Error analysis and suppression in visual signal processing

Chang, Lanlan

2008

Chang, L. (2008). Error analysis and suppression in visual signal processing. Doctoral thesis, Nanyang Technological University, Singapore.

<https://hdl.handle.net/10356/40463>

<https://doi.org/10.32657/10356/40463>

Error Analysis and Suppression in Visual Signal Processing

Chang Lanlan

School of Electrical and Electronic Engineering

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

2008

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

.....

Lanlan Chang

Summary

Error suppression is important for visual signal processing in producing high-quality images and videos. Due to various sources of errors, different suppression techniques are required to meet specific application requirements. The main objective of error analysis is to investigate the principle of error generation, and then incorporate suitable suppression techniques into the design of individual processing steps as well as whole processing chain. An alternative methodology is to analyze the distinct characteristics of the resultant errors, and devise effective removal methods accordingly. This thesis addresses error analysis and suppression in three applications – error suppression in color filter array (CFA) demosaicking, error analysis in digital still camera (DSC) image processing chain, and error detection and removal for composite video frames.

Firstly, we study the error suppression techniques for CFA demosaicking, which aims to recover full-color images from CFA samples. From spatial domain analysis, CFA demosaicking errors (artifacts) are caused due to isotropic interpolation or inconsistency between different color components. To suppress the demosaicking artifacts, we propose a method based on the adaptive use of spatial and spectral correlations in color images. Specifically, the proposed method performs weighted interpolation in color-difference components to adaptively estimate high-frequency information across color components. Furthermore, we demonstrate that existing CFA demosaicking methods can be enhanced by making better

use of spatial correlation. The thesis also investigates the CFA demosaicking problem from the view of Fourier domain analysis. From this perspective, demosaicking errors are caused by improper separation between aliased image spectra. Based on the specific characteristics of CFA spectra, we design a demosaicking method which first extracts the luminance component with adaptive filtering techniques and then recovers the color components based on the high correlation between color components. Experimental results confirm that the proposed methods outperform the existing methods both visually and in terms of peak signal-to-noise ratio (PSNR) measure.

Secondly, we analyze the error suppression in the DSC image processing chain. Several researchers have proposed to reverse the order of demosaicking and its following operation, such as compression or sharpening, in DSC and demonstrated that the alternative chain could outperform the conventional one in particular situations. This reveals a possibility of suppressing errors by exploiting the interaction between the individual operation steps. However, existing work in literature is mainly based on empirical results, and the conclusions can neither be generalized to emerging solutions of the involved operations nor be extended to other cases with different operation components. In this thesis, a statistical model is proposed to analyze the error interaction in a typical DSC image processing chain. This analysis model helps better understand the interaction between the sequential processes, and provides clues for better cooperative solutions. The analysis model is then applied to two specific cases in DSC, one on demosaicking and compression, and the other on demosaicking and sharpening.

Finally, we investigate the error detection and removal in digitized composite video frames. Our focus is on rainbow effect, which is caused by residual high-frequency luminance information in the extracted chrominance components after demodulation. Based on the observed characteristics of the artifacts, we propose a scheme to detect the artifact regions and then remove the artifacts. Furthermore, we investigate the source of rainbow-

effect artifacts and derive the governing formulas accordingly. Based on the formulation, we exploit the specific features in the luminance component to detect the artifact regions. After isolating the artifact regions, the derived correlation between the luminance and the resultant errors is exploited to remove the artifacts. We then extend our intra-frame restoration method to inter-frame for restoring video sequences by exploiting the inter-frame chrominance correlation. Experiments conducted on captured video sequences from composite video sources demonstrate that our method can effectively remove rainbow-effect artifacts. Experiments based on simulated video frames are also conducted to show that our method can improve the visual quality in terms of peak signal-to-noise ratio (PSNR) evaluation.

Acknowledgments

I would like to express my sincere gratitude to my advisor, Prof. Tan Yap-Peng, for his great guidance and support throughout my PhD study. I am especially grateful for his high standards in research and publication; his expert vision and instruction; and his patience. His dedicated and professional attitude for research and work inspires me a lot.

My special thanks go to my colleague Mr. Lian Naixiang, for his cooperation and discussion in my work. I also want to thank Prof. Chua Hock-Chuan and Prof. Vitali Zagorodnov for their generous help in revising some content parts of this thesis.

My thanks is also extended to all my colleagues in the Media Technology Lab for their sincere support and assistance. I want to thank all the faculty members in Division of Information Engineering for providing an excellent education and research environment.

I would also like to thank all my friends for their encouragements and generous support. Finally, I would like to thank my parents and Mr. Dang Yuxing for their love and support, which help me survive through tough periods.

Contents

Summary	3
Acknowledgements	6
List of Figures	12
List of Tables	20
1 Introduction	22
1.1 Motivation and Objectives	22
1.2 Background Knowledge	25
1.2.1 DSC image processing chain	25
1.2.2 Demodulation of composite video	27
1.3 Thesis Organization and Contributions	30
1.3.1 Part I: Error suppression in CFA demosaicking	30
1.3.2 Part II: Error analysis in DSC image processing chains	31
1.3.3 Part III: Error detection and removal in composite video frames	32
2 Adaptive Color Filter Array Demosaicking	34

2.1	Introduction	34
2.2	Demosaicking Strategies for Suppressing Artifacts	38
2.2.1	Exploiting spatial correlation	38
2.2.2	Exploiting spectral correlation	40
2.2.3	Suppressing demosaicking artifacts	42
2.3	Proposed Methods	43
2.4	Enhanced alternative projection (AP) method	49
2.5	Experimental Results	53
2.6	Computational Complexity	54
2.7	Conclusion	58
3	Demosaicking by Accurate Reconstruction of High-Frequency Information	61
3.1	Introduction	61
3.2	Observations About High-Frequency Content	64
3.3	Frequency Analysis of Bayer CFA Pattern	66
3.4	Better Luminance Filtering at Green Sample Positions	70
3.5	Adaptive Luminance Filtering at Red/Blue Sample Positions	73
3.6	Summary of the Proposed Method	77
3.7	Experimental Results	78
3.8	Discussion	86
3.8.1	Computational complexity	86
3.8.2	Modified CFA patterns	89
3.9	Conclusion	90

4	Demosaicking and Compression in DSC Image Processing	92
4.1	Introduction	92
4.2	Previous Work and Motivation	97
4.2.1	Alternative processing methods	97
4.2.2	Motivation and notation	98
4.3	Error Models of Processing Chains	100
4.3.1	Conventional: demosaicking-compression	101
4.3.2	Alternative: compression-demosaicking	105
4.3.3	Quick comparison of processing chains	109
4.4	Compression Error Model	110
4.4.1	Full-color image compression	112
4.4.2	CFA image compression	114
4.4.3	Evaluation of the compression error models	116
4.5	Performance Comparison of Two Processing Chains	117
4.6	Suitable Reference Images for Performance Evaluation	123
4.7	Conclusions	125
5	Demosaicking and Sharpening in DSC	128
5.1	Introduction	128
5.2	Error models	130
5.2.1	Conventional chain: demosaicking-sharpening	133
5.2.2	CFAS chain: sharpening- demosaicking	134
5.3	Proposed Solutions	136

5.3.1	Proposed method I	136
5.3.2	Proposed method II	141
5.4	Conclusion	147
6	Heuristic Solution of Rainbow Effect Removal	148
6.1	Introduction	148
6.2	Artifact Detection	150
6.2.1	First and second moments of prediction errors	152
6.2.2	Third moment of the chrominance	156
6.3	Artifact Removal	157
6.4	Experimental Results	161
6.5	Conclusion	162
7	Analysis, Detection, and Suppression of Rainbow-Effect	166
7.1	Introduction	166
7.2	Formulation of Rainbow-Effect Artifact	167
7.3	Detection of Artifact Regions	174
7.4	Suppression of Rainbow-Effect Artifacts	178
7.5	Removal of Rainbow-Effect Artifacts in Video Sequences	182
7.6	Experimental Results	185
7.7	Conclusion	189
8	Conclusion and Future Work	194
	Author's Publications	198

Bibliography

200

List of Figures

1.1	Diagram of image processing flow in a typical DSC.	25
1.2	Spectrum of composite signal in NTSC system.	29
2.1	(a) Bayer color filter array pattern and its (b) red, (c) green, and (d) blue samples.	35
2.2	Two common demosaicking artifacts. (a) Zipper-effect artifact(generated by AP method). (b) False-color artifact (produced by bilinear interpolation method).	37
2.3	Reference CFA samples, where subscripts i and j represent row and column indexes, respectively.	39
2.4	Original cropped regions from Image 19 and 27 and the demosaicked results.	46
2.5	Mean square error (MSE) evaluation of the reconstructed images with different numbers of refinement, each line represents a test image.	47
2.6	Original cropped regions from Image 27 and the demosaicked results.	48
2.7	Procedure of proposed method.	48
2.8	Results after each step of the original AP method (upper row) and the enhanced AP methods (lower row).	52
2.9	Procedure of proposed enhanced AP method.	52

2.10	Test images (Image 1 to Image 30, enumerated from left-to-right and top-to-bottom.)	54
2.11	Original cropped region from Image 3 and the results obtained by different demosaicking methods ('Proposed': the proposed method in Section 2.3, 'Enhanced AP': the proposed enhanced AP method in Section 2.4).	55
2.12	Original cropped region from Image 9 and the results obtained by different demosaicking methods.	56
2.13	Original cropped region from Image 27 and the results obtained by different demosaicking methods.	56
2.14	Original cropped region from Image 9 and the results obtained by different demosaicking methods.	57
3.1	An image and its spectrum.	65
3.2	(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 [45], which is not the concern of this chapter.	66
3.3	Fourier transforms of (a) quincuncial sampled green component, and (b) rectangular sampled red/blue component.	68
3.4	(a) Fourier transform and (b) a typical spectrum of a CFA image.	69
3.5	(a) The passband and (b) Fourier transform of the low-pass filter in [43]. . .	70
3.6	Proposed low-pass filter for green pixel luminance: (a) passband, (b) spectrum representation, and (c) impulse response.	73
3.7	(a) Luminance instead of green in CFA, (b) reference neighboring samples, and (c) the adaptive interpolation filter.	74

