G	26.45	27.22	27.60	28.14	23.76	25.55	26.58	28.37
B	26.18	27.53	27.83	28.39	25.87	27.51	28.45	29.24

smaller estimation error according to Lemma 3.

Another advantage of the proposed OCP denoising method is better edge preservation. To show this we partition the image into two regions corresponding to edges and smooth areas, and compare their denoising performance. We use Canny edge detection [97] followed by thresholding to obtain the classification. Tables 6.11 and 6.12 show the CPSNR of the two regions for *wiener2*, LSAI, MML, and OCP methods. LSAI uses intra-scale correlations and hence outperforms *wiener2* in smooth regions. However, both algorithms achieve similar performance in the edge regions. The proposed OCP denoising is superior to both LSAI and *wiener2*, outperforming LSAI by 2-3 dB in edge regions and by 1-1.5 dB in smooth regions. MML denoising performs well in the edge region, but still about 0.35-0.9 dB worse compared to OCP. However, its performance in the smooth regions is quite poor, even when compared to the LSAI approach.

The visual quality improvements are even more striking than the PSNR gains. We applied the *wiener2*, LSAI, MML, and uHMT on OCP to several test images. The results are shown in Figures 6.5, 6.6, and 6.7. The proposed OCP denoising leads to much fewer color artifacts and better edge preservation

Table 6.11: CPSNR performance (in dB) in the edge (E) and smooth (S) image regions for uncorrelated noise.

Color Image		Uniform				Nonuniform			
		<i>wiener2</i>	LSAI	MML	Ours	<i>wiener2</i>	LSAI	MML	Ours
1	E	23.78	23.67	25.81	26.17	23.38	23.60	26.71	27.77
	S	26.05	27.37	27.59	28.93	25.63	27.54	28.15	29.81
2	E	26.31	26.61	27.69	28.28	25.70	26.55	27.88	28.86
	S	27.83	32.10	30.48	32.05	26.95	31.91	30.27	31.89
3	E	25.87	25.86	27.24	27.68	25.36	25.78	27.68	27.96
	S	27.69	32.76	30.92	32.86	27.07	32.46	30.95	32.98
4	E	26.32	26.63	27.88	28.47	25.76	26.66	28.32	29.09
	S	27.56	31.93	30.59	32.30	26.93	31.80	30.57	32.48
5	E	24.31	23.28	25.53	25.78	23.93	23.19	26.25	26.95
	S	26.78	27.38	27.51	28.75	26.34	27.45	27.84	29.47
6	E	23.99	23.99	26.05	26.45	23.55	24.02	26.66	27.72
	S	26.79	28.45	28.46	29.88	26.22	27.91	28.05	29.60
7	E	25.88	25.16	26.84	27.48	25.31	25.27	27.53	28.45
	S	27.69	30.72	29.76	31.70	27.12	30.59	29.94	32.09
8	E	22.94	23.30	25.31	25.46	22.56	23.21	26.15	26.98
	S	25.33	26.64	27.31	28.51	24.96	26.47	27.52	29.09
9	E	25.68	25.33	27.22	27.84	25.17	25.30	27.85	28.73
	S	27.56	31.27	30.31	32.40	27.02	30.93	30.36	32.31
10	E	26.34	26.57	28.13	28.92	25.80	26.38	28.52	29.75
	S	27.64	33.60	31.34	34.01	27.00	33.10	31.11	34.28
11	E	25.11	24.55	26.46	27.04	24.65	24.43	27.13	28.01
	S	27.30	29.95	29.30	31.08	26.83	29.93	29.52	31.56
12	E	26.39	27.54	28.40	29.34	25.90	27.72	28.86	30.32
	S	27.57	31.85	30.51	32.58	26.94	31.41	30.32	32.62
Average	E	25.24	25.21	26.88	27.41	24.75	25.18	27.46	28.38
	S	27.15	30.34	29.51	31.25	26.58	30.13	29.55	31.51

compared with the other methods.

Figure 6.5 shows an image of a fence. The OCP denoising preserves edges better than the other methods, while almost completely removing the color artifacts. MML also produces sharp edges, but fails in the smooth regions. Similar observations can be made for the Parrot image in Figure 6.6. The LSAI oversmooths the cheek area but reveals some color artifacts in the area above the eye. The OCP preserves the cheek texture and successfully removes the artifacts. An example of the OCP's superior edge preservation is shown in Figure 6.7. In the original image there are several parallel background thin lines

Table 6.12: CPSNR performance (in dB) in the edge (E) and smooth (S) image regions for correlated noise.

Color Image		Uniform				Nonuniform			
		<i>wiener2</i>	LSAI	MML	Ours	<i>wiener2</i>	LSAI	MML	Ours
1	E	23.57	23.52	24.71	24.47	23.06	23.50	25.95	26.22
	S	26.28	27.18	27.31	28.21	25.85	27.27	28.02	29.02
2	E	26.17	26.64	27.13	27.49	25.64	26.73	27.89	28.48
	S	27.94	31.98	30.60	31.91	27.45	31.82	30.89	31.61
3	E	25.71	25.75	26.53	26.81	25.11	25.86	27.47	28.29
	S	27.77	32.48	30.86	32.46	27.06	32.15	31.02	33.03
4	E	26.22	26.62	27.35	27.67	25.68	26.77	28.07	28.65
	S	27.65	31.75	30.62	31.91	26.94	31.48	30.52	32.00
5	E	24.11	23.20	24.40	24.23	23.58	23.20	25.52	25.96
	S	27.00	27.13	27.38	28.12	26.47	27.18	27.91	29.08
6	E	23.80	23.77	25.14	25.13	23.29	23.92	26.41	27.32
	S	27.00	28.23	28.13	29.30	26.36	28.19	28.44	30.00
7	E	25.61	25.09	26.03	26.39	24.99	25.12	27.01	27.83
	S	27.91	30.45	29.67	31.13	27.06	30.28	30.02	31.85
8	E	22.72	23.12	24.22	23.73	22.24	23.00	25.55	25.92
	S	25.54	26.38	26.84	27.53	25.16	26.50	27.67	28.72
9	E	25.48	25.22	26.41	26.66	24.79	25.20	27.46	28.53
	S	27.76	30.96	30.12	31.70	27.01	30.75	30.55	32.68
10	E	26.21	26.28	27.40	27.80	25.52	26.49	28.36	29.46
	S	27.75	33.38	31.35	33.47	26.91	32.90	31.48	34.43
11	E	24.92	24.52	25.64	25.81	24.25	24.54	26.75	27.50
	S	27.50	29.68	29.27	30.61	26.73	29.52	29.30	31.01
12	E	26.31	27.46	27.88	28.48	25.66	27.18	28.30	29.42
	S	27.73	31.69	30.35	32.13	26.94	31.15	30.09	32.14
Average	E	25.07	25.10	26.07	26.22	24.48	25.13	27.06	27.80
	S	27.32	30.11	29.37	30.71	26.66	29.93	29.66	31.30

that go across the wheel. The OCP denoising is the only method that preserves these lines without introducing color artifacts in the smooth areas.

We also apply the OCP denoising on a real image captured by a commercially available digital camera; see Figure 6.8(a). For this image, the noise covariance estimated by (6.14) is

$$\hat{\Sigma}_n = \begin{pmatrix} 0.0134 & 0.0121 & 0.0124 \\ 0.0121 & 0.0132 & 0.0122 \\ 0.0124 & 0.0122 & 0.0143 \end{pmatrix} \quad (6.17)$$

Figure 6.5: Comparison of different denoising approaches on the Fence image corrupted by uncorrelated, nonuniform noise.

Note that the noise correlations between the color planes are quite large, with values 0.91, 0.89, and 0.89 for R&G, R&B and G&B, respectively. This happens because the camera uses a CFA sensor, and the demosaicking operation projects the noise in each color plane to the others when performing inter-color plane interpolation. Using the noise covariance and (6.7), we can compute the optimal projection vector,

$$\mathbf{v}_{opt} = \begin{pmatrix} 0.362 & 0.465 & 0.173 \end{pmatrix}^T \quad (6.18)$$

Interestingly, the projection vector is close to the luminance vector in the typical YCbCr color space, $\mathbf{v} = (0.299, 0.587, 0.114)^T$ [5]. Based on the optimal vector,

(a) Origin image

(b) Noisy image

(c) *wiener2*

(d) LSAI

(e) MML

(f) Ours

Figure 6.6: Comparison of different denoising approaches on the Parrot image corrupted by uncorrelated, nonuniform noise.

Figure 6.7: Comparison of different denoising approaches on the Bike image corrupted by correlated, nonuniform noise.

Figure 6.8: Real image and the OCP-denoising result.

our method removes the most noise in the real image, as shown in Figure 6.8(b).

6.8 Conclusions

Exploiting inter-color correlations can improve the performance of color image denoising. In this chapter we have developed a novel denoising approach that is based on similarity of detail wavelet coefficients across the color components. The denoising is done by projecting the image onto an optimal luminance/color-difference space adapted to image noise. The projection minimizes the noise in the luminance component, thus preserving the high-frequency contents of the image. On the other hand, the color-difference components are smooth and can be easily denoised regardless of the amount of the noise present.

Another contribution of this chapter is a novel approach to estimation of the inter-color covariance matrix of the noise that is used to determine the optimal color projection. The proposed estimator is shown to be more accu-

rate than the popular Donoho's estimator in the case of uncorrelated noise. It also demonstrates remarkable performance for correlated noise across the color components.

Experimental results suggest that the proposed optimal color projection (OCP) denoising method outperforms the existing solutions not only in PSNR but also in visual quality. In particular, the OCP denoising demonstrates superior edge preservation properties while removing almost all color artifacts.

Chapter 7

Vector Estimation of Wavelet Coefficients for Video Denoising

7.1 Introduction

Most digital still cameras (DSCs) nowadays come with video capturing capabilities. Moreover, the processing chains of digital video cameras are similar to those of DSCs, and hence may suffer from noise in low-light conditions. Image denoising can be straightforwardly extended to videos by applying it to each frame independently. However, better denoising should exploit inter-frame correlations inherently present in video sequences.

The OCP denoising approach proposed in Chapter 6 can not be straightforwardly extended to inter-frame correlations, because it assumes the perfect or near perfect correlation between wavelet coefficients. However, the inter-frame

correlations in videos can be significantly smaller than 1, especially for finer scales; see Table 7.1.

Typically, video denoising have been accomplished by spatio-temporal filtering [98] to exploit the inter-frame correlations, such as IIR filtering [99], Kalman filtering [100] and adaptive 3D Wiener filtering [101]. Kokoram [102,103] used a Bayesian framework to model the correlations between the pixels in the neighboring frames. Motion compensation [104,105] can further improve the denoising performance by increasing inter-frame correlations.

Temporal filtering can be combined with wavelet image denoising by simply denoising the frames independently and then taking a fixed weighted average of pixel intensities across the restored frames [106]. However, it is likely that more flexible averaging offered by Kalman or adaptive Wiener filtering can further improve the performance. One can go one step further and apply the wavelet transform in spatial and temporal domain. However, it was observed that using separable 3D wavelet transform leads to substantial visual artifacts [107]. Therefore, non-separable wavelet transforms are typically used, such as complex 3D wavelet transform [108] or planelets [109]. These methods have high memory and computational requirements, and can be difficult to extend to color videos.

Clearly, a good video denoiser should be able to preserve edges as well as to exploit inter-color and inter-frame correlations present in video sequences. We propose a novel video denoising approach, called vector estimation of wavelet coefficients (VEWC), which offers a systematic way of incorporating these correlations into a popular wavelet Hidden Markov Tree (HMT) based denoising approach [4, 55]. In essence, this approach is equivalent to applying flexible

filtering to wavelet coefficients, where the weights of the filter are derived from the spatial and temporal statistics of neighboring wavelet coefficients. The neighborhood is defined as a set of wavelet coefficients, located in the spatial and temporal proximity of the given coefficient. In case of color videos, the neighborhood includes all color components. Experimental results indicate that VEWC is superior to existing state-of-the-art methods, both in CPSNR and visual quality performance.