3.8	(a) color-difference image $Cr = L - R_I$ and its spectrum of regions with (b) diagonal edge, (c) horizontal edges, and (d) vertical edges.	75
3.9	(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.	76
3.10	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.	77
3.11	Image test set II (referred to as Flower and Portrait).	79
3.12	Demosaicked results of a cropped region from Flower.	82
3.13	Demosaicked results of a cropped region from Portrait.	83
3.14	Original cropped region from Image 3 and the results obtained by different demosaicking methods.	83
3.15	Original cropped region from Image 9 and the results obtained by different demosaicking methods.	84
3.16	Original cropped region from Image 27 and the results obtained by different demosaicking methods.	84
3.17	Original cropped region from Image 27 and the results obtained by different demosaicking methods.	86
3.18	Comparison of the AHD and proposed demosaicking method in texture re- gions.	87
3.19	Modified Bayer CFA patterns with (a) red and (b) blue in quincunx lattice. .	89

4.1	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; (c) feasible conventional image processing chain; (d) feasible alternative image processing chain.	95
4.2	Simplified block diagrams of DSC image processing chains: (a) conventional, (b) alternative.	96
4.3	Color transformation of a 2×2 Bayer CFA unit.	98
4.4	Y plane and its 45° rotation.	98
4.5	Cropped region of Bike image: (a) After bilinear demosaicking; (b) After bilinear demosaicking and JPEG (10:1) compression.	99
4.6	Block diagrams of error models.	100
4.7	Relationship between Ω^2 and ξ_s^2 for 15 test images. Each curve corresponds to one test image.	102
4.8	Relationship between Ω^2 and ζ^2 for 15 test images. Each curve corresponds to one test image.	103
4.9	Predicted total error Ω^2 of conventional chain using simplified model (4.2).	104
4.10	Relationship between Ψ_o^2 and ζ^2 for 15 test images. Each curve corresponds to one test image.	106
4.11	Relationship between Ψ_o^2 and ξ_o^2 for 15 test images. Each curve corresponds to one test image.	107
4.12	Predicted the error of alternative chain using simple model (4.5).	108
4.13	Average logarithm of compression error variance <i>versus</i> bitrate.	112
4.14	The ratio between the compression error variances of the full-color and CFA images.	117

4.15	Predicted <i>versus</i> exact performance comparison between conventional and alternative processing chains.	120
4.16	Performance comparison between the conventional processing chain using bilinear and the alternative one using IECL.	121
4.17	Comparison of the images reconstructed from a real CFA image by using <i>bilinear</i> demosaicking and JPEG2000 compression (10:1).	122
4.18	Comparison of the reconstructed images after demosaicking and JPEG2000 compression (10:1) of CFA data downsampled from a test image.	126
4.19	Relative performance of alternative and conventional processing chains for different reference images.	127
5.1	Conventional versus CFAS (Bilinear and AP refer to the demosaicking algorithms used).	131
5.2	Block diagrams of error models.	132
5.3	Relationship between Ω^2 and ζ_r^2 using different demosaicking techniques for the demosaicking-sharpening chain.	134
5.4	Block diagram of the proposed method I.	137
5.5	(a) Pass-band of proposed band-pass filter for extracting high-frequency components, two implementations: (b)(d) spectrum representation, and (c)(e) impulse response.	138
5.6	Result of the proposed method I and other methods under comparison. . . .	139
5.7	Result of the proposed method I and other methods under comparison. . . .	140
5.8	Block diagram of proposed II solution.	141
5.9	Result of the proposed method II and other methods under comparison. . .	144
5.10	Result of the proposed method II and other methods under comparison. . .	145

5.11	Result of the proposed methods I and II and other methods under comparison.	146
6.1	An example of rainbow-effect artifact on a captured video frame: (a) the artifact frame, (b) the enlarged artifact region.	149
6.2	Example of segmentation result: (a) segmented regions and (b) the artifact region.	151
6.3	Classification results using PCA projection and FLD thresholding of 4 test video frames.	155
6.4	Chrominance histogram of (a) an artifact region, and (b) a typical intact region.	156
6.5	Wavelet decomposition of a test image (rainbow-effect artifact region are circled).	158
6.6	Results for a captured video frame: (a) the captured frame, (b) the detected artifact region, and (c)–(e) restoration results.	162
6.7	Results for a captured video frame: (a) the captured frame, (b) the detected artifact region, and (c)–(e) restoration results.	163
6.8	Results for a simulated video frame: (a) the original image, (b) the simulated frame, (c) the detected artifact region, and (d)–(f) restoration results.	164
6.9	Results of a simulated video frame: (a) the original image, (b) the simulated frame, (c) the detected artifact region, and (d)–(f) restoration results.	165
7.1	Rainbow-effect artifacts: (a) vertical rainbow-like color strips, (b) the chrominance I values along the scan line marked in (a).	171
7.2	Artificially introducing rainbow-effect artifact onto a digital image: (a) original image region, (b) simulated artifact region; (c) and (d) are the chrominance I values along the scan line marked in (a) and (b), respectively.	172

7.3	Frequency distribution of the luminance Y and chrominance I signals of (a) the captured video frame shown in Figure 7.1, and (b) the simulated frame shown in Figure 7.2.	173
7.4	Decomposition structure of non-decimated wavelet packet transform for the NTSC luminance signal.	175
7.5	Luminance signal in the six subbands for the marked scan line of: (a) the captured video frame shown in Figure 7.1 and (b) the simulated video frame shown in Figure 7.2(b).	176
7.6	Distribution of feature values for four test video frames captured from composite video sources.	177
7.7	Decomposition structure of non-decimated wavelet packet transform for the NTSC chrominance signals.	178
7.8	Artifact chrominance signals in the three subbands for the marked scan line of: (a) the captured video frame shown in Figure 7.1, and (b) the simulated frame shown in Figure 7.2(b).	179
7.9	Histograms of the wavelet coefficients for the original and the artifact chrominance in subbands: (a) I_1 , (b) I_2 , and (c) I_3	180
7.10	Test video frames. (a) captured video frames. (b) simulated video frames. .	185
7.11	Detected artifact masks of the test video frames in Figure 7.10.	186
7.12	Restoration results on captured video frames using: (a) 2D Wiener filter, (b) SMIQ filter, and (c) our proposed method.	187
7.13	Restoration results on simulated video frames. (a) original frames, (b)-(d) restored frames using 2D Wiener filter, SMIQ filter, and our proposed method, respectively.	188

7.14 Removing rainbow-effect artifacts in video sequences. (a)-(b) Two successive captured video frames (the artifact region is highlighted with circle). (c) Result of inter-frame detection. Restoration results of (d) chroma-averaging, (e) proposed intra-frame removal, and (f) proposed inter-frame removal method.	190
--	-----

List of Tables

2.1	PSNR performance (in dB) comparison of bilinear and adaptive refinements.	46
2.2	Inter-pixel correlations of LH and HL subbands along the horizontal and vertical directions.	50
2.3	PSNR performance (in dB) comparison. For each image the PSNR results of 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.	58
2.4	Comparison of demosaicking Complexity.	59
3.1	MSE performance of the estimated green pixel luminance by different filters.	72
3.2	MSE performance of the estimated luminance at red/blue samples by different filters.	78
3.3	PSNR performance (in dB) comparison of bilinear and adaptive filters in Step 3.	79
3.4	PSNR performance (in dB) comparison. For each image the red, green and blue planes are listed in the 1st, 2nd and 3rd rows. The best PSNR of each row is shown in bold and underlined.	81
3.5	CPSNR performance (in dB) comparison in the edge (1st row) and smooth (2nd row) image regions. The best PSNR of each row is shown in bold and underlined.	85

3.6	Comparison of demosaicking complexity.	88
3.7	Comparison of PSNR performance (in dB) by applying FS approach [43] 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. . .	90
4.1	PSNR performance (in dB) of the conventional processing chain with bilinear demosaicking and JPEG/JPEG2000 compression.	99
4.2	Estimated values of a_c^2 and ρ_a in the model of conventional processing chain [(4.1)].	104
4.3	Estimated values of b_d^2 and ρ_b in the error model of alternative processing chain [(4.4)].	108
4.4	Similarity between $\rho_0 \cdot \nu_{s_1, s_2}$ and its exact values.	113
4.5	Comparison of processing chains (conventional vs. alternative).	118
5.1	Image variances and correlations after sharpening process.	136
5.2	Estimated values of α_f^2 and β_f^2	136
6.1	Average correlation coefficients of neighboring pixels of the component signals in artifact and intact regions.	152
6.2	Third moment of chrominance histogram for artifact and intact regions. . .	157
6.3	Comparison of PSNR results.	164
7.1	Triggering luminance subband and its resultant chrominance-artifact subbands.	179
7.2	Average inter-frame correlations of chrominance components and their subbands in the artifact and intact regions.	184
7.3	Comparison of PSNR performance.	189

Chapter 1

Introduction

1.1 Motivation and Objectives

With the development of image capturing devices and visual signal applications, there is a high demand for high-quality images. They not only give the viewer an accurate or a pleasing representation of captured scene but also provide sufficient details that may be critical in various applications. For example, high-quality images are required in medical imaging to make correct diagnosis and operation decisions. Surveillance systems require high-quality video frames to recognize faces, licence plates, etc. In aerial/satellite imaging, high-quality images are required to resolve small objects and to make correct detection/classification decisions.