Performance of VEWC denoising approach can suffer from changing lighting conditions, or object/camera motion, all of which reduce inter-frame correlations. In this case, our experiments suggest that simple motion alignment and change compensation algorithms can improve the denoising performance.

The chapter is organized as follows. In Section 7.2 we explain the motivation of the proposed denoising approach and discuss related previous work. The vector MMSE estimator is derived in Section 7.3, and a vector HMT model for estimating the parameters required by the vector estimator is developed in Section 7.4. In Section 7.5 we describe simple motion alignment and change compensation techniques that can improve inter-frame correlations and denoising performance. We conclude in Section 7.6 with experimental results that show the superiority of the proposed method over existing solutions, both in CPSNR and visual quality sense.

7.2 Motivation and Previous Work

For color images the detail wavelet coefficients in the finest scales of different color components are strongly correlated, with correlation coefficients ranging

Table 7.1: Average inter-frame correlation of wavelet coefficients of video “MOBILE”.

Scale	Frame k & k+1			Frame k & k+2		
	LH	HL	HH	LH	HL	HH
1	0.781	0.462	0.387	0.437	-0.147	-0.168
2	0.904	0.708	0.645	0.710	0.197	0.097
3	0.972	0.913	0.882	0.912	0.711	0.628
4	0.992	0.974	0.962	0.974	0.909	0.879
5	0.998	0.993	0.989	0.993	0.974	0.963

from 0.98 to 1 for most images [2, 64]. In the thesis introduction, Section 1.3.3, we replicated the experimental results of [2] using Daubechies-4 wavelet [63] and three finest scales for the RGB components of 12 color images in Table 1.1. The results demonstrate that the correlations remain strong even for coarser scales.

Detail wavelet coefficients belonging to neighboring video frames are also correlated. However, unlike inter-color correlations, the inter-frame correlations are generally small for finer scales, but larger for coarser scales, as shown in Table 7.1. This can be explained by the presence of motion affecting primarily high-frequency information of the video frames. Hence a denoising method designed to exploit inter-color correlations may not be suitable for inter-frame correlations. To build a video denoiser capable of exploiting both types of correlations we adopt a multivariate estimation approach on the vector of wavelet coefficients in a local neighborhood.

Wavelet denoising based on vector of neighboring coefficients is not a completely new idea. Scheunders and Driesen [68] extended previous work by Mihcak *et al.* [110] to denoise multispectral images. This approach assumes the coefficients in the local neighborhood to have similar variance, and uses a two-step estimator for denoising. In the first step local neighboring components are used to compute local signal variance. Wavelet coefficients normalized by the estimated variance are shown to be approximately Gaussian and hence are re-

stored using a simple linear estimator in the second step. Such an estimator is sub-optimal because the second step doesn't take into account the uncertainty associated with the variance estimation.

Wavelet denoising based on Gaussian Scale Mixtures (GSM) model has been proposed by Portilla *et al.* [61]. The authors assumed that, given a common hidden scale parameter, each vector of neighboring wavelet coefficients is jointly Gaussian. However, to avoid sub-optimality of two-step estimation, they used neighboring vectors to compute the conditional probabilities for all possible values of scaling coefficients. Then the resultant Bayesian least squares estimator is obtained by integrating conditional linear MMSE estimates weighted by the conditional probabilities of the scaling coefficients.

Our approach is a vector extension of wavelet HMT based denoising [4, 55], which classifies wavelet coefficients to two states, small and large, as discussed in Section 1.3.2. At first glance, HMT-based denoising appears to be similar to the scalar version of GSM-based denoising [61] with binary scaling coefficient. However, HMT-based approach is superior to GSM-based approach in the way the posterior probabilities of the scaling coefficients are estimated. GSM-based approach performs this estimation independently for each wavelet vector, while in HMT-based approach the estimation is done jointly over all coefficients along the wavelets quad-tree, which will be discussed in greater detail in Section 7.4. Furthermore, GSM model assumes the signal correlation matrix to be the same for all values of the scaling coefficients, an assumption that is not necessarily satisfied in practice (see Section 7.4 for details). On the other hand, HMT model allows for state-dependent modeling of the signal covariance matrix.

7.3 Vector Estimator

We extend the scalar estimation (1.7) to exploit all possible correlations present in video sequences. Let x_i be an arbitrary wavelet coefficient and $\mathbf{x}$ be a vector of size M composed of x_i and its neighbors. The neighborhood can include coefficients in the same scale, or the same location in the neighboring scales or frames. For color videos, the neighborhood contains all color components.

For example, for a grayscale image, we can consider a standard 4-neighborhood inside the scale and a parent from the coarser scale, yielding $M = 6$. For a video one can take a neighborhood of K frames before and after the current frame yielding vectors of size $M = 2K+6$ (grayscale) or $M = 6K+18$ (color). Increasing the size of vector $\mathbf{x}$ leads to a higher computational cost, so in practice smaller vectors incorporating only some of the correlations could be used. In our experiments we restricted the neighborhood to the three color components (RGB) and neighboring frames, and hence $M = 6K+3$. See Figure 7.1 for illustrations on how the vectors are yielded.

Let $\tilde{\mathbf{x}} = \mathbf{x} + \mathbf{n}$ be a version of $\mathbf{x}$ corrupted by zero-mean Gaussian noise. Let s_i be the state of x_i and $\mathbf{s}$ be the vector of all states of coefficients in $\mathbf{x}$. Since the signal vector and its noisy version are conditionally Gaussian, the conditional mean estimate, also the minimum-mean-squared-error (MMSE) estimate of $\mathbf{x}$ given $\tilde{\mathbf{x}}$ and $\mathbf{s}$ is a linear function of $\tilde{\mathbf{x}}$ (see [111]):

$$E(\mathbf{x}|\tilde{\mathbf{x}}, \mathbf{s}) = \Sigma_{x|s}(\Sigma_{x|s} + \Sigma_n)^{-1}\tilde{\mathbf{x}} \quad (7.1)$$

where $\Sigma_{x|s}$ and Σ_n are the conditional signal and noise covariance matrices respectively.

Figure 7.1: Vector of coefficients (for $K = 1$) in video sequence.

Proof: Let $\mathbf{A}\tilde{\mathbf{x}}$ be a linear estimate of $\mathbf{x}$. The MMSE estimate of $\mathbf{x}$ given $\tilde{\mathbf{x}}$ and $\mathbf{s}$ minimizes a functional $E[(\mathbf{A}\tilde{\mathbf{x}} - \mathbf{x})^T(\mathbf{A}\tilde{\mathbf{x}} - \mathbf{x})]$. Derive the functional,

$$\begin{aligned} E[(\mathbf{A}\tilde{\mathbf{x}} - \mathbf{x})^T(\mathbf{A}\tilde{\mathbf{x}} - \mathbf{x})] &= \text{Tr} \{ E[(\mathbf{A}\tilde{\mathbf{x}} - \mathbf{x})(\mathbf{A}\tilde{\mathbf{x}} - \mathbf{x})^T] \} \\ &= \text{Tr} [(\mathbf{A} - \mathbf{I})\Sigma_{x|s}(\mathbf{A} - \mathbf{I})^T + \mathbf{A}\Sigma_n\mathbf{A}^T] \end{aligned}$$

where Tr is the trace operation of a matrix. Setting the derivative of the functional to zero yields

$$\Sigma_{x|s}^T(\mathbf{A} - \mathbf{I}) + \Sigma_n^T\mathbf{A} = 0 \Rightarrow \mathbf{A} = \Sigma_{x|s}^T(\Sigma_{x|s} + \Sigma_n)^{-T}$$

Since the covariance matrix is symmetrical, i.e., $\Sigma_{x|s}^T = \Sigma_{x|s}$ and $\Sigma_n^T = \Sigma_n$, we have (7.1). ■

It follows that the conditional mean estimate of $\mathbf{x}$ given $\tilde{\mathbf{x}}$ is

$$E(\mathbf{x}|\tilde{\mathbf{x}}) = \sum_{\mathbf{s}} p(\mathbf{s}|\tilde{\mathbf{x}})\Sigma_{x|s}(\Sigma_{x|s} + \Sigma_n)^{-1}\tilde{\mathbf{x}} \quad (7.2)$$

where $p(\mathbf{s}|\tilde{\mathbf{x}})$ is the conditional probability of state $\mathbf{s}$ given $\tilde{\mathbf{x}}$. Estimate (7.2) is difficult to compute in general, since the number of summation terms grows exponentially with the vector size K . For example, a color video with $K = 1$ yields $2^{6K+3} = 512$ possibilities for vector $\mathbf{s}$ and hence 512 different conditional covariance matrices $\Sigma_{x|\mathbf{s}}$. Moreover, to compute the posterior probabilities $p(\mathbf{s}|\tilde{\mathbf{x}})$ one would need to use upward-downward algorithm with 512 states for each coefficient vector, which is computationally quite intractable.

To avoid the increase in complexity we assume that all coefficients in $\mathbf{x}$ are in the same state q and use the following estimator:

$$\hat{\mathbf{x}} = \sum_{q \in \{\mathcal{S}, \mathcal{L}\}} p(q|\tilde{\mathbf{x}}) \Sigma_{x|q} (\Sigma_{x|q} + \Sigma_n)^{-1} \tilde{\mathbf{x}} \quad (7.3)$$

This assumption is similar to the assumption made in the GSM-based approach [61], where the components of the vector were assumed to have the same scaling coefficient.

Note that (7.3) provides the estimate of each coefficient in $\mathbf{x}$. Partitioning the video into a set of non-overlapping neighborhoods is likely to produce artifacts at the discontinuities of neighborhood boundaries [61]. Hence we build a set of overlapping neighborhoods around each wavelet coefficient, and use each neighborhood to denoise only its center coefficient x_c (three coefficients for a color video).

There are three unknowns in (7.3): the conditional state probabilities $p(q|\tilde{\mathbf{x}})$, $q \in \{\mathcal{S}, \mathcal{L}\}$, the noise covariance matrix Σ_n , and the conditional signal covariance matrices $\Sigma_{x|q}$. We assume uncorrelated noise between the RGB color components and frames, and adopt the Donoho's estimator [53] for Σ_n . To

compute $p(q|\tilde{\mathbf{x}})$ and $\Sigma_{x|q}$ we extend the scalar HMT model to the case of vector $\mathbf{x}$ in Section 7.4.

7.4 Vector Hidden Markov Tree

For a noisy scalar wavelet coefficient $\tilde{x}$, we can estimate its conditional state probabilities $p(q|\tilde{x})$ ($q \in \{\mathcal{S}, \mathcal{L}\}$) using

$$p(q|\tilde{x}) = \frac{p(q)f(\tilde{x}|q)}{\sum_{s \in \{\mathcal{S}, \mathcal{L}\}} p(s)f(\tilde{x}|s)}$$

where $p(q)$ is the prior state probabilities, and $f(\tilde{x}|q)$ is the conditional probability density function, defined from the $\tilde{x}$'s variances $\sigma_{\tilde{x}|q}^2 = \sigma_{x|q}^2 + \sigma_n^2$ in (5.11). The more accurate estimate can be obtained based on the scalar HMT model by using parent-child state transition matrix $p(q_i|q_j)$ to propagate the state information throughout the tree of wavelet coefficients [55]. The upward-downward algorithm (also called forward-backward algorithm [112]) is an efficient method for propagating this information. The upward step provides the information about the fine-scale wavelet coefficients to the states of the coarse-scale wavelet coefficients; the down step propagates the information about the coarse-scale wavelet coefficients down to the states of the fine-scale wavelet coefficients. Combining the information from these two steps obtains the estimate of $p(q|\tilde{x})$ for each wavelet coefficient in the tree. Details of the HMT and upward-downward algorithms can be found in [55].

The upward-downward algorithm requires three groups of parameters, the prior state probabilities $p(\mathcal{S})$ and $p(\mathcal{L})$, conditional variances of wavelet coefficients $\sigma_{x|\mathcal{S}}^2$ and $\sigma_{x|\mathcal{L}}^2$, and parent-child state transition matrix $p(q_i|q_j)$. Crouse

et al. [55] proposed an EM algorithm to estimate these parameters for each scale. Romberg *et al.* [4] later developed a universal parameter model, which was shown to achieve performance comparable to EM algorithm based approach at a much lower computational cost.

In the scalar HMT model, conditional variances $\sigma_{x|q}^2$ are used to define the conditional pdf of scalar x as (5.11). The upward-downward algorithm can be straightforwardly extended to vector $\mathbf{x}$ by substituting $\sigma_{x|q}^2$ for the conditional covariance matrices $\Sigma_{x|q}$. The conditional pdf of vector $\mathbf{x}$ then becomes

$$f(\mathbf{x}|q) = \frac{1}{\sqrt{(2\pi)^M \det \Sigma_{x|q}}} \exp \left(-\frac{\mathbf{x}^T \Sigma_{x|q} \mathbf{x}}{2} \right) \quad (7.4)$$

There are several possible ways to go about estimating the unknown parameters. The most straightforward one is by extending the EM algorithm [55] to the vector case. This, however, would require iterating between the upward-downward algorithm and the parameter updates, which may not be practical in case of video because of the sheer size of data involved.

One way to avoid iterations is by extending the scalar universal model [4] to the vector case. This means preserving the models of $p(q)$ and $p(q_i|q_j)$, and providing a new model for $\Sigma_{x|q}$. Furthermore, one can preserve the models of $\sigma_{x|q}^2$ and consider only the remaining conditional correlation matrices $\mathbf{R}_{x|q}$. For example, if the vector $\mathbf{x}$ contains coefficients of the same scale and hence each coefficient x_i in $\mathbf{x}$ has the same variance model $\sigma_{x|q}^2$, we obtain

$$\Sigma_{x|q} = \sigma_{x|q}^2 \mathbf{R}_{x|q} \quad (7.5)$$

The downside of this approach is the inability to model the inherent vari-

ability of inter-frame correlations. For example, the horizontal motion in video “MOBILE” causes larger inter-frame correlations in the LH subbands compared with the HL and HH subbands, as shown in Table 7.1. On the other hand, a video with a vertical motion will cause larger inter-frame correlations in the HL subbands. A universal model cannot be expected to provide a sufficient fit in such a changing environment.

Alternatively, one can assume that $\mathbf{R}_{x|q}$ doesn’t depend on state q , and estimate $\mathbf{R}_x$ directly from the noisy data. This approach was originally proposed by Portilla *et al.* [61] for their GSM model. In our context

$$\mathbf{R}_x = \frac{\hat{\Sigma}_x}{\hat{\sigma}_x^2} = \frac{E(\tilde{\mathbf{x}}\tilde{\mathbf{x}}^T) - \Sigma_n}{E[E(x_i^2)]} \quad (7.6)$$

However, assuming the same correlations in two states doesn’t provide a good description of the real data, especially for the finest scales, as follows from Table 7.2. For example, inter-color correlations are about 20% larger for the coefficients in the large state. The difference in inter-frame correlations is even more substantial.

Hence, we propose to use correlation matrix (7.6) for computing the conditional state probabilities $p(q|\tilde{\mathbf{x}})$ (via the upward-downward algorithm [55]). We then update the conditional signal covariance matrices using

$$\hat{\Sigma}_{x|q} = \frac{\sum_{\tilde{\mathbf{x}}} \tilde{\mathbf{x}}\tilde{\mathbf{x}}^T \hat{p}(q|\tilde{\mathbf{x}})}{\sum_{\tilde{\mathbf{x}}} \hat{p}(q|\tilde{\mathbf{x}})} - \Sigma_n \quad (7.7)$$

and apply it in the vector MMSE estimate (7.3).

In summary, the proposed denoising algorithm via the vector estimation of wavelet coefficients (VEWC) consists of the following five steps:

Table 7.2: Average correlations of wavelet coefficients in two states.

State	Scale	Inter-color			Inter-frame	
		R & G	R & B	G & B	k & k+1	k & k+2
$\mathcal{S}$	1	0.784	0.705	0.798	0.294	0.058
	2	0.852	0.772	0.870	0.628	0.338
	3	0.773	0.627	0.833	0.851	0.677
	4	0.751	0.579	0.827	0.952	0.879
	5	0.792	0.623	0.796	0.983	0.949
$\mathcal{L}$	1	0.978	0.960	0.980	0.546	0.039
	2	0.969	0.946	0.978	0.753	0.334
	3	0.927	0.871	0.950	0.924	0.751
	4	0.911	0.827	0.925	0.976	0.921
	5	0.892	0.780	0.906	0.993	0.977

1. Construct a noisy vector $\tilde{\mathbf{x}}$ by combining the wavelet coefficients of the current frame $\mathcal{F}$ together with the neighborhoods in K frames before and after it.
2. For each subband, use the estimate of the signal correlation matrix (7.6) and the universal model for $\sigma_{x|q}^2$ to compute the conditional signal covariance matrix $\hat{\Sigma}_{x|q}$ according to (7.5).
3. Use $\hat{\Sigma}_{x|q}$ in the upward-downward algorithm [55] to estimate the conditional state probabilities $\hat{p}(q|\tilde{\mathbf{x}})$.
4. Based on the estimated $\hat{p}(q|\tilde{\mathbf{x}})$, update the signal covariance matrices $\hat{\Sigma}_{x|q}$ according to (7.7).
5. Apply the estimated parameters in the vector MMSE estimator (7.3) to denoise the wavelet coefficients of frame $\mathcal{F}$.

One may consider iterating between steps 3 and 4 to obtain more precise estimates of the conditional signal covariance matrices. However, our experimental results show that performing further iterations leads only to minor improvements, which might not justify the additional computational cost.

Compensating for camera and object motion can increase inter-frame correlations and hence lead to better denoising performance. Hence, we have also investigated the effect of simple motion alignment and change compensation algorithms on the performance of the proposed video denoiser. These motion compensation algorithms are described in Section 7.5.

7.5 Motion Alignment and Change Compensation

Taking motion into account can improve the denoising performance. Even basic motion alignment and change compensation approaches, such as the ones described in this section, lead to substantial improvement as follows from the experimental results in Section 7.6. We expect that larger performance gains can be achieved by using more advanced or sophisticated motion alignment and change compensation techniques.

7.5.1 Motion alignment

Neighboring frames in a video sequence contain similar regions, possibly misplaced from each other due to camera or object motion [113]. The misalignment leads to decrease in correlation between neighboring frames and hence reduces the denoising performance.

The various kinds of motion present in video sequences include global translation and rotation due to camera motion, camera zoom, object translation, object rotation, etc [113]. As an example, we consider here only the global

translation. Let $\mathcal{F}$ be the current frame and $\mathcal{F}_k$ be its neighboring frame, related to $\mathcal{F}$ by a global integer-pixel translation (t_1, t_2) . We adopt simple image registration approach [114] to estimate (t_1, t_2)

$$(\hat{t}_1, \hat{t}_2) = \arg \min_{i,j} E \{ [\mathcal{F}_k(n+i, m+j) - \mathcal{F}(n, m)]^2 \} \quad (7.8)$$

and align the frame $\mathcal{F}_k$ to $\mathcal{F}$ by

$$\mathcal{F}'_k(m, n) = \mathcal{F}_k(m + \hat{t}_1, n + \hat{t}_2) \quad (7.9)$$

Translation estimation can be improved if the frames are first denoised independently from each other and then used in (7.8).

7.5.2 Change compensation

For some regions there may be no corresponding regions in the neighboring frames, as shown in Figure 7.2 the frames 90 and 91 of video “NEWS”. For such regions, joint denoising may actually lead to worse results than denoising the frames independently. To isolate such regions we can use existing change detection methods (see [115] for an excellent review).

Here we adopt a simple change detection based on pair-wise pixel differences. Let $\mathcal{F}$ and $\mathcal{F}'_k$ be the current frame and its neighboring frame after motion alignment. We first obtain a changed pixel map $\mathcal{M}_k$ by thresholding the frame differences:

$$\mathcal{M}_k(m, n) = \begin{cases} 1 & |\mathcal{F}(m, n) - \mathcal{F}'_k(m, n)| > T \\ 0 & \text{otherwise} \end{cases} \quad (7.10)$$

Figure 7.2: Two neighboring frames in test video “News”.

where T is a pre-defined threshold. To decrease the noise influence, one can denoise the frames independently prior to change detection. Remaining noise can still lead to spurious errors, which can be reduced by applying some morphological operations (opening and closing) [116] on the binary image $\mathcal{M}_k$. In our experiments, we used a 2×2 all-1 matrix as a structuring element for the morphological operations.

The resultant changed pixel map $\mathcal{M}_k$ is used to exclude the corresponding regions from denoising of the current video frame.

7.6 Experimental Results

To evaluate the performance of the proposed denoising approach, we conducted experiments on a set of six test videos shown in Figure 7.3 and Table 7.3, and adopted the CPSNR metric, which measures the average Mean Square Error (MSE) of the denoised color components of the frames and is defined as

$$\text{CPSNR} = -10 \log_{10} \left(\frac{1}{3} \sum_{c \in \{R, G, B\}} \text{MSE}_c \right).$$

Table 7.3: Test videos.

Video	Size	Frame no.	noisy (dB)
MOBILE	352×288	300	20.35
FOREMAN	352×288	300	20.21
HARBOUR	352×288	300	19.43
CITY	352×288	300	19.36
NEWS	352×288	300	19.95
GARDEN	352×240	115	19.69

There are many factors that are contributing to the overall performance of our approach: inter-color correlations, inter-frame correlations, the estimation of the conditional state probabilities and conditional covariance matrices. We evaluate individual contributions of these components in Sections 7.6.1-7.6.2. In Section 7.6.3 we show the improvements obtained by the simple motion compensation and change detection schemes used. Next, our approach is compared with several existing video denoising techniques in Section 7.6.4. Finally, in Section 7.6.5 we compare the proposed VEWC to the OCP denoising approach proposed in Chapter 6.

7.6.1 Contribution of inter-color and inter-frame correlations

The proposed vector denoising approach is flexible and can be used to exploit either inter-color correlations or inter-frame correlations, or both. Table 7.4 shows the contribution of each type of correlation. Here $M = 1$ corresponds to denoising of each color component and each frame independently; $M = 3$, joint denoising of three color components (RGB) on each frame independently; and $M = 9$, joint denoising of three neighboring frames and three color components.

(a) MOBILE

(b) FOREMAN

(c) HARBOUR

(d) CITY

(e) NEWS

(f) GARDEN

Figure 7.3: Test videos.

Table 7.4: Average CPSNR performance for denoising videos using vectors $\mathbf{x}$ with different dimension M

Video	$M = 1$	$M = 3$	$M = 9$	TAF, $M = 3$
MOBILE	24.76	26.15	27.66	26.19
FOREMAN	29.07	30.80	32.23	31.15
HARBOUR	27.10	30.48	32.35	31.10
CITY	28.73	31.64	32.58	31.58
NEWS	27.20	30.42	32.32	31.21
GARDEN	23.57	25.86	26.22	25.53

The incorporation of inter-color correlations leads to 1.4-3.4 dB improvement in the CPSNR performance. The gain from incorporating the inter-frame correlations is more modest because of camera/object motion. For videos with little motion, such as “NEWS”, the CPSNR gain is quite large at 1.9 dB. On the other hand, for videos with substantial motion, such as “GARDEN”, the gain decreases to 0.36 dB. This can be explained by the reduced inter-frame correlations due to the motion. Motion alignment and change compensation can help in denoising such videos, as shown later in Section 7.6.3.