Applying more advanced hardware is a straightforward way to improve the image quality; however, the cost could be too high for general commercial applications. Furthermore, the quality degradation may originate from limitations of acquiring or producing principles, such as limited field-of-view, limited spatial resolution, limited bandwidth, insufficient luminance or exposure. For example, when a digital still camera (DSC) captures continuous scenes with

limited pixels, the sensor sampling could introduce aliasing errors. When the luminance and chrominance components of a composite video signal are modulated into one signal, luminance/chrominance cross-talk errors could occur. These kinds of errors are inevitable and irrecoverable once produced. In these cases, an alternative approach is to develop more advanced image processing techniques that can eliminate or remove errors for better image quality. This is the main objective of this thesis.

Preserving sharp edges is important for high image quality, since edges contain the most important information of image. From psychological point of view, edges characterize the shape and texture of objects in images. From a statistical point of view, edges correspond to a large amount of uncertainty, and therefore could carry most of the visual information in images according to Shannon theory. However, edges are easy to be destroyed or eliminated; they are also prone to invoke errors in visual signal processing. For example, the above-mentioned aliasing errors and cross-talk errors are caused by overlapping high-frequency information within edge regions. As discussed in Chapter 2, demosaicking errors are mainly introduced around edge regions, owing to the difficulty in predicting the high-frequency content. Moreover, as indicated in Chapter 3, demosaicking errors can be regarded as the errors occur in separating the aliased high frequencies in the “luminance” and “chrominance” components of CFA spectrum. In Chapter 6, we show that the annoying rainbow-effects in composite videos are mainly caused by the residual high-frequency luminance in the extracted chrominance components after demodulation. Therefore, preserving the useful high frequencies while suppressing the undesirable errors is important in visual signal processing.

To suppress errors, we need the knowledge of the underlying model and the characteristics of the errors. Furthermore, as visual signal processing errors are closely correlated with high-frequency information processing, the properties of high-frequency information are important for devising error-suppression techniques. In this thesis, two key properties are

exploited. One is the anisotropic characteristics of high-frequency information, i.e., the inter-pixel correlation is larger along the edges while smaller across the edges. The other is the strong inter-color correlation in high-frequency content, and the corresponding diminishing high-frequency information in color-difference components and chrominance components. Specifically, the thesis addresses two visual signal processing applications: color filter array (CFA) demosaicking of DSC and rainbow-effect artifact removal for digitized composite video frames. In CFA demosaicking, anisotropic characteristics of high-frequency information are utilized to guide the interpolation direction, and inter-color correlations are exploited to reconstruct better high-frequency contents. In rainbow-effect removal, distinct features of high-frequency information in the luminance and chrominance components are exploited to detect the artifact regions. The artifacts are then removed based on the specific characteristics of errors and the property that chrominance components possess diminishing high-frequency information.

In addition to designing and optimizing individual processes in visual signal processing, some researchers proposed to eliminate errors by utilizing the system interaction among the concerned components. For example, several researchers proposed to reverse the order of demosaicking and compression operations in DSC and demonstrated that the alternative chain could outperform the conventional chain in low compression ratios [1–6]. On the other hand, a new sharpening processing was proposed to perform sharpening before demosaicking and the presented results demonstrated its effectiveness. However, existing work in the literature is mainly based on empirical results, and the conclusions can neither be generalized to emerging techniques of the involved operations nor be extended to other cases with different operation components. In this thesis, a statistical model is proposed to analyze the error interaction in image processing chains. This analysis model helps us better understand the interaction between the sequential processes, and provides clues for better cooperative solu-

tions. In this thesis, this analysis model is applied in two specific cases in DSC, one involving demosaicking and compression, and the other comprising demosaicking and sharpening.

1.2 Background Knowledge

In this section, we present an overview of the image acquisition procedure in DSC and the demodulation technique for composite video. These background knowledge will help us to understand not only the context of our addressed problems but also the sources of errors.

1.2.1 DSC image processing chain

A typical DSC uses an electronic sensor to acquire spatial variations in light intensity and then applies image processing techniques to reconstruct a full-color image from the data provided by the sensor [7, 8]. The flowchart shown in Figure. 1.1 briefly summarizes a general single-sensor DSC processing chain. In the following, the principle of a DSC image processing chain is introduced in four parts: Image Capture, CFA Demosaicking, Subsequent Image Processing, and Compression.

Image Capture DSC uses an electronic sensor, charge-coupled device (CCD) or complementary metal-oxide-semiconductor (CMOS), to record the image by using photosensitive elements that convert light to electrical signals.

Figure 1.1: Diagram of image processing flow in a typical DSC.

As a digital color image comprises three color components, three color values need to be measured at each pixel location. This requires the use of more than one sensor. As a cheaper and more robust solution, most digital still cameras use a single electronic sensor overlaid with a CFA to acquire imagery, in a way that each sensor pixel samples only one of the three primary colors.

Note that errors occur in the CFA sampling process because each color component of the image is usually sampled at one-half or one-quarter of the original resolution. The sub-sampling results in periodic replication of image spectrum, and the successive periods of the spectrum will overlap in high-frequency part [9, 10]. This error is referred to as aliasing artifact. As a means of suppressing the artifact, most cameras use an optical low-pass filter, which acts as an anti-aliasing filter (AA filter), before sampling [11]. The drawback is that there is a tradeoff between image sharpness and aliasing artifact.

CFA Demosaicking With CFA, each pixel of sensor measures only one of the primary colors. To restore a full color image, the two missing color values at each pixel need to be estimated. This estimation process is referred to as CFA demosaicking. Apart from image compression, demosaicking is probably the most computationally intensive step in a DSC processing chain, thus enhancing the operation/energy efficiency of demosaicking solution is critical for improving the overall performance of a DSC system.

Subsequent Image Processing The subsequent processes aim to make the captured images, when displayed, look similar to the original scenes as perceived by human eyes [7, 8].

- *White Balance* removes color tint of an image to ensure white objects appear white.
- *Color Correction* transforms the interpolated colors from the sensor color space to a standard color space for storage or output.
- *Gamma Correction* adjusts the image intensity to compensate the non-linearity of

cathode-ray-tube (CRT) display.

Some proprietary steps may also be included by different camera manufacturers, to further enhance the image quality. For example, sharpening is a common post-processing step employed in many DSCs. These optional techniques are mostly heuristic and camera specific, and require considerable fine-tuning.

- *Sharpening* compensates the blur caused by the lens, optical anti-aliasing filter, and CCD aperture to provide a subjectively sharper image.

Compression Compression reduces the number of bytes required to represent an image with no or some loss in perceived image quality. Since demosaicking produces full-color images, standard compression techniques can be used for the resultant images. EXIF storage format, a standard for most current DSCs [7], uses JPEG compression, which is based on discrete cosine transform [12]. Recently, wavelet-based compression (now formalized in the JPEG2000 compression standard [13]) has been shown to yield better performance than JPEG [14], requiring much fewer coefficients to represent image discontinuities. Because of its superiority over JPEG, JPEG2000 compression is likely to be incorporated into the future DSCs [7].

1.2.2 Demodulation of composite video

All present analog TV systems transmit color TV signals in a composite format [15, 16]. The composite format is also used for video storage on some analog tapes, such as video home system (VHS). Composite video formats were invented to make color TV signal compatible with monochrome TV system, in such a way that the luminance component can be extracted and displayed separately [15, 16]. In addition, a composite signal has a bandwidth that is significantly lower than the total bandwidth of the three component signals R , G and B , and

therefore can be transmitted or stored more efficiently. This is possible as the chrominance signals normally have a significantly smaller bandwidth than the luminance signal. However, these benefits are achieved at the expense of image quality: there often exist noticeable artifacts caused by aliasing between the luminance and chrominance components [15–17].

National television system committee (NTSC) and phase alternating line (PAL) are two most common color encoding systems used in broadcast television systems. Here we briefly describe the mechanism used by NTSC. First, the two chrominance components I and Q are obtained by rotating (by 33°) the two color-difference components $B - Y$ and $R - Y$. This process makes I correspond to colors in orange – cyan range, and Q the green – purple range [15]. The I and Q components are then quadrature amplitude modulated into a chroma signal C on a color subcarrier frequency f_c (3.58 MHz) and added to the luminance component Y . Finally, the audio signal is frequency modulated using an audio subcarrier frequency and added to the composite video signal. The entire composite signal is modulated onto a picture carrier f_y . The frequency of the picture carrier f_y depends on broadcasting channel [15].

Figure 1.2 illustrates the spectral composition of the composite signal. The bandwidth of the luminance is 4.2 MHz [16], and extends to 1.25 MHz below the picture carrier due to vestigial sideband modulation (VSB). Due to less sensitivity of human vision system (HVS) to the changes in chrominance signals, the I and Q are bandlimited and have much lower bandwidth than the luminance Y . Furthermore, the HVS is more sensitive to the spatial variations in orange - cyan range than in the green - purple range. Therefore, the I component is bandlimited to about 1.5 MHz, while the Q component to about 0.5 MHz [15]. Sometimes the I component is also bandlimited to 0.5 MHz to avoid interference between the audio signal and luminance signal; the same bandwidth is plotted for I and Q in Figure 1.2. The signal spectrum of PAL system is similar to that of NTSC system, except that the subcarrier

Figure 1.2: Spectrum of composite signal in NTSC system.

f_c is at 4.43 MHz while the luminance bandwidth is 5.5 MHz. Another difference is that the PAL system utilizes the color-difference components $B - Y$ and $R - Y$ to represent the chrominance information. Furthermore, in PAL the phase of the $R - Y$ signal is reversed by 180 degree from line to line [16]. Since in this thesis we consider and process the signal in each individual line separately, the difference from line to line is ignored.

At the receiver, the composite video signal is first demodulated to the baseband, and then the audio and three components of the video signals are demultiplexed. The separation of luminance and chrominance is challenging because of the overlapped spectrum [15]. Several techniques have been tried over the years to improve the separating performance. The simplest and least costly method is the low pass/high pass filter separation technique, where a low-pass filter with a cutoff around 2 MHz to 2.5 MHz is used to extract the luminance signal and a high-pass filter with a bandpass of 2.5 MHz to 5 MHz is used to filter the chrominance signal. Another technique uses a notch filter centered at the color subcarrier frequency to remove the chrominance information from 2.5 MHz to 4.5 MHz [15]. These two methods lead to cross-talk artifacts including cross-color and cross-luminance due to the frequency interleaving inherent in the composite video signal. Cross-color refers to spurious colors created by the high-frequency luminance signal that is close to the color subcarrier

frequency. Cross-luminance refers to false high-frequency edge patterns caused by the modulated chrominance information [15, 18].