Table 7.4 also shows the results of applying a fixed temporal averaging filter (TAF) to the frames denoised independently ($M = 3$), where we chose the same filter as proposed in [106]. As expected, the performance of the fixed filter is worse than that of the proposed adaptive filter (see column $M = 9$ of Table 7.4). Moreover, for two videos, “CITY” and “GARDEN”, the performance of TAF is even worse than that without temporal filtering (compare column $M = 3$ and column TAF $M = 3$). This happens because the motion in the two videos substantially reduces the inter-frame correlations. Applying a fixed temporal filter in this case leads to averaging of wavelet coefficients belonging to different objects, resulting in drop of CPSNR.

Table 7.5: Influence of conditional signal covariance matrix update on the CP-SNR performance.

Video	No update	Single update	Second iteration
MOBILE	27.17	27.66	27.69
FOREMAN	31.79	32.23	32.27
HARBOUR	31.97	32.35	32.38
CITY	32.27	32.58	32.64
NEWS	31.76	32.32	32.34
GARDEN	25.37	26.22	26.30

7.6.2 Contribution of signal covariance matrix estimation

Computing correlation matrix using (7.6) constrains it to be the same for two states, which might not be sufficiently accurate for denoising purposes. The “GARDEN” video provides an extreme case, where the denoising performance of such simplistic modelling is even worse than the performance of denoising the video frame by frame, see Tables 7.4 and 7.5. Improving estimation via EM algorithm requires computationally expensive iterations, which might not be practical in the case of video data. We propose to update the matrix using (7.7), prior to applying it in MMSE vector estimator (7.3). The significance of this update is shown in Table 7.5. The update improves the CPSNR performance by 0.3-0.8 dB. The next iteration improves the performance by less than 0.08 dB (see the last column of Table 7.5), but increases the computational burden by 80%.

Table 7.6: Influence of motion alignment (MA) and change compensation (CC).

Video	Overall			Changed regions		
	none	MA	MA&CC	none	MA	MA&CC
MOBILE	27.66	27.83	27.95	22.95	22.66	23.87
FOREMAN	32.23	32.36	32.37	27.35	27.06	27.48
HARBOUR	32.35	32.40	32.61	27.20	26.88	28.71
CITY	32.58	33.08	33.04	30.26	29.60	29.65
NEWS	32.32	32.32	32.48	26.45	26.45	27.51
GARDEN	26.22	26.99	26.99	24.85	24.40	25.07

7.6.3 Contribution of motion alignment and change compensation

Incorporating motion alignment (MA) and change compensation (CC) can improve the denoising performance by increasing the inter-frame correlations, as shown in Table 7.6. For example, in videos with substantial camera motion (“CITY” and “GARDEN”), motion alignment improves the performance by 0.5-0.7 dB. However, there is no improvement for the video with no camera motion, such as “NEWS”. Change compensation leads to much more modest overall improvement, only about 0.1 dB. Nevertheless, the improvement is quite substantial (up to 1.8 dB) when we restrict the attention to the changed regions only (see the right column of Table 7.6).

7.6.4 Comparison with existing methods

Table 7.7 compares the denoising CPSNR performance between the wiener filtering *wiener3*¹, a recent wavelet-based method CWDT [106]², and the proposed method VEWC. As shown in Table 7.7, the proposed method improves CPSNR

¹The method applies adaptive wiener filtering in spatial and temporal domains. The implementation is obtained by extending the 2-D function “*wiener2*” in MATLAB to 3-D.

²The method applies a temporal-average filtering to the frames restored by the LSAI image denoising method [67].

Table 7.7: Comparison of average CPSNR performance.

Video	noisy	<i>wiener3</i>	CWDT	VEWC
MOBILE	20.35	24.67	26.12	27.66
FOREMAN	20.21	28.76	30.84	32.23
HARBOUR	19.43	26.43	28.28	32.35
CITY	19.36	28.23	29.53	32.58
NEWS	19.95	28.56	30.26	32.32
GARDEN	19.69	21.05	24.36	26.22

by 3.0-5.9 dB compared to *wiener3*, and 1.5-4.0 dB compared to CWDT. Figure 7.4 shows that this improvement is consistent over all video frames.

We have also compared our method to a recently proposed video denoising approach by Kokaram [103]. However, since we couldn't obtain the software and our implementation couldn't obtain the same results as reported, we simply compared the experimental results of denoising video "MOBILE" under the same noise conditions as reported in Figure 12 of [103]. According to the results, Kokaram's approach achieves 4 dB performance gain over noisy frames, while VEWC achieves a gain of more than 7 dB.

The superiority of VEWC approach can also be confirmed visually. We compared selected denoised frames from *wiener3*, CWDT, and VEWC in Figures 7.5, 7.6, and 7.7. Compared with other approaches, the VEWC denoising leads to much fewer color artifacts and better edge preservation. For example, when applied to the cluttered frame in Figure 7.5, our approach better preserves the edges than *wiener3* and has much fewer color artifacts than CWDT. Similar observation can be made from the colorful frame shown in Figure 7.6. The VEWC denoising here preserves more edges on the leaves and has fewer artifacts in the upper-right corner. Flower field edges and colors in Figure 7.7 are better preserved in the VEWC denoising compared with the other two methods.

Figure 7.4: CPSNR performance comparison along video frame.

7.6.5 Comparison of VEWC and OCP denoising

The proposed VEWC denoising approach can also be applied to denoise color images. Compared to the OCP denoising approach proposed in Chapter 6, VEWC is more general — it doesn't need the assumption of perfect correlation across color components. It is easy to show that given perfect inter-color correlations and correct classification of wavelet coefficients, the vector MMSE estimate (7.1) used in VEWC has the same error as the OCP-based scalar esti-

(a) Original

(b) noisy

(c) *wiener3*

(d) CWDT

(e) VWC

(f) VWC&MA&CC

Figure 7.5: Comparison of denoised Frame 3 of “MOBILE”.

(a) Original

(b) noisy

(c) *wiener3*

(d) CWDT

(e) VEW

(f) VEW&MA&CC

Figure 7.6: Comparison of denoised Frame 55 of “MOBILE”.

(a) Original

(b) noisy

(c) *wiener3*

(d) CWDT

(e) VEW

(f) VEW&CC&MA

Figure 7.7: Comparison of denoised Frame 41 of “GARDEN”.

mate. The proof is given in Lemma 8, which can be found in Appendix C.

Table 7.8: CPSNR performance (in dB) of the proposed OCP and VEWC denoising methods.

Color Image	Uncorrelated noise				Correlated noise			
	Uniform		Nonuniform		Uniform		Nonuniform	
	OCP	VEWC	OCP	VEWC	OCP	VEWC	OCP	VEWC
1	27.42	27.66	28.73	29.04	26.07	26.40	27.48	27.89
2	30.22	30.47	30.48	30.97	29.68	29.89	30.14	30.79
3	30.54	30.78	30.76	31.37	29.86	30.00	30.97	31.13
4	30.41	30.58	30.85	31.35	29.77	29.87	30.39	30.72
5	27.11	27.29	28.11	28.27	25.86	26.09	27.34	27.43
6	28.01	28.23	28.66	28.85	26.94	27.16	28.59	28.72
7	29.74	29.79	30.46	30.76	28.86	28.87	30.01	30.11
8	27.04	27.46	28.13	28.46	25.62	26.22	27.39	27.86
9	30.24	30.38	30.72	31.23	29.25	29.36	30.77	30.82
10	32.08	32.38	32.63	33.05	31.23	31.44	32.56	32.88
11	29.02	29.16	29.79	30.11	28.05	28.15	29.27	29.48
12	31.04	31.29	31.59	31.99	30.35	30.56	30.89	31.33
13	26.52	26.90	27.80	28.20	24.81	25.34	26.28	26.78
14	28.23	28.32	28.50	28.77	27.33	27.48	28.42	28.57
15	29.96	30.07	30.16	30.45	29.34	29.44	29.48	29.73
16	30.83	31.06	31.70	31.95	29.97	30.12	31.52	31.64
17	30.23	30.29	31.00	31.16	29.35	29.39	30.24	30.28
18	28.19	28.37	28.93	29.19	27.21	27.39	28.53	28.63
19	28.39	28.79	29.16	29.66	27.21	27.76	29.02	29.37
20	28.45	28.47	28.03	28.14	27.41	27.32	27.37	27.17
21	28.76	28.94	29.66	29.95	27.59	27.77	28.70	29.34
22	28.90	29.03	29.54	29.68	28.07	28.19	29.05	29.25
23	31.40	31.66	31.46	31.91	30.74	30.94	31.32	31.86
24	27.52	27.76	28.65	28.82	26.43	26.72	27.57	27.85
Average	29.18	29.38	29.81	30.14	28.21	28.41	29.31	29.57

In reality, the detail wavelet coefficients of typical color images are strongly but not perfectly correlated, which may decrease the performance of the OCP-based denoising, and could lead to a small disadvantage over the VEWC approach, as shown in Table 7.8. The performance difference is 0.2-0.3dB on average, and is difficult to observe visually.

However, the OCP approach offers an advantage over VEWC in that it can be combined with any grayscale image denoising approach while VEWC is currently limited to HMT model. Hence, OCP combined with an advanced

grayscale denoising method can potentially outperform VEWC in color image denoising.

7.7 Conclusions

Exploiting inter-color and inter-frame correlations can improve the denoising performance of video sequences. In this chapter we have proposed a novel denoising approach based on a vector extension of a popular HMT model for wavelet coefficients [55]. By vector we mean a set of wavelet coefficients in local color and frame neighborhood. Flexible subband and temporal adaptation is achieved by estimating inter-color and inter-frame correlations directly from the noisy video data.

Experimental results suggest that the proposed denoising method, which we call vector estimation of wavelet coefficients (VEWC), outperforms the existing solutions, not only in CPSNR but also in visual quality. The VEWC denoising removes color artifacts while preserving edges. The experimental results also show that incorporating motion alignment and change compensation can further improve the performance.

Chapter 8

Conclusions and Future Work

In this thesis we have considered several interesting research problems related to the DSC processing chain, which involves compression, demosaicking, and quality improvements of digital images acquired through a CFA sensor.

In Part I of the thesis we have concentrated on DSC's processing error modeling and analysis, which allowed us to predict the superiority conditions of the alternative processing chain (reversed demosaicking and compression) over conventional processing approach. Using mathematical modeling of the reconstruction error we have shown that the alternative chain can outperform the conventional one at low compression ratios. The conclusions have been verified by experimental results, and also well agree with and explain the existing empirical studies.

The error modeling has also allowed us to explain the error inhomogeneity of wavelet-based compression. By modeling the error of even/odd pixel positions, we have found that the cause of inhomogeneity is mainly due to the asymmetric

filtering of quantization errors following the upsampling in wavelet synthesis process. Based on the models developed, we have also designed a framework to create balanced wavelets with reduced error inhomogeneity, at a small cost of overall PSNR performance reduction.

In Parts II and III of the thesis we have investigated several ways to incorporate various correlations between image/video high-frequency content to improve the quality of demosaicking and denoising.

We have first developed a novel adaptive filtering algorithm for demosaicking, which achieves better high-frequency preservation by using the similarity across color components. We have also developed a novel filter to obtain the green pixel luminance, and adopted an adaptive filtering interpolation for the red/blue pixel luminance. Experimental results show that the proposed approach can obtain better PSNR and visual performance than existing state-of-the-art approaches.

In addition, we have proposed three novel wavelet-based denoising approaches capable of exploiting inter- and intra-scale, inter-color and inter-frame correlations. WMCA approach works on smooth grayscale images, such as color-difference components, and achieves superior performance by embedding intra-scale correlations into a well-known min-cut optimization technique. The optimal color projection (OCP) approach uses inter-color similarity of detail wavelet coefficients to estimate an optimal color space for image denoising. The main advantage of this approach is that it can be combined with any existing grayscale denoising technique. Another developed denoising approach, called the vector estimator of wavelet coefficients (VEWC), can be used on both color images and videos. Experimental results show that all proposed methods can obtain better PSNR and edge preservation than the corresponding existing

state-of-the-art approaches.