1.3 Thesis Organization and Contributions

This thesis addresses the problem of error suppression in visual signal processing system. The investigations are organized into three parts: Error-suppressed CFA Demosaicking (Part I, Chapters 2 and 3), Error Model Analysis (Part II, Chapters 4 and 5), and Error Detection and Removal (Part III, Chapters 6 and 7). We conclude the thesis and sketch the future research direction in Chapter 8.

1.3.1 Part I: Error suppression in CFA demosaicking

Since the main error produced in DSC image capturing originates from color filter array (CFA) demosaicking, Part I of this thesis is devoted to addressing this problem.

Chapter 2 presents a new CFA demosaicking method which adaptively exploits the image spatial and spectral correlations to suppress demosaicking errors. The proposed method is based on two key properties of high-frequency information: anisotropic characteristic of high-frequency information and the similarity of high-frequency information across three color components. Our examination shows that edge-adaptive interpolation and high-frequency estimation across color components are effective in suppressing zipper-effect and false-color artifacts, respectively. Inspired by the analysis, the proposed method applies weighted interpolation in color-difference components for adaptive high-frequency estimation across color components. Experimental results confirm that the proposed method produces sharpened edges with less artifacts. It outperforms the existing methods in both visual

and peak signal-to-noise ratio (PSNR) evaluations at a moderate computational cost. Moreover, we enhance an existing algorithm – the alternative projections (AP) method by Gunturk *et al.* [19]– by making better use of spatial correlation. Experimental results of the enhanced method demonstrate the improvement with more visually favored results.

Chapter 3 describes a new adaptive filtering demosaicking method for better reconstruction of high-frequency information. By analyzing the spatial spectrum of a CFA image using sampling theory, we show that the luminance component preserve more high-frequency information than other color components due to the smaller bandwidth of the modulated color-difference components. Furthermore, we observe that extra high-frequency information is preserved in green component. Based on these observations, we have 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 methods both visually and in terms of PSNR measure, at a lower computational cost.

1.3.2 Part II: Error analysis in DSC image processing chains

The captured images are further processed for specific application requirement. For example, the images could be sharpened for more visually favored results, or be compressed to reduce the storage space and transmission bandwidth. In this case, optimizing the image processing chain, in addition to refining each individual process, is necessary for enhancing the produced image quality. In Part II, a system analysis is provided for CFA imaging chain by considering demosaicking jointly with compression or sharpening. A statistical error in-

teraction model is proposed based on the image variance and correlation for comparing the performance of different processing chains.

Chapter 4 provides better insight into the relative performance of the alternative compression chain and the conventional chain by analyzing 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 improvement 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.

Chapter 5 investigates the reconstruction error variances of the alternative sharpening methods in CFA image processing chain. Based on the analysis on the interaction between demosaicking and sharpening errors, we obtain two principles in solving the demosaicking-sharpening problem: 1) avoid enlarging errors when extracting the high-frequency information for sharpening, and 2) adding high-frequency component after demosaicking process to avoid degrading the demosaicking performance. According to the principles we propose two solutions. Experimental results show our proposed algorithms outperform existing chains.

1.3.3 Part III: Error detection and removal in composite video frames

Some errors occur at the receiving side of TV signals. The errors can only be removed by post-processing. Part III of this thesis introduces the rainbow-effect artifacts, which are caused by leaving high-frequency luminance information in the separated chrominance components. We investigate the underlying reason, formulate the rainbow effect artifacts, and summarize the corresponding properties. Based on the distinct features observed and the properties summarized, we design efficient methods to detect and remove the rainbow-effect

artifacts.

Chapter 6 proposes a heuristic method to restore video frames with rainbow-effect artifacts. Two distinct characteristics are observed from the artifacts: they usually occur in areas of high luminance frequencies and the resultant artifact signals are rainbow-like. Based on the observed characteristics, we propose a scheme to detect the artifact regions and then remove the artifacts. Our detection method exploits features in the component signals (luminance and chrominance) to discriminate artifact regions from intact regions. Once the artifact regions are identified and isolated, we suppress the artifacts by wavelet thresholding. Experiments conducted on captured video frames as well as simulated frames show that our method is effective in detecting artifact regions and restoring the contaminated region. Our method also improves the image quality quantitatively on the simulated frames in terms of PSNR measure.

Chapter 7 investigates the underlying reason of rainbow-effect artifacts in composite videos and derives the governing formulas accordingly. Based on the formulation, we exploit features in the luminance and the chrominance components to detect the regions contaminated by rainbow-effect artifacts. Once the artifact regions are isolated, we remove the artifacts by modifying large wavelet coefficients in the chrominance subbands. We then extend our intra-frame restoration method to inter-frame for restoring videos by exploiting the inter-frame chrominance correlation. Experiments conducted on video sequences from composite video sources demonstrate that our method can effectively remove rainbow-effect artifacts. Results on simulated video frames also show that our method can improve the image quality in terms of PSNR.

Chapter 2

Adaptive Color Filter Array

Demosaicking

2.1 Introduction

To reduce the cost and size, most digital still cameras (DSC) use a single electronic sensor covered with a color filter array (CFA) to acquire imagery, sampling only one color value at each pixel location. The most commonly used CFA configuration today is Bayer pattern [20], a schematic of which is shown in Figure 2.1 (a). In this pattern, green (G) values are sampled in a quincuncial lattice, while red (R) and blue (B) values are in two separate rectangular lattices, as shown in Figure 2.1 (b)-(d).

To restore a full-color image from its CFA samples, the two missing color values at each pixel are estimated by a process referred to as CFA demosaicking. A large number of demosaicking methods have been proposed in the literature [19,21–32]. Among them, the simplest and most referenced is bilinear interpolation [27, 33], which estimates a missing color value with the average of neighboring CFA samples of the same color. The estimation is possi-

ble because of the substantial spatial correlation between neighboring pixel intensities. The spatial correlation refers to the fact that within a homogenous region of an image, the pixels assume similar color values. In real images the spatial correlation holds in smooth regions but fails in edge regions. This explains why most demosaicking algorithms, including the simplest bilinear interpolation, perform well in smooth regions but usually produce visible artifacts around edge regions [34]. Consequently, the largest challenge of demosaicking lies in the accurate estimation of missing color values around edges.

Furthermore, the spatial correlation is anisotropic in edge regions, i.e., the correlation is larger along the edges while smaller across the edges. Several researchers proposed to exploit this property by edge-adaptive interpolation to improve the interpolation quality around edge regions [23, 25, 26, 28, 29]. The applied edge-adaptive interpolation operates by analyzing the pixel samples around each location to determine the edge direction, and then performing interpolation accordingly. This helps avoid interpolating across edges. Various techniques for identifying edges exist in literature. For example, Cok [21, 26] proposed to classify local image regions into several texture patterns and estimate the missing color values accordingly. Several edge-sensing methods [23, 26, 28] have also been proposed to determine the preferred interpolation direction with edge indicators along horizontal and vertical directions. Kimmel [29] adopted edge indicators as weights to approximate the

Figure 2.1: (a) Bayer color filter array pattern and its (b) red, (c) green, and (d) blue samples.

likelihood of edge direction. Experimental results demonstrated that edge-adaptive methods generally produce sharper edges and fewer artifacts in demosaicked images than isotropic algorithms [23, 25, 26, 28, 29, 33].

Another characteristic usually exploited to improve demosaicking performance is the strong inter-color correlation (also commonly referred to as spectral correlation) [25, 28, 32]. Many existing demosaicking algorithms have used this property by interpolating in color-difference images [24, 32, 34], since they are generally “smooth” and more suitable for interpolation. Representative algorithms include the effective color interpolation (ECI) proposed by Pei *et al.* [32], where color-difference components are bilinear interpolated for estimating missing color values, and Freeman’s method [24], which filters the color-difference components by a median filter to remove false colors. Furthermore, Gunturk *et al.* recently reported that the high-frequency information from different color components is strongly correlated and demonstrated this property by calculating the correlation values between the detail (high-frequency) wavelet coefficients [19]. Based on the strong correlation, they proposed an alternative projection (AP) method to reconstruct the aliased detail wavelet coefficients in the red and blue components from the corresponding coefficients in the green component. Their method achieves good performance and becomes a popular benchmark algorithm in CFA research literature. Experimental results in [35] further confirm that the detail wavelet coefficients of three color components are not only strongly correlated, but also similar to each other. This can explain why color-difference images are smooth and suitable for interpolation. The similarity also implies that in demosaicking one could reconstruct a full-resolution image component containing the high-frequency information, and then use it to estimate the missing color values.

By comparing the results obtained from various existing demosaicking methods, we have observed that utilizing spatial and spectral correlation can help suppress demosaicking arti-

facts. Furthermore, our test illustrates that exploiting spatial correlation can help suppress zipper-effect – artificial “on-off” patterns created around edges – as shown in Figure 2.2(a). On the other hand, utilizing spectral correlation benefits eliminating false-color artifacts – spurious colors which are not present in the original image scene – as seen in Figure 2.2(b).

(a) Zipper-effect artifact

(b) False-color artifact

Figure 2.2: Two common demosaicking artifacts. (a) Zipper-effect artifact (generated by AP method). (b) False-color artifact (produced by bilinear interpolation method).

In this chapter we propose a new demosaicking method which adaptively exploits image spatial and spectral correlations. It exploits weighted interpolation on color-difference components to adaptively estimate the missing high frequencies. Experimental results show that the proposed method generates better performance than several existing algorithms in terms of both visual quality and peak signal-to-noise-ratio (PSNR). In addition, the proposed method is competitive in computational efficiency. Also presented is an enhanced AP method which makes better use of spatial correlation. The experimental results demonstrate the improvement.