While our work provides substantial contribution to understanding and improvement of DSC processing, the following is the list of interesting issues which are promising future work.

We have made several simplified assumptions in modeling the error of DSC processing. For example, we neglected the quadratic terms and the term that corresponds to cross-correlation between compression and demosaicking errors. Adding these to the analysis could yield a more general model and hence more accurate prediction of errors of DSC processing. Moreover, the developed compression model assumes the images to be stationary random fields, which is clearly only an approximation. It would be interesting to find out whether relaxing this assumption can lead to more accurate predictions.

Although we have mentioned several simple applications for the developed error inhomogeneity theory, our future work is to extend it to other potential applications. For example, treating the CFA image as grayscale and compressing it using JPEG2000 will yield a CFA image with RGB components of different qualities. Applying standard demosaicking techniques on such an image may not be optimal, and hence a balanced wavelet should be used for compression.

In all of the proposed denoising methods, we assumed i.i.d. noise. In reality, however, the noise may be correlated, which is often the case for medical images — speckle noise in ultrasound images [117], overlapping resonances in magnetic resonance spectroscopy (MRS) [118, 119], and metal artifact in X-ray computed tomography (CT) [120]. Another potential future work is to extend the proposed algorithms to images with arbitrary noise structures.

Appendix A

Error Inhomogeneity in Multi-Level Wavelet Decompositions

We show an example of the inhomogeneity analysis in 2-level wavelet decomposition. Since going from 1-level to 2-level decomposition is equivalent to applying the analysis filter on LL sub-band, the second level LL coefficients satisfy a relation similar to (3.1):

$$\begin{aligned}\hat{w}_0 &= \hat{g}_0 * w_{00} + \check{g}_1 * w_2 \\ \check{w}_0 &= \check{g}_0 * w_{00} + \hat{g}_1 * w_2\end{aligned}\tag{A.1}$$

where w_{00} and w_2 are the wavelet coefficients in the second-level low and high-frequency subbands, respectively. Plugging these into (3.2) will yield four different error expressions (as opposed to two in one-level decomposition), corre-

sponding to four locations of filter samples $i \in [0, 3]$, denoted by ${}^i f$.

$$\begin{aligned}
{}^0 e &= ({}^0 g_0 * \hat{g}_0 + {}^2 g_0 * \check{g}_0) * q_{00} + ({}^0 g_0 * \check{g}_1 + {}^2 g_0 * \hat{g}_1) * q_2 + \check{g}_1 * q_1 \\
{}^1 e &= ({}^1 g_0 * \hat{g}_0 + {}^3 g_0 * \check{g}_0) * q_{00} + ({}^1 g_0 * \check{g}_1 + {}^3 g_0 * \hat{g}_1) * q_2 + \hat{g}_1 * q_1 \\
{}^2 e &= ({}^2 g_0 * \hat{g}_0 + {}^0 g_0 * \check{g}_0) * q_{00} + ({}^2 g_0 * \check{g}_1 + {}^0 g_0 * \hat{g}_1) * q_2 + \check{g}_1 * q_1 \\
{}^3 e &= ({}^3 g_0 * \hat{g}_0 + {}^1 g_0 * \check{g}_0) * q_{00} + ({}^3 g_0 * \check{g}_1 + {}^1 g_0 * \hat{g}_1) * q_2 + \hat{g}_1 * q_1 \quad (\text{A.2})
\end{aligned}$$

where q_{00} and q_2 are the quantization errors of w_{00} and w_2 respectively. Let ${}^i \mathcal{G}_0 = {}^i g_0 * \hat{g}_0 + {}^{i+2} g_0 * \check{g}_0$ and ${}^i \mathcal{G}_1 = {}^i g_0 * \check{g}_1 + {}^{i+2} g_0 * \hat{g}_1$. Assuming the independence of q_{00} , q_2 and q_1 , we can substitute (3.4) with the following:

$$\begin{aligned}
\sigma_{0_e}^2 &= {}^0 \mathcal{G}_0 E\{q_{00,n}^T q_{00,n}\} {}^0 \mathcal{G}_0^T + {}^0 \mathcal{G}_1 E\{q_{2,n}^T q_{2,n}\} {}^0 \mathcal{G}_1^T + \check{g}_1 E\{q_{1,n}^T q_{1,n}\} \check{g}_1^T \\
\sigma_{1_e}^2 &= {}^1 \mathcal{G}_0 E\{q_{00,n}^T q_{00,n}\} {}^1 \mathcal{G}_0^T + {}^1 \mathcal{G}_1 E\{q_{2,n}^T q_{2,n}\} {}^1 \mathcal{G}_1^T + \hat{g}_1 E\{q_{1,n}^T q_{1,n}\} \hat{g}_1^T \\
\sigma_{2_e}^2 &= {}^2 \mathcal{G}_0 E\{q_{00,n}^T q_{00,n}\} {}^2 \mathcal{G}_0^T + {}^2 \mathcal{G}_1 E\{q_{2,n}^T q_{2,n}\} {}^2 \mathcal{G}_1^T + \check{g}_1 E\{q_{1,n}^T q_{1,n}\} \check{g}_1^T \\
\sigma_{3_e}^2 &= {}^3 \mathcal{G}_0 E\{q_{00,n}^T q_{00,n}\} {}^3 \mathcal{G}_0^T + {}^3 \mathcal{G}_1 E\{q_{2,n}^T q_{2,n}\} {}^3 \mathcal{G}_1^T + \hat{g}_1 E\{q_{1,n}^T q_{1,n}\} \hat{g}_1^T \quad (\text{A.3})
\end{aligned}$$

Let $\Sigma_{00} = E\{q_{00,n}^T q_{00,n}\}$ and $\Sigma_2 = E\{q_{2,n}^T q_{2,n}\}$. Then we have,

$$\begin{aligned}
\sigma_{0_e}^2 &= {}^0 \mathcal{G}_0 \Sigma_{00} {}^0 \mathcal{G}_0^T + {}^0 \mathcal{G}_1 \Sigma_2 {}^0 \mathcal{G}_1^T + \check{g}_1 \Sigma_1 \check{g}_1^T \\
\sigma_{1_e}^2 &= {}^1 \mathcal{G}_0 \Sigma_{00} {}^1 \mathcal{G}_0^T + {}^1 \mathcal{G}_1 \Sigma_2 {}^1 \mathcal{G}_1^T + \hat{g}_1 \Sigma_1 \hat{g}_1^T \\
\sigma_{2_e}^2 &= {}^2 \mathcal{G}_0 \Sigma_{00} {}^2 \mathcal{G}_0^T + {}^2 \mathcal{G}_1 \Sigma_2 {}^2 \mathcal{G}_1^T + \check{g}_1 \Sigma_1 \check{g}_1^T \\
\sigma_{3_e}^2 &= {}^3 \mathcal{G}_0 \Sigma_{00} {}^3 \mathcal{G}_0^T + {}^3 \mathcal{G}_1 \Sigma_2 {}^3 \mathcal{G}_1^T + \hat{g}_1 \Sigma_1 \hat{g}_1^T \quad (\text{A.4})
\end{aligned}$$

Due to the mirror symmetrical property of wavelet filter g_0 , $g_0(n) = g_0(-n)$, we can obtain ${}^1 g_0(n) = {}^3 g_0(-n)$, and then ${}^1 \mathcal{G}_i(n) = {}^3 \mathcal{G}_i(-n)$. Since $\Sigma_{00}^T = \Sigma_{00}$

Table A.1: PSNR performance (in dB) at pixel locations 0-3 (9-7 wavelet, SPIHT).

	0	1	2	3
0	34.72	33.70	34.23	33.65
1	33.66	33.01	33.40	32.97
2	34.32	33.49	33.85	33.46
3	33.59	32.92	33.34	32.93

and $\Sigma_2^T = \Sigma_2$, we have

$$\begin{aligned} {}^1\mathcal{G}_0\Sigma_{00} {}^1\mathcal{G}_0^T &= {}^3\mathcal{G}_0\Sigma_{00} {}^3\mathcal{G}_0^T \\ {}^1\mathcal{G}_1\Sigma_2 {}^1\mathcal{G}_1^T &= {}^3\mathcal{G}_1\Sigma_2 {}^3\mathcal{G}_1^T \end{aligned}$$

Hence,

$$\sigma_{1_e}^2 = \sigma_{3_e}^2 \tag{A.5}$$

This is confirmed in Table A.1, where PSNR difference at locations 1 and 3 is not larger than 0.09 dB, and can be ignored.

If the error variances at locations 0 and 2 would also be same, the 2-level analysis would lead to exactly the same conclusions as the one-level analysis. However, Table A.1 shows that PSNR difference at locations (0, 0) and (2, 2) is 0.87 dB, implying the significant different variances at locations 0 and 2, i.e. $\sigma_{0_e}^2 \neq \sigma_{2_e}^2$. In other words, the structure of error inhomogeneities will change slightly. The inhomogeneities between locations 0-1 and 3-0 will increase slightly, by approximately 0.4 dB according to experiments, while inhomogeneities between locations 1-2 and 2-3 will decrease slightly by the same amount.

The design of balanced wavelet can be straightforwardly extended to multi-level decomposition by designing different wavelet filter for each level.

Appendix B

Optimal Color Projection for Arbitrary Signal Covariance Matrix

In Section 6.3 we assumed perfect correlation between detail wavelet coefficients of RGB color components. In this appendix we explain how to relax this assumption to an arbitrary covariance matrix Σ_x . The direction that preserves the energy of the signal is given by the eigenvector of Σ_x corresponding to largest eigenvalue. Let $\mathbf{u}_0$ be such an eigenvector. It's easy to check that $\mathbf{u}_0$ is a generalization of the signal vector $\mathbf{s}$, since $\mathbf{u}_0 = \mathbf{s}$ when $\Sigma_x = \mathbf{s}\mathbf{s}^T$. New color-difference projection vectors, $\mathbf{d}_1$ and $\mathbf{d}_2$, are defined as being orthogonal to $\mathbf{u}_0$.

By normalizing the luminance projection vector $\mathbf{v}$ by $\mathbf{v}^T \mathbf{u}_0 = 1$ we satisfy the conditions of Lemma 5. The basis $(\mathbf{v}, \mathbf{d}_1, \mathbf{d}_2)$ so defined fixes the contribution of the luminance $x_v = \mathbf{v}^T \mathbf{x}$ in the inverse projection. The estimation error of

color differences is comparatively small and can be neglected. Hence, choosing the optimal $\mathbf{v}$ for the luminance projection leads to optimal denoising.

However, unlike in the perfect correlation case, the constraint $\mathbf{v}^T \mathbf{u}_0 = 1$ does not fix the variance of x_v . Hence, instead of minimizing the noise projection, we seek a projection that maximizes the signal-to-noise ratio (SNR). The variance of the signal and the noise projections onto a vector $\mathbf{v}$ are $E[(\mathbf{v}^T \mathbf{x})^2] = \mathbf{v}^T \Sigma_x \mathbf{v}$ and $E[(\mathbf{v}^T \mathbf{n})^2] = \mathbf{v}^T \Sigma_n \mathbf{v}$ respectively. The following Lemma shows how to obtain $\mathbf{v}$ that maximizes ratio $\frac{\mathbf{v}^T \Sigma_x \mathbf{v}}{\mathbf{v}^T \Sigma_n \mathbf{v}}$.