The remainder of the chapter is organized as follows. Section 2.2 discusses the techniques of exploiting the above mentioned key characteristics in demosaicking and their effectiveness in suppressing artifacts. Section 2.3 describes the procedure of the proposed method and Section 2.4 presents the improved AP method. The experimental results and complexity

analysis of the methods under comparison are presented in Section 2.5 and Section 2.6, respectively. The conclusion of the chapter is given in Section 2.7.

2.2 Demosaicking Strategies for Suppressing Artifacts

2.2.1 Exploiting spatial correlation

To exploit the anisotropic property of spatial correlation, several researchers assume simplified edge model which consists of edges in only two directions: vertical and horizontal. The rationale behind is that human visual system is more sensitive to edges and textures in the horizontal and vertical directions than those in other directions [36, 37]. The most commonly used technique calculates edge indicators in the horizontal and vertical directions respectively, and predicts edge orientation by evaluating the indicators. A simplest format of edge indicators is the first-order derivatives (gradient) of green values [23]. For example, for location in Figure 2.3(a), horizontal and vertical gradients at a location (i, j) are calculated as $\Delta H_1 = |G_{i,j-1} - G_{i,j+1}|$ and $\Delta V_1 = |G_{i-1,j} - G_{i+1,j}|$.

Different edge indicators exist in the literature, for example, the second-order derivatives of red (blue) values in [25]: $\Delta H_2 = |(R_{i,j-2} + R_{i,j+2})/2 - R_{i,j}|$ and $\Delta V_2 = |(R_{i-2,j} + R_{i+2,j})/2 - R_{i,j}|$. Hamilton's adaptive color interpolation method (ACI) [28] combines the first-order gradients of green values and second-order derivatives of red (blue) values as:

$$\begin{aligned}\Delta H &= |(R_{i,j-2} + R_{i,j+2})/2 - R_{i,j}| + |G_{i,j-1} - G_{i,j+1}| \\ \Delta V &= |(R_{i-2,j} + R_{i+2,j})/2 - R_{i,j}| + |G_{i-1,j} - G_{i+1,j}|\end{aligned}\tag{2.1}$$

Edge-indicator schemes generally require modest computational resources, and provide decent performance in regions with horizontal and vertical edges [32, 33]. Especially, Hamil-

Figure 2.3: Reference CFA samples, where subscripts i and j represent row and column indexes, respectively.

ton's edge indicator is reported to perform better than other edge indicators [33]. It is applied in the initial step for interpolating green component in Gunturk's AP method [19].

An extension to these approaches is to use a larger region around the pixel in question and more delicate edge predictors. For example, in the latest algorithms, primary-consistent soft-decision (PCSD) method [38] reconstructs the missing values of each pixel using consistent horizontal and vertical interpolations and choose among the two directional interpolations based on training results; and adaptive homogeneity-directed demosaicing algorithm (AHD) [39] first interpolates the missing color values by performing filter-bank interpolation in the horizontal and vertical directions, separately, and selects the proper interpolated value using some homogeneity criteria. These techniques incur high computational cost. Furthermore, the above-mentioned techniques assume edges in only two directions, and are prone to generating artifacts in regions with complex edges [33, 34].

Instead of choosing the edge direction, Kimmel [29] proposed to determine the likelihood of an edge direction and perform interpolation based on the likelihood. The algorithm defines edge indicators in eight possible directions with first-order derivatives of the green component, and then exploits the edge indicators as weights to estimate the missing color values. If the likelihood of an edge crossing a particular direction is high, the edge indicator

returns a small value and the neighboring pixels in that direction will contribute less to the estimation. Experimental results demonstrate that the weighted interpolation is capable of modeling edges in arbitrary directions, and performs well in regions with complex edges and fine details [29]. However it also incurs high computational complexity. In this chapter, we propose a simplified version of weighted-adaptive technique. We replace the first-gradient of green component by combining absolute first-gradients of green and red/blue components as edge indicators, and consider only weights in four directions – up, down, left, and right.

Refer to Figure 2.3(a), the weights along the four adjacent directions are defined as follows:

$$\begin{aligned}
 \alpha_{i-1,j} &= |R_{i-2,j} - R_{i,j}| + |G_{i-1,j} - G_{i+1,j}| \\
 \alpha_{i+1,j} &= |R_{i+2,j} - R_{i,j}| + |G_{i+1,j} - G_{i-1,j}| \\
 \alpha_{i,j-1} &= |R_{i,j-2} - R_{i,j}| + |G_{i,j-1} - G_{i,j+1}| \\
 \alpha_{i,j+1} &= |R_{i,j+2} - R_{i,j}| + |G_{i,j+1} - G_{i,j-1}|.
 \end{aligned} \tag{2.2}$$

The effectiveness of the proposed weighted interpolation will be demonstrated in Section 2.3.

2.2.2 Exploiting spectral correlation

Gunturk *et al.* have demonstrated that the high-frequency information is highly correlated across three color components, with correlation values ranging from 0.98 to 1 [19]. Generally, high correlation between two variables does not necessarily imply their equality. However, the smoothness of color-difference components suggests that the high-frequency information from different color components is not only correlated but also similar to each other. Experimental results in [35] show that generally the R, G, and B detail wavelet coefficients are approximately equal.

The ECI and AP methods are two commonly used methods based on image spectral correlation. ECI interpolates color-difference components by incorporating information from other color components, while AP replaces the high-frequency information across color components directly. From our analysis, color-difference interpolation scheme is also a means of exploiting the similarity of high-frequency information from three color components, which is consistent with the AP method. The following lemma explains why this approach works.

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

$$\mathcal{C} \approx \mathcal{I}(\mathcal{C}_s - \mathcal{F}_s) + \mathcal{F} \quad (2.3)$$

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 information, 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

$$\mathcal{I}(\mathcal{C}_s - \mathcal{F}_s) + \mathcal{F} \approx \mathcal{C}^l - \mathcal{F}^l + \mathcal{F}^l + \mathcal{F}^h \approx \mathcal{C} \quad (2.4)$$

■

In general, the full-resolution reference $\mathcal{F}$ in (2.3) can be any color component. However, the more densely-sampled green component is preferred [19, 32, 38]. This explains why many demosaicking algorithms take the similar procedure by estimating a full-resolution green component first and then using it to predict the red and blue components from color-difference images [19, 25, 28, 31, 32, 38, 40, 41].

2.2.3 Suppressing demosaicking artifacts

The artifacts are dominated by zipper-effect and false-color, as shown in Figure 2.2(a) and (b), respectively. One cause of zipper-effect could be the isotropic interpolation or interpolation across edges in quincunx-sampled green component [42]. The captured samples are interlaced with the estimated values, causing the “on-off” patterns. On the contrary, interpolation in red and blue component produces blurred edges or jagged edges, but not zipper-effect artifacts. Our study also shows that zipper-effect artifacts are more likely to occur around edges not aligned in the diagonal directions with 45° or 135° angles. Furthermore, as discussed in Section 2.2.2 green component usually serves as reference in red/blue interpolation, then less estimation errors will be propagated to red and blue components with better green interpolation. It can be concluded that edge-adaptive interpolation is more necessary for green component interpolation than other components.

Another cause of zipper-effect artifact is the indiscriminate replacement of high-frequency information from other color components. Specifically, the high spectral correlation is evident by considering the entire color components as a whole [19]; the local correlations between two color components may not be high at all pixel locations, e.g., around edge regions with abrupt hue change. Consequently, replacing high-frequency subbands between different color components indiscriminately may cause errors. This explains why the ECI [32] and AP methods [19] produce visible zipper-effect artifacts around edges, as shown in the experimental results in Section 2.5. This also implies that it is necessary to incorporate spatial correlation when exploiting spectral correlation. Based on this observation, we improve the AP method by making better use of spatial correlation in Section 2.4 and obtain satisfactory demosaicked results with less zipper-effect. However, the computational cost is high for real applications. On the contrary, color-difference interpolation, which can also predict high frequency from other color components, is computationally efficient. Further-

more, it is rather straightforward to apply existing edge-adaptive interpolation techniques in color-difference components for suppressing zipper-effect artifacts.

False colors are caused by the inconsistency between different color components. Applying spectral correlation can increase the inter-color consistence effectively and thus suppress false colors. Based on these insights, we propose a demosaicking method with the following steps: 1) reconstructing the green component with edge-adaptive interpolation, and 2) predicting the red and blue components by applying edge-adaptive interpolation in color-difference components. The procedure of the proposed method is detailed in the next section.

2.3 Proposed Methods

The proposed method contains two successive steps: initialization step and refinement step. The initialization step is inspired by Pei's ECI [32] method, replacing bilinear interpolation for color-difference components with edge-adaptive interpolation.

1. **Initialization step:** In this step, the missing green values at red and blue pixels positions are first estimated. According to Lemma 1, we can estimate the missing green values by using the color-difference components $K_r = G - R$ or $K_b = G - B$. This is a chicken-and-egg problem, as the references must have full resolution, which requires extension of the subsampled red/blue components to full resolution. However, the red/blue components also require a full-resolution reference, such as the full-resolution green.

We resolve 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 green (using Lemma 1). Referring to Figure 2.3 (a) as an example, the color-difference value $K_{r(i,j)}$ is estimated as

$$\tilde{K}_{r(i,j)}^R = \frac{\frac{\tilde{K}_{r(i-1,j)}^G}{1+\alpha_{i-1,j}} + \frac{\tilde{K}_{r(i+1,j)}^G}{1+\alpha_{i+1,j}} + \frac{\tilde{K}_{r(i,j-1)}^G}{1+\alpha_{i,j-1}} + \frac{\tilde{K}_{r(i,j+1)}^G}{1+\alpha_{i,j+1}}}{\frac{1}{1+\alpha_{i-1,j}} + \frac{1}{1+\alpha_{i+1,j}} + \frac{1}{1+\alpha_{i,j-1}} + \frac{1}{1+\alpha_{i,j+1}}} \quad (2.5)$$

where $\tilde{\cdot}$ denotes an estimated value. The four adjacent $\tilde{K}_r^G$ values are obtained by subtracting bilinearly interpolated red values from green values at each pixel, and the four weights are calculated according to (2.2). The missing green value at the red pixel location is then computed as

$$\tilde{G}_{i,j}^R = R_{i,j} + \tilde{K}_{r(i,j)}^R \quad (2.6)$$

The G values at B pixels can be similarly obtained.

After recovering the full-resolution green component, we use it as a reference to interpolate the red and blue components. For the red component, the missing red values at blue sample locations are first estimated due to the present of more supporting information: four red samples in the 3×3 neighbor, while other locations just have two. Referring to Figure 2.3(d), the missing red values at blue pixel locations are estimated with a similar adaptive weighted scheme using the four adjacent $\tilde{K}_r^R$ values along diagonal directions – $\tilde{K}_{r(i-1,j-1)}^R$, $\tilde{K}_{r(i+1,j+1)}^R$, $\tilde{K}_{r(i+1,j-1)}^R$, and $\tilde{K}_{r(i-1,j+1)}^R$. The weights

along the four adjacent directions are obtained as

$$\begin{aligned}
 \alpha_{i-1,j-1} &= |B_{i-2,j-2} - B_{i,j}| + |R_{i-1,j-1} - R_{i+1,j+1}| \\
 \alpha_{i+1,j+1} &= |B_{i+2,j+2} - B_{i,j}| + |R_{i+1,j+1} - R_{i-1,j-1}| \\
 \alpha_{i+1,j-1} &= |B_{i+2,j-2} - B_{i,j}| + |R_{i+1,j-1} - R_{i-1,j+1}| \\
 \alpha_{i-1,j+1} &= |B_{i-2,j+2} - B_{i,j}| + |R_{i-1,j+1} - R_{i+1,j-1}|
 \end{aligned} \tag{2.7}$$

The desired $K_{r(i,j)}^B$ is then computed by

$$\tilde{K}_{r(i,j)}^B = \frac{\frac{\tilde{K}_{r(i-1,j-1)}^R}{1+\alpha_{i-1,j-1}} + \frac{\tilde{K}_{r(i+1,j+1)}^R}{1+\alpha_{i+1,j+1}} + \frac{\tilde{K}_{r(i+1,j-1)}^R}{1+\alpha_{i+1,j-1}} + \frac{\tilde{K}_{r(i-1,j+1)}^R}{1+\alpha_{i-1,j+1}}}{\frac{1}{1+\alpha_{i-1,j-1}} + \frac{1}{1+\alpha_{i+1,j+1}} + \frac{1}{1+\alpha_{i+1,j-1}} + \frac{1}{1+\alpha_{i-1,j+1}}} \tag{2.8}$$

and the missing $R_{i,j}$ value in the blue pixel location is then obtained as

$$\tilde{R}_{i,j}^B = \tilde{G}_{i,j}^B - \tilde{K}_{r(i,j)}^B \tag{2.9}$$

where $\tilde{G}_{i,j}^B$ is the estimated green value at a blue pixel location.

Once all the missing red values at blue pixel locations have been obtained, the four adjacent $\tilde{K}_r$ values around the green pixels (as in Figure 2.3(b) and (c)) become available. We then compute the weights of these adjacent color-difference values with (2.2) and estimate the missing red values by using (2.5) and (2.6) to fully populate the red component.

The missing blue values can be similarly estimated. The quantitative evaluation of the initialization step is presented in Table 2.1 with PSNR results, comparing with those of the ACI method [28] and ECI method [32]. The result demonstrates the efficacy of incorporating edge-adaptive techniques, outperforming the ECI method

by about 1 dB on average in each color component. The result also shows that our proposed weighted interpolation is more effective than the ACI method with edge indicators. Two experimental results in Figure 2.4 show an improved performance of our proposed method in suppressing demosaicking artifacts.

Figure 2.4: Original cropped regions from Image 19 and 27 and the demosaicked results.

Table 2.1: PSNR performance (in dB) comparison of bilinear and adaptive refinements.

Image		ACI	ECI	Initial	Bilinear refinement	Adaptive refinement	Image		ACI	ECI	Initial	Bilinear refinement	Adaptive refinement	
1	R	33.19	33.21	34.96	37.40	37.35	6	R	34.80	34.65	35.92	38.29	38.04	
	G	34.71	35.65	36.54	41.11	40.64		G	36.10	36.98	37.69	41.51	41.20	
	B	33.32	33.38	35.22	38.04	37.85		B	34.33	34.12	35.43	37.31	37.32	
2	R	37.20	36.85	37.98	37.18	37.57	7	R	40.63	40.31	42.09	41.78	42.10	
	G	40.84	41.28	42.43	43.32	43.86		G	42.20	42.16	43.98	45.14	45.42	
	B	39.47	39.35	40.95	41.23	41.67		B	40.24	39.69	41.30	40.74	41.11	
3	R	40.22	40.51	41.73	41.64	42.16	8	R	31.70	30.16	32.82	34.86	34.60	
	G	42.21	43.11	43.89	45.32	45.75		G	33.48	33.09	35.18	39.00	38.65	
	B	39.90	40.07	40.93	41.33	41.61		B	31.70	29.87	32.77	35.10	34.83	
4	R	36.62	36.88	37.67	36.88	37.38	9	R	39.40	38.66	40.95	41.39	41.84	
	G	40.72	42.16	42.34	44.58	44.43		G	41.37	41.37	43.05	45.19	45.29	
	B	39.54	40.77	41.45	43.26	42.95		B	40.04	39.32	40.96	42.94	42.18	
5	R	34.75	35.04	36.47	37.47	37.46	10	R	39.07	39.37	40.90	41.00	41.28	
	G	35.93	36.84	37.82	41.05	40.95		G	41.24	42.07	43.38	45.57	45.53	
	B	34.38	34.74	35.93	36.71	36.86		B	39.21	39.43	40.77	42.09	41.71	
							Average	R	36.76	36.56	38.15	38.79	38.98	
								G	38.88	39.47	40.63	43.18	43.17	
								B	37.21	37.07	38.57	39.88	39.81	

- Refinement step:** After the initialization step, the three color components are fully populated, and their estimates can be further refined with the initialized color values.

The motivation is that in the initialization step the reference red and blue components for the green estimation are obtained by bilinear interpolation, and the reconstructed red and blue components are supposed to be more accurate and thus better as reference. The refinement is performed by weighted interpolating (adaptive refinement) or averaging (bilinear refinement) the four adjacent color-difference values, and updating three color components using Lemma 1. The adaptive refinement is similar to our initialization step with given full-resolution red and blue components in the first green estimation.

The refinement process can be repeated several times until no further improvement is obtained. However, excessive iterations will negate the improvement; see Figure 2.5. This negation occurs because the equivalent processing region after several iterations becomes quite large, involving a substantial number of pixels which are slightly correlated. Experimental results show that the performance negation generally occurs at the second iteration for most test images, and hence we apply the refinement step only once.

Figure 2.5: Mean square error (MSE) evaluation of the reconstructed images with different numbers of refinement, each line represents a test image.

Table 2.1 shows that the refinement step improves the initial results especially in green

component for about 2.3 dB (for adaptive refinement) and 2.5 dB (for bilinear refinement). Note that the PSNR results in Table 2.1 indicate that bilinear interpolation scheme is sufficient for refinement; it even produces slightly better PSNR results, on average, in green and blue components. However, our visual results give the different conclusion. One example is in Figure 2.6. It reveals that bilinear refinement could yield extra zipper-effect due to isotropic interpolation in color-difference components. Since the perceptual quality is the ultimate goal of most demosaicking, we apply adaptive refinement in this step. The observation also suggests the necessity in using subjective evaluations along with objective measures.

Figure 2.6: Original cropped regions from Image 27 and the demosaicked results.

The proposed method can be summarized with Figure 2.7.

Figure 2.7: Procedure of proposed method.

2.4 Enhanced alternative projection (AP) method

In this section, we describe an improvement made to the AP method by making better use of spatial correlation. The AP method consists of three main steps: initial step, update step, and projection step. The initial step uses an edge-adaptive demosaicking method (i.e., ACI) [28] to interpolate the green component and bilinear interpolation to fill the red and blue components. In the update step, the estimated green component is refined by replacing the high-frequency information from captured red and blue component. The high frequency substitution is conducted in wavelet subbands. For example, the captured red samples are aggregated to form an image R_r , and the estimated green values at red pixel locations r are also integrated into an image G_r . R_r and G_r are then decomposed into four subbands - LL, LH, HL, and HH, and G_r is reconstructed with the LH, HL, and HH subbands from R_r :

$$\tilde{G}_r = \text{Rec}(G_{rLL}, R_{rLH}, R_{rHL}, R_{rHH}) \quad (2.10)$$

where Rec represents the wavelet composition, G_{rLL} , R_{rLH} , R_{rHL} , and R_{rHH} denote the subbands obtained from G_r and R_r , respectively.

In the projection step, the red and blue components are reconstructed by replacing the high-frequency subbands (LH, HL, and HH) from green component. After each time of high-frequency information update, the captured red and blue samples are inserted to the reconstructed red and blue components. The projection step repeats until a stopping criterion is attained.

Our observation shows that the zipper-effect artifacts are caused by indiscriminately replacing the inter-color high frequencies in the update and projection steps. In particular, zipper-effect artifacts in the reference component, if any, would be spread to other component in high frequency replacement. To suppress the zipper-effect artifacts, we modify the update step for green component to make it adaptive to local spectral correlation: in image

regions with low inter-color correlations, we estimate the high-frequency information from neighboring green samples instead of from red/blue components. We make use of the fact that the values of the LH subband are highly correlated along the horizontal direction, while that for the HL subband holds along the vertical direction. Table 2.2 shows the correlation values of the LH and HL subbands of 10 test images along the horizontal and vertical directions. In the enhanced update step, when the inter-color correlation is low, the LH and HL subbands of the green component can be estimated from the subband values of the captured green samples along the row and column, respectively.