Lemma 7. *The optimal vector $\mathbf{v}$ that yields the largest SNR projection is the eigenvector of the matrix $\Sigma_n^{-1} \Sigma_x$ corresponding to the largest eigenvalue.*

Proof: Setting the derivative of $\frac{\mathbf{v}^T \Sigma_x \mathbf{v}}{\mathbf{v}^T \Sigma_n \mathbf{v}}$ to zero yields

$$\frac{d \left\{ \frac{\mathbf{v}^T \Sigma_x \mathbf{v}}{\mathbf{v}^T \Sigma_n \mathbf{v}} \right\}}{d\mathbf{v}} = 0 \quad \Rightarrow \quad \Sigma_n^{-1} \Sigma_x \mathbf{v} = \frac{\mathbf{v}^T \Sigma_x \mathbf{v}}{\mathbf{v}^T \Sigma_n \mathbf{v}} \mathbf{v}$$

Hence the necessary condition for $\mathbf{v}$ is to be an eigenvector of $\Sigma_n^{-1} \Sigma_x$. Let $\Sigma_n^{-1} \Sigma_x \mathbf{v} = \lambda \mathbf{v}$. Then

$$\frac{\mathbf{v}^T \Sigma_x \mathbf{v}}{\mathbf{v}^T \Sigma_n \mathbf{v}} = \frac{\mathbf{v}^T \Sigma_n \lambda \mathbf{v}}{\mathbf{v}^T \Sigma_n \mathbf{v}} = \lambda$$

Setting $\mathbf{v}$ to an eigenvector corresponding to the largest eigenvalue maximizes $\frac{\mathbf{v}^T \Sigma_x \mathbf{v}}{\mathbf{v}^T \Sigma_n \mathbf{v}}$ and hence the SNR. ■

It is easy to check that vector $\mathbf{v}$ obtained using Lemma 7 coincides with the one obtained by Lemma 6 when $\Sigma_x = \mathbf{s} \mathbf{s}^T$.

Signal covariance matrix varies across subbands and scales. Hence one can potentially use a different color projection for each subband and scale of the wavelet tree. However, the main problem here is to estimate the noise and

Table B.1: CPSNR performance (in dB) using the perfect correlation assumption versus arbitrary signal covariance matrix. The covariance matrix of the noise is assumed known.

	Uncorrelated noise		Correlated noise	
	Uniform	Nonuniform	Uniform	Nonuniform
Assumption	29.19	29.89	28.28	29.51
Arbitrary	29.23	29.92	28.29	29.49

signal covariance matrices simultaneously. If the noise covariance matrix is known beforehand, the signal covariance matrix can be estimated using

$$\hat{\Sigma}_x = E(\tilde{\mathbf{x}}\tilde{\mathbf{x}}^T) - \Sigma_n \quad (\text{B.1})$$

However, even if the noise covariance matrix is known, applying scale dependent projections leads to only modest improvements in the denoising performance, as shown in Table B.1. The improvements in CPSNR performance don't exceed 0.04 dB and don't justify the increased computational complexity.

Appendix C

Estimation Error of VEWC and OCP Denoising Approaches

Lemma 8. *Assuming correct classification of wavelet coefficients and perfectly correlated inter-color correlations, the error of vector MMSE estimate (7.1) used in VEWC is equal to that of OCP-based (6.7) scalar estimate (1.7), i.e.,*

$$\varepsilon_{VEWC}^2 = \varepsilon_{OCP}^2 \quad (\text{C.1})$$

Proof: Based on the vector MMSE estimate (7.1) and the correct classification assumption, the estimation error variance in state $q \in \{\mathcal{S}, \mathcal{L}\}$ is

$$\begin{aligned} \varepsilon_{VEWC}^2 &= E \left\{ [\mathbf{x} - E(\tilde{\mathbf{x}}|q)]^T [\mathbf{x} - E(\tilde{\mathbf{x}}|q)] \right\} \\ &= Tr \left(E \left\{ [\mathbf{x} - E(\tilde{\mathbf{x}}|q)] [\mathbf{x} - E(\tilde{\mathbf{x}}|q)]^T \right\} \right) \end{aligned}$$

where Tr is the trace of the matrix. Let $\Upsilon = E \left\{ [\mathbf{x} - E(\tilde{\mathbf{x}}|q)] [\mathbf{x} - E(\tilde{\mathbf{x}}|q)]^T \right\}$.

Since $\Sigma_{x|q}^T = \Sigma_{x|q}$ and $\Sigma_n^T = \Sigma_n$, we have

$$\begin{aligned}\Upsilon &= E \left\{ \left[\mathbf{x} - \Sigma_{x|q}(\Sigma_{x|q} + \Sigma_n)^{-1}(\mathbf{x} + \mathbf{n}) \right] \left[\mathbf{x} - \Sigma_{x|q}(\Sigma_{x|q} + \Sigma_n)^{-1}(\mathbf{x} + \mathbf{n}) \right]^T \right\} \\ &= \Sigma_{x|q}(\Sigma_{x|q} + \Sigma_n)^{-1}\Sigma_n - \Sigma_{x|q}(\Sigma_{x|q} + \Sigma_n)^{-1}\Sigma_{x|q}(\Sigma_{x|q} + \Sigma_n)^{-1}\Sigma_n \\ &\quad + \Sigma_{x|q}(\Sigma_{x|q} + \Sigma_n)^{-1}\Sigma_n(\Sigma_{x|q} + \Sigma_n)^{-1}\Sigma_{x|q}\end{aligned}$$

It can be shown that $\Sigma_{x|q}(\Sigma_{x|q} + \Sigma_n)^{-1}\Sigma_n = \Sigma_n(\Sigma_{x|q} + \Sigma_n)^{-1}\Sigma_{x|q}$. Hence,

$\Upsilon = \Sigma_{x|q}(\Sigma_{x|q} + \Sigma_n)^{-1}\Sigma_n$, which yields

$$\varepsilon_{VEWC}^2 = Tr \left[\Sigma_{x|q}(\Sigma_{x|q} + \Sigma_n)^{-1}\Sigma_n \right] \quad (C.2)$$

According to (6.2), the perfect correlation in $\mathbf{x}$ leads to $\Sigma_{x|q} = \mathbf{s}\mathbf{s}^T$, where $\mathbf{s}$ is the signal vector in state q . Then (C.2) becomes

$$\begin{aligned}\varepsilon_{VEWC}^2 &= Tr \left[\mathbf{s}\mathbf{s}^T(\mathbf{s}\mathbf{s}^T + \Sigma_n)^{-1}\Sigma_n \right] \\ &= \frac{\mathbf{s}^T \mathbf{s}}{1 + \mathbf{s}^T \Sigma_n^{-1} \mathbf{s}}\end{aligned} \quad (C.3)$$

The perfect correlation in $\mathbf{x}$ also leads to zero signal left in color-difference image $x_{d_i} = 0$ after OCP transformation (6.5). Then the resultant estimation error only depends on denoising the optimally projected luminance $\tilde{x}_v = \mathbf{v}_{opt}^T \tilde{\mathbf{x}}$. Let ε_{opt}^2 be the estimation error after denoising $\tilde{x}_v$. The inverse OCP transformation (6.6) yields,

$$\varepsilon_{OCP}^2 = \mathbf{s}^T \mathbf{s} \varepsilon_{opt}^2 \quad (C.4)$$

According to Lemma 4 in Chapter 6 (Page 143), the signal variance in $\tilde{x}_v$ is fixed, i.e. $\sigma_{x_v}^2 = 1$. The optimal projection vector $\mathbf{v}_{opt}$ in (6.7) makes the noise

variance in $\tilde{x}_v$ equal to

$$\sigma_{n_v}^2 = E \left\{ (\mathbf{v}_{opt}^T \mathbf{n})^2 \right\} = \frac{1}{\mathbf{s}^T \boldsymbol{\Sigma}_n^{-1} \mathbf{s}} \quad (\text{C.5})$$

According to the error of scalar MMSE estimate in (6.1), we have

$$\varepsilon_{opt}^2 = \frac{\frac{1}{\mathbf{s}^T \boldsymbol{\Sigma}_n^{-1} \mathbf{s}}}{1 + \frac{1}{\mathbf{s}^T \boldsymbol{\Sigma}_n^{-1} \mathbf{s}}} = \frac{1}{1 + \mathbf{s}^T \boldsymbol{\Sigma}_n^{-1} \mathbf{s}} \quad (\text{C.6})$$

Combining (C.6) and (C.4) yields

$$\varepsilon_{OCP}^2 = \frac{\mathbf{s}^T \mathbf{s}}{1 + \mathbf{s}^T \boldsymbol{\Sigma}_n^{-1} \mathbf{s}} \quad (\text{C.7})$$

which is exactly the same error obtained by vector MMSE estimate in (C.3). ■

Publications

- **Journal papers**

1. Nai-Xiang Lian, Vitali Zagorodnov, and Yap-Peng Tan, "Error Inhomogeneity of Wavelet Based Compression", *accepted for publication in IEEE Trans. on Signal Processing*, Oct 2006.
2. Nai-Xiang Lian, Lanlan Chang, Vitali Zagorodnov, and Yap-Peng Tan, "Reversing Demosaicking and Compression in Color Filter Array Image Processing: Modeling and Analysis", *IEEE Trans. on Image Processing*, vol.15, no.11, pp.3261-3278, Nov 2006.
3. Nai-Xiang Lian, Vitali Zagorodnov, and Yap-Peng Tan, "Edge-Preserving Image Denoising via Optimal Color Space Projection", *IEEE Trans. on Image Processing*, vol 15, no 9, pp 2575-2587, Sept 2006.
4. Nai-Xiang Lian, Vitali Zagorodnov, and Yap-Peng Tan, "Color Image Denoising Using Wavelets and Minimum Cut Analysis", *IEEE Signal Processing Letters*, vol.12, no.11, pp.741-744, Nov 2005.
5. Nai-Xiang Lian, Lanlan Chang, Yap-Peng Tan, and Vitali Zagorodnov, "Adaptive Filtering for Demosaicking Color Filter Array Images", *revised and resubmitted to IEEE Trans. on Image Processing*, Dec 2006.
6. Nai-Xiang Lian, Vitali Zagorodnov, and Yap-Peng Tan, "Retinal vessel detection by self-matched filtering", *submitted to IEEE Trans. on Image Processing*, Nov 2006.
7. Nai-Xiang Lian, Vitali Zagorodnov, and Yap-Peng Tan, "Video Denoising Using Vector Estimation of Wavelet Coefficients", *submitted to IEEE Trans. on Circuit System for Video Technology*, Aug 2005.

- **Conference papers**

1. Nai-Xiang Lian, Vitali Zagorodnov, and Yap-Peng Tan, "Error Inhomogeneity of Wavelet Image Compression", *IEEE International Conference on Image Processing (ICIP 2006)*, Oct 2006.
2. Nai-Xiang Lian, Lanlan Chang, Vitali Zagorodnov, and Yap-Peng Tan, "Reversing Demosaicking and Compression in Color Filter Array Images", *IEEE International Conference on Image Processing (ICIP 2006)*, Oct 2006.
3. Nai-Xiang Lian, Vitali Zagorodnov, and Yap-Peng Tan, "Video Denoising Using Vector Estimation of Wavelet Coefficients", *IEEE International Symposium on Circuits and Systems (ISCAS 2006)*, May 2006.
4. Nai-Xiang Lian, Vitali Zagorodnov, and Yap-Peng Tan, "Image Denoising Using Optimal Color Space Projection", *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2006)*, May 2006.
5. Nai-Xiang Lian, Lanlan Chang, and Yap-Peng Tan, "Improved Color Filter Array Demosaicking By Accurate Luminance Estimation", *IEEE International Conference on Image Processing (ICIP 2005)*, Sept 2005.
6. Nai-Xiang Lian and Yap-Peng Tan, "Probabilistic Approach to K -Nearest Neighbor Video Retrieval", *IEEE International Symposium on Circuits and Systems (ISCAS 2004)*, May 2004.
7. Nai-Xiang Lian, Yap-Peng Tan, and Kap Luk Chan, "Efficient Video Retrieval Using Shot Clustering and Alignment", *International Conference on Information, Communications & Signal Processing and Pacific-Rim Conference on Multimedia (ICICS-PCM 2003)*, Dec 2003.

Bibliography

- [1] J. Hamilton and J. Adams, “Adaptive color plane interpolation in single sensor color electronic camera,” *United States Patent*, no. 5,629,734, 1997.
- [2] B. K. Gunturk, Y. Altunbasak, and R. M. Mersereau, “Color plane interpolation using alternating projections,” *IEEE Transactions on Image Processing*, vol. 11, no. 9, pp. 997–1013, 2002.
- [3] M. Lang, H. Guo, J. E. Odegard, C. S. Burrus, and R. O. Wells, “Noise reduction using an undecimated discrete wavelet transform,” *IEEE Signal Processing Letters*, vol. 3, no. 1, pp. 10–12, Jan. 1996.
- [4] J. Romberg, H. Choi, and R. G. Baraniuk, “Bayesian tree-structured image modeling using wavelet-domain hidden markov models,” *IEEE Transactions on Image Processing*, vol. 10, no. 7, pp. 1056–1068, July 2001.
- [5] C. A. Poynton, *A technical introduction to digital video*. John Wiley & Sons, Inc., 1996.
- [6] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*. Pearson Education, Inc., 2002.
- [7] D. Alleysson, S. Susstrunk, and J. Herault, “Linear demosaicking inspired by human visual system,” *IEEE Transactions on Image Processing*, vol. 14, no. 4, pp. 439–449, 2005.
- [8] M. Mancuso and S. Battiato, “An introduction to the digital still camera technology,” *ST Journal of System Research*, vol. 2, December 2001.
- [9] B. Bayer, “Color imaging array,” *United States Patent*, no. 3,971,065, 1976.
- [10] J. Adams, “Interactions between color plane interpolation and other image processing functions in electronic photography,” *Proc. SPIE Cameras*

and *Systems for Electronic Photography and Scientific Imaging*, vol. 2416, pp. 144–151, 1995.

- [11] E. Chang, S. Cheung, and D. Pan, “Color filter array recovery using a threshold-based variable number of gradients,” *Proc. of SPIE on Sensors, Cameras, and Applications*, vol. 3650, pp. 36–43, 1999.
- [12] D. Cok, “Signal processing method and apparatus for producing interpolated chrominance values in a sampled color image signal,” *United States Patent*, no. 4,642,678, 1987.
- [13] C. Laroche and M. Prescott, “Apparatus and method for adaptively interpolating a full color image utilizing chrominance gradients,” *United States Patent*, no. 5,373,322, 1994.
- [14] R. Kimmel, “Demosaiicing: image reconstruction from color CCD samples,” *IEEE Trans. Image Processing*, vol. 7, no. 3, pp. 1221–1228, 1999.
- [15] T. W. Freeman, “Median filter for reconstructing missing color samples,” *United States Patent*, no. 4,724,395, 1988.
- [16] S. C. Pei and I. K. Tam, “Effective color interpolation in CCD color filter array using signal correlation,” *IEEE Int. Conf. Image Processing*, vol. 3, 2000.
- [17] L. Chang and Y.-P. Tan, “Effective use of spatial and spectral correlations for color filter array demosaicking,” *IEEE Trans. on Consumer Electronics*, vol. 50, pp. 355–365, Feb. 2004.
- [18] K. Parulski and K. E. Spaulding, “Color image processing for digital cameras,” *Digital Color Imaging Handbook*, edited by G. Sharma. CRC Press, Boca Raton - FL, pp. 728–757, 2002.
- [19] R. Ramanath, W. E. Snyder, Y. Yoo, and M. S. Drew, “Color image processing pipeline,” *IEEE Signal Processing Magazine*, vol. 22, no. 1, pp. 34–43, Jan. 2005.
- [20] International Electrotechnical Commission, *Colour Measurement and Management in Multimedia Systems and Equipment - Part 2-1: Default RGB Colour Space - sRGB*. IEC 61966-2-1, 1999.
- [21] Y. Tsai, “Color image compression for single-chip cameras,” *IEEE Transaction on Electron Devices*, vol. 38, no. 5, pp. 1226–1232, May 1991.

- [22] T. Toi and M. Ohita, "A subband coding technique for image processing in single ccd cameras with bayer color filter arrays," *IEEE Trans. on Consumer Electronics*, vol. 45, no. 1, pp. 176–180, Feb. 1999.
- [23] N. Zhang and X. Wu, "Lossless compression of color mosaic images," *IEEE Int. Conf. Image Processing*, vol. 1, pp. 517 – 520, 2004.
- [24] S.-Y. Lee and A. Ortega, "A novel approach of image compression in digital cameras with a bayer color," *IEEE Int. Conf. Image Processing*, vol. 3, pp. 482–485, 2001.
- [25] B. Parrein, M. Tarin, and P. Horain, "Demosaicking and JPEG2000 compression of microscopy images," *IEEE Int. Conf. Image Processing*, vol. 1, pp. 521–524, 2004.
- [26] C. C. Koh, J. Mukherjee, and S. K. Mitra, "New efficient methods of image compression in digital cameras with color filter array," *IEEE Trans. on Consumer Electronics*, vol. 49, no. 4, pp. 1448–1456, Nov. 2003.
- [27] H. Fujita, K. Doi, and L. E. Fencil, "Image feature analysis and computer-aided diagnosis in digital radiography. ii. computerized determination of vessel sizes in digital subtraction angiography," *Medical Physics*, vol. 14, p. 549556, July/Aug. 1987.
- [28] V. Bhaskaran and K. Konstantinides, *Image and video compression standards*. Kluwer Academic Publishers, 1995.
- [29] J. M. Shapiro, "Embedded image coding using zerotrees of wavelet coefficients," *IEEE Trans. on Signal Processing*, vol. 41, no. 12, pp. 3445–3462, Dec. 1993.
- [30] A. Said and W. A. Pearlman, "New, fast and efficient image codec based on set partitioning in hierarchical trees," *IEEE Trans. on Circuits and System for Video Technology*, vol. 6, no. 3, pp. 243–249, June. 1996.
- [31] C. Chrysafis and A. Ortega, "Efficient context-based entropy coding for lossy wavelet image compression," *Processings of 1997 Data Compression Conference*, pp. 241–250, Mar. 1997.
- [32] Z. Xiong, K. Ramchandran, and M. Orchard, "Space-frequency quantization for wavelet image coding," *IEEE Trans. on Image Processing*, vol. 6, no. 5, pp. 677–693, Mar. 1997.

- [33] D. Taubman, "High performance scalable image compression with ebcot," *IEEE Trans. on image Processing*, vol. 9, no. 7, pp. 1158–1170, July. 1997.
- [34] X. Wu, "High-order context modeling and embedded conditional entropy coding of wavelet coefficients for image compression," *Processings of 31st Asilomar Conference on Singals, Systems, and Computers*, pp. 1378–1382, Nov. 1997.
- [35] X. Wu, "Context quantization with fisher discriminant for adaptive embedded wavelet image coding," *Processings of 1999 Data Compression Conference*, pp. 102–111, Mar. 1999.
- [36] E. S. Hong and R. E. Ladner, "Group testing for image compression," *Processings of 2000 Data Compression Conference*, pp. 3–12, Mar. 2000.
- [37] K. Peng and J. Kieffer, "Embedded image compression based on wavelet pixel classification and sorting," *IEEE Trans. on image Processing*, vol. 13, no. 8, pp. 1011–1017, Aug. 2004.
- [38] D. S. Taubman and M. W. Marcellin, "JPEG2000: Standard for interactive imaging," *Processings of the IEEE*, vol. 90, no. 8, pp. 1336–1357, Aug. 2002.
- [39] J. Katto and Y. Yasuda, "Performance evaluation of subband coding and optimization," *Processing SPIE Visual Communication and Image Processing*, vol. 1605, pp. 95–106, 1991.
- [40] M. Lightstone, E. Majani, and S. K. Mitra, "Low bit-rate design considerations for wavelet-based image coding," *Multidimensional Systems and Signal Processing*, vol. 8, no. 2, pp. 111–128, 1997.
- [41] D. B. H. Tay, "Rationalizing the coefficients of popular biorthogonal wavelet filters," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 10, pp. 998–1005, 2000.
- [42] N. S. Jayant and P. Noll, *Digital coding of waveforms*. Prentice-Hall Press, 1984.
- [43] R. M. Gray and D. L. Neuhoff, "Quantization," *IEEE Transaction on Information Theory*, vol. 44, no. 6, pp. 2325–2382, Oct. 1998.
- [44] O. Egger, P. Fleury, T. Ebrahimi, and M. Kunt, "High-performance compression of visual information: a tutorial review – part I: Still pictures," *Proceedings of the IEEE*, vol. 87, no. 6, pp. 976–1011, Jun. 1999.

- [45] T. D. Tran and T. Q. Nguyen, "A progressive transmission image coder using linear phase uniform filterbanks as block transforms," *IEEE Transaction on Image Processing*, vol. 8, no. 11, pp. 1493–1507, Nov. 1999.
- [46] T. D. Tran, R. L. de Queiroz, and T. Q. Nguyen, "Linear-phase perfect reconstruction filter bank: Lattice structure, design, and application in image coding," *IEEE Transaction on Signal Processing*, vol. 48, no. 1, pp. 133–147, Jan. 2000.
- [47] J. Liang and T. D. Tran, "Fast multiplierless approximations of the dct with the lifting scheme," *IEEE Transaction on Signal Processing*, vol. 49, no. 12, pp. 3032–3044, Dec. 2001.
- [48] T. Sakamoto, C. Nakanishi, and T. Hase, "Software pixel interpolation for digital still cameras suitable for a 32-bit MCU," *IEEE Transaction on Consumer Electronics*, vol. 44, no. 4, pp. 1342–1352, Nov. 1998.
- [49] X. Wu and N. Zhang, "Primary-consistent soft-decision color demosaicking for digital cameras," *IEEE Trans. on Image Processing*, vol. 13, pp. 1263–1274, 2004.
- [50] K. Hirakawa and T. W. Parks, "Adaptive homogeneity-directed demosaicking algorithm," *IEEE Trans. on Image Processing*, vol. 14, no. 3, pp. 360–369, Mar. 2005.
- [51] A. Jain, *Fundamentals of Digital Image Processing*. Prentice-Hall Press, 1989.
- [52] J. S. Lim, *Two-Dimensional Signal and Image Processing*. Prentice-Hall PTR.
- [53] D. L. Donoho and I. M. Johnstone, "Ideal spatial adaptation by wavelet shrinkage," *Biometrika*, vol. 81, pp. 425–455, 1994.
- [54] D. L. Donoho and I. M. Johnstone, "Adapting to unknown smoothness via wavelet shrinkage," *J. Amer. Statist. Assoc.*, vol. 90, pp. 1200–1224, 1995.
- [55] M. S. Crouse, R. D. Nowak, and R. G. Baraniuk, "Wavelet-based statistical signal processing using hidden markov models," *IEEE Trans. on Signal Processing*, vol. 46, no. 4, pp. 886–902, Apr. 1998.