Table 2.2: Inter-pixel correlations of LH and HL subbands along the horizontal and vertical directions.

Image No.	Horizontal						Vertical					
	R		G		B		R		G		B	
	LH	HL	LH	HL	LH	HL	LH	HL	LH	HL	LH	HL
1	0.8998	0.2686	0.9022	0.2481	0.8999	0.2708	0.3255	0.8711	0.3051	0.8771	0.3313	0.8788
2	0.8905	0.2825	0.8984	0.3076	0.9025	0.3056	0.3365	0.8581	0.3986	0.8449	0.3941	0.8443
3	0.8579	0.0481	0.8596	0.0420	0.8617	0.1132	0.3439	0.8446	0.3483	0.8449	0.3933	0.8437
4	0.8160	0.2504	0.8156	0.2702	0.8147	0.2623	0.1698	0.8426	0.1763	0.8574	0.1737	0.8564
5	0.7955	0.1627	0.7997	0.1445	0.7981	0.1802	0.1710	0.7372	0.1551	0.7384	0.1903	0.7300
6	0.8907	0.2932	0.8891	0.2920	0.8881	0.3088	0.3657	0.8212	0.3718	0.8216	0.4077	0.8134
7	0.8640	0.0721	0.8678	0.0712	0.8744	0.0680	0.1533	0.8306	0.1208	0.8283	0.1609	0.8309
8	0.8908	0.3373	0.8904	0.3300	0.8881	0.3543	0.3430	0.9282	0.3363	0.9277	0.3615	0.9282
9	0.8771	0.2156	0.8780	0.1927	0.8744	0.2458	0.2123	0.9110	0.1937	0.9119	0.2368	0.9090
10	0.8601	0.2620	0.8631	0.2415	0.8584	0.2641	0.1680	0.8171	0.1456	0.8166	0.1818	0.8110
Average	0.8642	0.2193	0.8664	0.2140	0.8660	0.2373	0.2589	0.8462	0.2552	0.8469	0.2831	0.8446

To update the green component, we first compute the local inter-color correlation between the concerned color components from the initial step [19]. The local correlation is calculated with the green and red/blue values in a 5×5 window using (2.11).

$$Corr_{local}(i, j) = \frac{\sum_{m=i-w/2}^{i+w/2} \sum_{n=j-h/2}^{j+h/2} (x(m, n) - \mu_x)(y(m, n) - \mu_y)}{\sqrt{\sum_{m=i-w/2}^{i+w/2} \sum_{n=j-h/2}^{j+h/2} (x(m, n) - \mu_x)^2} \sqrt{\sum_{m=i-w/2}^{i+w/2} \sum_{n=j-h/2}^{j+h/2} (y(m, n) - \mu_y)^2}} \quad (2.11)$$

where (i, j) represents the pixel coordinates, x denotes the red or blue value, y denotes the green value, and μ_x and μ_y are the average of x and y , respectively, in the local window consisting of $(w+1) \times (h+1)$ pixels. Note that accurate correlation evaluation relies on the

accurate estimation from the initial step; thus, we replace the original initial step with our enhanced initialization step. Experimental results demonstrate its superiority by improving the AP performance for about 0.5 dB, on average. According to the local inter-color correlation, we estimate the high frequency for green component selectively from red/blue samples (in the case of high correlation) or neighboring green samples (in the case of low correlation). The following is an example of the low correlation case. Referring to Figure 2.3(a), the missing green values at red pixel positions are interpolated as:

- Obtain the interpolated LH and HL subbands by

$$\begin{aligned}\tilde{G}r_{LH} &= f_0(n1) * Gg_{LH} \\ \tilde{G}r_{HL} &= f_0(n2) * Gg_{HL}\end{aligned}\quad (2.12)$$

where $f_0(n1) = [1 \ 3 \ 3 \ 1]/8$, $f_0(n1) = f_0(n2)^T$, and $[.]^T$ denotes the transpose.

- Reconstruct the alternative green component $\tilde{G}r$ by using the interpolated $\tilde{G}r_{LH}$ and $\tilde{G}r_{HL}$ subbands as:

$$\tilde{G}r = \text{Rec}(Gr_{LL}, \tilde{G}r_{LH}, \tilde{G}r_{HL}, Rr_{HH}) \quad (2.13)$$

The missing green values at blue pixel locations, the $\tilde{G}b$ component, can be updated similarly. The original projection step from the AP method is retained. Figure 2.8 shows the results after each step in the original and enhanced AP methods. It demonstrates the effectiveness of the proposed adaptive update for green component in suppressing zipper-effect artifacts. Note that although in the projection step the high-frequency information of the red/blue components are totally replaced with that from the green component, our experimental results (see more in Section 2.5) show that less visible zipper-effect is introduced.

This is due to the accurate estimation of the green component and the spectral correlation used in the initial step and update step consisting of sufficiently high inter-color correlation. Hence we discard incorporating edge adaptive scheme in the projection step to strike a balance between the performance and computational complexity.

Figure 2.8: Results after each step of the original AP method (upper row) and the enhanced AP methods (lower row).

The proposed enhanced AP method can be summarized with Figure 2.9.

Figure 2.9: Procedure of proposed enhanced AP method.

2.5 Experimental Results

To evaluate the performance of our proposed adaptive CFA demosaicking method and the enhanced AP method, we compare their demosaicked results with those of the five existing demosaicking methods—bilinear interpolation, Freeman’s median filter method (MF) [24], Hamilton’s ACI method [28], Pei’s ECI method [32], and Gunturk’s AP method [19]. Among the methods under comparison, bilinear interpolation is the simplest and also the most common reference for comparison in demosaicking literature. Both the MF method and ACI method can obtain good demosaicked results as reported in [24, 28, 33]. The ECI and AP methods, popular algorithms in the recent literature, can produce superior results as presented in [19, 32] and confirmed by our empirical study.

Figure 2.10 shows our test image set of 30 color images from Kodak PhotoCD; most of the images have been widely used for CFA demosaicking research and experimentation in the literature [19, 29, 31–33].

Figures 2.11, 2.12, and 2.13 show the cropped regions from three original test images (Images 3, 9 and 27 in Figure 2.10) and the corresponding demosaicked images. It can be observed that the three adaptive methods – ACI method, enhance AP method and the proposed methods – produced visually favored results with sharper edges and fewer zipper-effect in Figures 2.11 and 2.12, while other isotropic demosaicking algorithms yielded visible zipper-effect artifacts around edges. Similar observation applies to Figure 2.13 and Figure 2.14. It is also observed that the ACI method produces some zipper-effect and false-color around edges along the diagonal directions. This can be seen in Figure 2.4. Our proposed adaptive demosaicking method and the enhanced AP method obtain the most visually favored results with few demosaicking artifacts.

Table 2.3 presents the PSNR evaluation of the methods under comparison. The PSNR

Figure 2.10: Test images (Image 1 to Image 30, enumerated from left-to-right and top-to-bottom.)

values show that our proposed method outperforms others in almost all test images. On the other hand, the enhanced AP method improves the original AP method by 0.65 dB, 0.67 dB and 0.5 dB in the red, green, and blue components, respectively.

2.6 Computational Complexity

Demosaicking is typically performed in digital still cameras, that have limited computational resource. Hence, the complexity is an important factor for evaluating a demosaicking algorithm.

Let $M \times N$ be the size of the image to be demosaicked. For the proposed method, the green component interpolation requires $15.5MN$ additions, $4MN$ absolute conversions, $4.5MN$ multiplications, and $2MN$ bit-shift operations. The red and blue interpolations re-

Figure 2.11: Original cropped region from Image 3 and the results obtained by different demosaicking methods ('Proposed': the proposed method in Section 2.3, 'Enhanced AP': the proposed enhanced AP method in Section 2.4).

quire a total of $11.5MN$ additions, $4MN$ absolute conversions, and $4.5MN$ multiplications. As for the refinement step, $8MN$ additions and $10MN$ multiplications are needed. All in all, a total of $39MN$ additions, $19MN$ multiplications, $8MN$ absolute conversions, and $2MN$ bit-shift operations are required by the proposed method.

For the original AP method, the initial step requires $5MN$ additions, $2MN$ absolute conversions, and $2MN$ bit-shift operations to compute each missing green value, and another $2.5MN$ additions and $1.5MN$ bit-shift operations to bilinearly interpolate the missing red or blue values, amounting to a total of $7.5MN$ additions, $2MN$ absolute conversions, and $3.5MN$ bit-shift operations for the entire image. As reported in [19], the remaining steps of the original AP method requires about $384MN$ additions and $384MN$ multiplications. Our modification made to the AP's initial interpolation requires $31MN$ additions, $9MN$ absolute conversions, $8MN$ multiplications, and $2MN$ bit-shift operations. Additionally, the change made to the green component update step requires $38MN$ additions and $27.5MN$ multi-

Figure 2.12: Original cropped region from Image 9 and the results obtained by different demosaicking methods.

Figure 2.13: Original cropped region from Image 27 and the results obtained by different demosaicking methods.

Figure 2.14: Original cropped region from Image 9 and the results obtained by different demosaicking methods.

plications. Hence, the enhanced AP method requires a total of $453MN$ additions, $9MN$ absolute conversions, $2MN$ bit-shift operations, and $419.5MN$ multiplications.

Table 2.4 lists the number of computations required by different methods. The complexity figure of the ECI method [32] is taken directly from the corresponding paper. For the MF and ACI methods, we perform the complexity analysis based on the algorithms described in the respective papers [24,28]. Here we approximate the median filter in a 3×3 window with 16 additions, but in implementation median filtering could be time-consuming.