- [56] M. Malfait and D. Roose, "Wavelet-based image denoising using a markov random field a prior model," *IEEE Trans. on Image Processing*, vol. 4, pp. 549–565, Apr. 1997.
- [57] A. Pizurica, W. Philips, I. Lemahieu, and M. Acheroy, "A joint inter- and intrascale statistical model for bayesian wavelet based image denoising," *IEEE trans. on Image Processing*, vol. 11, no. 5, pp. 545–556, May 2002.
- [58] J. Scharcanski, C. R. Jung, and R. T. Clarke, "Adaptive image denoising using scale and space consistency," *IEEE trans. on Image Processing*, vol. 11, no. 9, pp. 1092–1101, Sept. 2002.
- [59] G. Y. Chen, T. D. Bui, and A. Krzyzak, "Image denoising using neighbouring wavelet coefficients," *IEEE ICASSP 2004*, no. 2, pp. 917–920, 2004.
- [60] P. Bao and L. Zhang, "Noise reduction for magnetic resonance images via adaptive multiscale products thresholding," *IEEE Trans. on Medical Imaging*, vol. 22, no. 9, pp. 1089–1099, Sept. 2003.
- [61] J. Portilla, V. Strela, M. J. Wainwright, and E. P. Simoncelli, "Image denoising using scale mixtures of gaussians in the wavelet domain," *IEEE Trans. on Image Processing*, vol. 12, no. 11, pp. 1338–1351, Nov. 2003.
- [62] J. Liu and P. Moulin, "Information-theoretic analysis of interscale and intrascale dependencies between image wavelet coefficients," *IEEE Trans. on Image Processing*, vol. 10, no. 11, pp. 1647–1658, Nov. 2001.
- [63] I. Daubechies, "Orthonormal bases of compactly supported wavelets," *Communications on Pure and Applied Mathematics*, vol. 16, pp. 909–996, Jan. 1988.
- [64] Y. Hel-Or, "The canonical correlations of color images and their use for demosaicing," *Technical Report, HP Labs, HPL-2003-164R1*, Feb. 2004.
- [65] D. D. Muresan and T. P. Parks, "Demosaicking using optimal recovery," *IEEE Trans. on Image Processing*, vol. 14, no. 2, pp. 267–278, Feb. 2005.
- [66] P. Scheunders, "Wavelet thresholding of multivalued images," *IEEE Trans. on Image Processing*, vol. 13, no. 4, pp. 475–483, Apr. 2004.
- [67] A. Pizurica and W. Philips, "Estimating probability of presence of a signal of interest in multiresolution single- and multiband image denoising," *IEEE Trans. on Image Processing*, vol. 15, no. 3, pp. 654– 665, Mar. 2006.

- [68] P. Scheunders and J. Driesen, "Least-squares interband denoising of color and multispectral images," *IEEE Int. Conf. Image Processing*, Singapore, Oct 2004.
- [69] O. Ben-Shahar and S. W. Zucher, "Hue fields and color curvatures: a perceptual organization approach to color image denoising," *IEEE conference on CVPR*, June 2003.
- [70] T. F. Chan, S. H. Kang, and J. Shen, "Total variation denoising and enhancement of color images based on the cb and hsv color models," *Journal of Visual Communication and Image Representation*, vol. 12, pp. 422–435, June 2001.
- [71] B. Tang and V. Caselles, "Color image enhancement via chromaticity diffusion," *IEEE Trans. on Image Processing*, vol. 10, no. 5, pp. 701–707, May 2001.
- [72] S. Kim and S. H. Kang, "Implicit procedures for pde-based color image denoising via brightness-chromaticity decomposition," *Technical Report, Department of Mathematics, University of Kentucky, Lexington, KY*, July 2002.
- [73] L. Zhang, X. Wu, and P. Bao, "Real-time lossless compression for mosaic video sequences," *Real-Time Imaging*, vol. 11, no. 5-6, pp. 370–377, Oct.-Dec. 2005.
- [74] B. Usevitch, "Context based entropy coding of block transform coefficients for image compression," *Processings of 1996 Data Compression Conference*, pp. 387–395, Mar. 1996.
- [75] M. Vetterli and J. Kovacevic, *Wavelet and Subband Coding*. New Jersey: Prentice-Hall, 1995.
- [76] J. D. Villasenor, B. Belzer, and J. Liao, "Wavelet filter evaluation for image compression," *IEEE Trans. on image Processing*, vol. 4, no. 8, pp. 1053–1060, Aug. 1995.
- [77] P. P. Vaidyanathan, *Multirate Systems and Filter Banks*. New Jersey: Prentice-Hall, 1993.
- [78] O. Rioul, "Simple regularity criteria for subdivision schemes," *SIAM J. Math Anal*, vol. 23, pp. 1544–1576, 1992.

- [79] S. C. Pei and I. K. Tam, "Effective color interpolation in ccd color filter arrays using signal correlation," *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 13, no. 6, pp. 503–513, June 2003.
- [80] X. Li and M. T. Orchard, "New edge-directed interpolation," *IEEE Transactions on Image Processing*, vol. 10, no. 10, pp. 997–1013, 2001.
- [81] J. W. Glotzbach and R. W. Schafer, "A method of color filter array interpolation with alias cancellation properties," *Proc. Int'l Conf. on Image Processing, (ICIP'01), Thessaloniki, Greece*, vol. 1, pp. 141–144, Oct.2001.
- [82] B. K. Gunturk, J. W. Glotzbach, Y. Altunbasak, R. W. Schafer, and R. M. Mersereau, "Demosaicking: Color filter array interpolation," *IEEE Signal Processing Magazine*, vol. 22, no. 1, pp. 44–54, Jan. 2005.
- [83] X. Li, "Demosaicking by successive approximation," *IEEE Transactions on Image Processing*, vol. 14, no. 3, pp. 370–379, 2005.
- [84] F.W.Campbell and J. Kulikowski, "Orientational selectivity of the human visual system," *Journal of Physiology*, vol. 187, no. 2, pp. 437–445, 1966.
- [85] M. J. McMahon and D. I. A. MacLeod, "The origin of the oblique effect examined with pattern adaptation and masking," *Journal of Vision*, vol. 3, no. 3, pp. 230–239, April 17, 2003.
- [86] M. Vetterli and J. Kovacevic, *Wavelets and Subband Coding*. Prentice-Hall PTR, 1995.
- [87] D. Alleysson, S. Susstrunk, and J. Herault, "Supplementary material for linear demosaicking inspired by human visual system," http://ivrgwww.epfl.ch/index.php?name=IEEE_Demosaicking.
- [88] Y. B. Boykov and V. Kolmogorov, "An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision," *IEEE Trans. on PAMI*, vol. 26, no. 9, pp. 1124–1137, Sept. 2004.
- [89] D. Greig, B. Porteous, and A. Seheult, "Exact maximum a posteriori estimation for binary images," *J. Royal Statistical Soc., Series B*, vol. 51, no. 2, pp. 271–279, 1989.
- [90] C. A. Bouman, "Markov random fields and stochastic image models," *Tutorial Presented at IEEE International Conference on Image Processing*, Oct. 1995.

- [91] V. Kolmogorov and R. Zabini, "What energy functions can be minimized via graph cuts," *IEEE Trans. on Pattern Anal. Machine Intell.*, vol. 26, no. 2, pp. 147–159, Feb. 2004.
- [92] X. Zhang and B. A. Wandell, "Color image fidelity metrics evaluated using image distortion maps," *Signal Processing*, vol. 70, 1998.
- [93] D. P. Bertsekas, *Constrained optimization and Lagrange multiplier methods*. New York, Academic Press, 1982.
- [94] B. Widrow and S. Stearns, *Adaptive signal processing*. New Jersey, Prentice Hall, 1990.
- [95] G. C. Holst, *CCD Arrays, cameras, and displays*. JCD Publishing and SPIE Optical Engineering Press, 1998.
- [96] D. A. Harville, *Matrix algebra from a statistician's perspective*. New York, Springer, 1997.
- [97] J. Canny, "A computational approach to edge detection," *IEEE Transactions on PAMI*, vol. 8, no. 6, 1986.
- [98] J. C. Brailean, R. P. Kleihorst, S. Efstratiadis, A. K. Katsaggelos, and R. L. Lagendijk, "Noise reduction filters for dynamic image sequences: a review," *Proceedings of the IEEE*, vol. 83, no. 9, pp. 1272–1292, Sept. 1995.
- [99] D. L. Wilson, K. N. Jabri, and R. Aufrichtig, "Perception of temporally filtered x-ray fluoroscopy images," *IEEE Trans. on Medical Imaging*, vol. 18, no. 1, pp. 22–31, Jan 1999.
- [100] R. Dugad and N. Ahuja, "Video denoising by combining kalman and wiener estimates," *IEEE International Conference on Image Processing*, vol. 4, pp. 153–156, Oct. 1999.
- [101] F. Dekeyser, P. Bouthemy, and P. Perez, "Spatio-temporal wiener filtering of image sequences using a parametric motion model," *IEEE International Conference on Image Processing*, vol. 1, pp. 208–211, Sept. 2000.
- [102] A. C. Kokaram and S. J. Godsill, "Mcmc for joint noise reduction and missing data treatment in degraded video," *IEEE Trans. on Signal Processing*, vol. 50, no. 2, pp. 189–205, Feb. 2002.

- [103] A. C. Kokaram, "On missing data treatment for degraded video and film archives: A survey and a new bayesian approach," *IEEE Trans. on Image Processing*, vol. 13, no. 3, pp. 397–415, Mar. 2004.
- [104] K. J. Boo and N. K. Bose, "A motion-compensated spatio-temporal filter for image sequences with signal-dependent noise images," *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 8, no. 3, pp. 287–298, June 1999.
- [105] R. Rajagopalan and M. T. Orchard, "Synthesizing processed video by filtering temporal relationships," *IEEE Trans. on Image Processing*, vol. 11, no. 1, pp. 26–36, Jan. 2002.
- [106] A. Pizurica, V. Zlokolica, and W. Philips, "Combined wavelet domain and temporal video denoising," *IEEE Conference on Advanced Video and Signal Based Surveillance*, pp. 334 – 341, July 2003.
- [107] I. W. Selesnick and K. Y. Li, "Video denoising using 2d and 3d dual-tree complex wavelet transforms," *Wavelet Applications in Signal and Image Processing X (Proc. SPIE 5207)*, pp. 334 – 341, San Diego, August, 2003.
- [108] F. Shi and I. W. Selesnick, "Video denoising using oriented complex wavelet transforms," *IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 2, pp. 949–952, May 2004.
- [109] N. Rajpoot, Z. Yao, and R. Wilson, "Adaptive wavelet restoration of noisy video sequences," *IEEE International Conference on Image Processing*, vol. 2, Oct. 2004.
- [110] M. K. Mihcak, I. Kozintsev, K. Ramchandran, and P. Moulin, "Low complexity image denoising based on statistical modeling of wavelet coefficients," *IEEE Signal Processing Letters*, vol. 6, no. 12, p. 300303, Dec. 1999.
- [111] H. V. Poor, *An Introduction to Signal Detection and Estimation: second Edition*. Springer, 1998.
- [112] L. Rabiner, "A tutorial on hidden markov models and selected applications in speech recognition," *Proceedings of the IEEE*, vol. 77, no. 2, pp. 257–286, Feb. 1989.
- [113] C. Stiller and J. Konrad, "Estimating motion in image sequences," *IEEE Signal Processing Magazine*, vol. 16, no. 4, pp. 70–91, July 1999.

- [114] B. Reddy and B. N. Chatterji, "An fft-based technique for translation, rotation, and scale-invariance image registration," *IEEE Trans. on Image Processing*, vol. 5, no. 8, pp. 1266–1271, Aug. 1996.
- [115] R. J. Radke, S. Andra, O. Al-Kofahi, and B. Roysam, "Image change detection algorithm: a systematic survey," *IEEE Trans. on Image Processing*, vol. 14, no. 3, pp. 294–307, Mar. 2005.
- [116] W. Pratt, *Digital image processing: second Edition*. John Wiley & Sons, Inc., 1991.
- [117] J. G. Abbott and F. L. Thurstone, "Acoustic speckle: theory and experimental analysis," *Ultrason. Imag.*, vol. 1, pp. 303–324, 1979.
- [118] A. Macovski, "Noise in mri," *Magn. Reson. Med.*, vol. 36, pp. 494–497, 1996.
- [119] H. Gudbjartsson and S. Patz, "The rician distribution of noisy mri data," *Magn. Reson. Med.*, vol. 34, pp. 910–914, 1995.
- [120] G. H. Glover and N. J. Pelc, "An algorithm for the reduction of metal clip artifact in ct reconstruction," *Med. Phys.*, vol. 8, pp. 799–807, 1981.