As shown in Table 2.4, the ECI method and ACI method are the least complex, followed by the proposed adaptive method, the original AP method, and the enhanced AP method. Although the proposed adaptive method requires more computations than the ACI and ECI method, this increase in complexity is justifiable for the superior demosaicking quality. On the other hand, it is more computationally effective, and performs better in terms of both visual and PSNR results than the original AP method. The enhanced AP method, which is the most comparable to the proposed method in visual and PSNR results, requires nearly 12

Table 2.3: PSNR performance (in dB) comparison. For each image the PSNR results of 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.

This row is shown in bold and underlined															
Image	Bilinear	MF	ACI	ECI	AP	Enhanced AP	Proposed	Image	Bilinear	MF	ACI	ECI	AP	Enhanced AP	Proposed
1	25.31	30.75	33.19	33.21	37.17	<u>37.86</u>	37.35	16	30.30	35.37	38.21	37.51	<u>41.46</u>	40.93	41.08
	29.58	37.08	34.71	35.65	40.02	<u>40.76</u>	40.64		34.73	41.78	39.71	40.14	<u>44.39</u>	44.17	44.19
	25.35	30.90	33.32	33.38	36.96	37.50	<u>37.85</u>		30.39	35.56	38.06	37.32	40.46	40.02	<u>40.59</u>
2	31.89	35.95	37.20	36.85	38.29	<u>39.62</u>	37.57	17	31.63	38.02	38.51	39.14	41.48	<u>41.87</u>	41.42
	36.26	42.36	40.84	41.28	40.20	41.62	<u>43.86</u>		34.54	41.94	39.30	40.35	43.65	<u>44.01</u>	43.73
	32.36	37.77	39.47	39.35	38.73	40.00	<u>41.67</u>		30.86	37.22	37.39	38.59	40.13	40.33	<u>40.66</u>
3	33.45	39.02	40.22	40.51	41.29	<u>42.42</u>	42.16	18	27.32	33.39	33.53	34.92	36.74	<u>37.03</u>	36.28
	37.17	44.27	42.21	43.11	43.23	44.64	<u>45.75</u>		30.54	37.48	34.85	36.54	39.47	<u>39.60</u>	<u>39.60</u>
	33.83	39.08	39.90	40.07	39.82	41.27	<u>41.61</u>		26.82	32.88	33.42	34.82	35.95	36.22	<u>36.85</u>
4	32.61	37.07	36.62	36.88	37.10	<u>38.90</u>	37.38	19	26.78	31.75	36.67	34.53	39.44	<u>39.63</u>	39.32
	36.53	43.44	40.72	42.16	42.15	43.42	<u>44.43</u>		31.74	39.23	38.33	37.23	42.58	42.71	<u>42.78</u>
	25.91	32.44	39.54	40.77	41.29	42.15	<u>42.95</u>		26.95	32.11	37.01	34.76	39.11	39.02	<u>40.04</u>
5	32.75	32.88	34.75	35.04	37.10	<u>38.92</u>	37.46	20	30.81	37.08	38.61	38.93	41.10	<u>41.80</u>	41.65
	29.32	38.82	35.93	36.84	39.42	40.54	<u>40.95</u>		34.58	42.38	39.76	40.56	43.20	43.76	<u>44.27</u>
	25.92	33.02	34.38	34.74	35.53	36.20	<u>36.86</u>		30.59	36.35	37.32	37.48	38.20	38.60	<u>39.53</u>
6	26.70	32.14	34.80	34.65	<u>38.34</u>	38.21	38.04	21	27.63	33.33	35.15	35.63	38.85	<u>39.38</u>	39.16
	31.05	38.56	36.10	36.98	41.21	<u>41.42</u>	41.20		31.54	39.06	36.40	37.57	41.53	<u>42.04</u>	41.91
	27.00	32.30	34.33	34.12	37.07	36.83	<u>37.32</u>		27.53	33.20	34.62	34.97	37.27	37.48	<u>37.89</u>
7	32.57	39.10	40.63	40.31	41.85	<u>42.68</u>	42.10	22	29.84	34.92	35.97	36.41	37.71	<u>37.95</u>	37.65
	36.47	44.18	42.20	42.16	43.32	44.26	<u>45.42</u>		33.34	39.32	37.91	38.64	39.79	40.35	<u>41.01</u>
	32.58	38.84	40.24	39.69	39.54	40.36	<u>41.11</u>		29.22	34.05	35.78	35.74	36.29	36.69	<u>37.43</u>
8	22.53	27.40	31.70	30.16	35.36	<u>35.60</u>	34.60	23	34.43	40.10	41.15	40.84	41.90	<u>42.97</u>	41.72
	27.40	34.87	33.48	33.09	37.99	38.40	<u>38.65</u>		37.98	44.88	43.47	43.84	43.38	44.36	<u>45.65</u>
	22.48	27.40	31.70	29.87	34.26	34.32	<u>34.83</u>		34.12	40.08	41.68	41.68	39.84	41.02	<u>42.40</u>
9	31.56	36.86	39.40	38.66	41.37	<u>41.89</u>	41.84	24	26.40	32.73	32.51	34.37	34.90	<u>35.41</u>	34.75
	35.72	43.09	41.37	41.37	44.00	44.26	<u>45.28</u>		29.38	36.62	35.52	35.54	37.27	<u>37.77</u>	37.69
	31.38	37.09	40.04	39.32	40.81	41.11	<u>42.18</u>		25.29	30.37	30.65	31.94	32.59	32.99	<u>33.00</u>
10	31.84	38.02	39.07	39.37	41.44	<u>41.92</u>	41.28	25	25.90	30.36	30.75	31.92	32.48	32.62	<u>32.65</u>
	35.37	43.66	41.24	42.07	44.30	44.77	<u>45.53</u>		30.28	36.97	33.23	35.62	36.75	<u>37.08</u>	37.00
	31.23	37.51	39.21	39.43	40.63	40.94	<u>41.71</u>		26.01	30.22	30.61	31.72	32.37	<u>32.44</u>	32.42
11	28.17	33.70	35.63	35.62	38.66	<u>39.00</u>	38.32	26	29.65	34.31	33.90	34.41	35.37	35.57	<u>36.62</u>
	32.22	39.80	37.21	38.06	40.93	41.41	<u>42.20</u>		32.72	38.51	34.42	36.73	39.05	38.52	<u>39.66</u>
	28.34	34.20	36.05	36.09	38.36	38.53	<u>39.27</u>		29.58	34.66	33.10	34.98	36.33	36.29	<u>37.43</u>
12	32.67	37.77	39.94	39.22	42.08	<u>42.31</u>	41.80	27	22.47	28.01	29.78	29.28	30.77	31.83	<u>32.37</u>
	36.82	43.74	42.36	42.40	44.30	45.00	<u>45.77</u>		26.10	33.12	30.97	31.22	33.35	34.71	<u>34.91</u>
	32.35	37.70	40.28	39.38	41.17	41.36	<u>42.11</u>		22.32	27.56	29.08	28.62	30.21	31.01	<u>31.40</u>
13	23.09	29.52	29.65	31.43	34.40	<u>35.14</u>	34.69	28	25.70	31.49	33.90	33.22	35.57	35.94	<u>36.72</u>
	26.49	34.20	30.69	32.64	36.79	<u>37.66</u>	36.89		29.76	37.68	35.12	35.29	38.49	39.14	<u>39.86</u>
	23.00	29.14	29.23	30.64	33.71	<u>33.52</u>	33.43		25.94	31.64	33.73	32.91	34.98	35.42	<u>36.14</u>
14	28.15	33.54	34.97	34.78	35.81	<u>38.26</u>	35.86	29	32.75	38.06	39.89	38.55	39.43	39.57	<u>41.07</u>
	32.01	38.91	37.27	37.87	37.75	39.86	<u>40.66</u>		36.34	43.20	42.02	41.14	42.78	43.27	<u>45.08</u>
	28.60	34.24	35.21	35.15	34.12	35.78	<u>36.65</u>		33.49	39.33	40.82	39.63	40.39	40.59	<u>42.57</u>
15	31.05	35.13	35.84	36.33	38.02	<u>39.04</u>	36.65	30	27.01	29.54	29.19	28.99	28.38	29.07	<u>30.11</u>
	35.36	42.35	39.44	41.11	40.09	41.47	<u>42.87</u>		29.46	46.48	34.55	31.48	30.72	31.60	<u>32.88</u>
	32.26	38.17	37.88	39.31	38.70	39.58	<u>40.53</u>		24.09	26.63	26.83	26.58	26.50	27.12	<u>27.81</u>
Average	28.87	34.24	34.76	35.72	37.83	<u>38.44</u>	38.10	Average	32.69	39.80	37.34	38.29	40.40	41.09	<u>41.78</u>
	28.76	34.24	34.76	35.77	37.02	37.49	<u>38.39</u>								

times the number of additions and multiplications.

2.7 Conclusion

In this chapter, we advocate the advantage of exploiting image spatial and spectral correlation for CFA demosaicking. First, we summarized the efficient techniques in the literature for exploiting spatial correlation and spectral correlation, and their effectiveness in suppressing demosaicking artifacts. There are two key techniques: edge-adaptive interpolation and

Table 2.4: Comparison of demosaicking Complexity.

Method	No. of computational operations			
	Addition	Absolute	Bit-shift	Multiplication
MF	$53.5MN$		$2.5MN$	
ACI	$11MN$	$4MN$	$6MN$	
ECI	$10.0MN$		$2.5MN$	
AP	$392.5MN$	$2MN$	$3.5MN$	$384MN$
Enhanced AP	$468.5MN$	$9MN$	$2MN$	$447MN$
Proposed	$39MN$	$8MN$	$2MN$	$19MN$

high frequency approximation. Our analysis concludes that weighted interpolation is effective in modeling complex edges and details. We also show that owing to the similarity of high-frequency across three color components, one full-resolution component can be used to estimate the high-frequency information of other color components through color-difference interpolation.

Based on our analysis, we propose a new demosaicking method which incorporates both the spatial and spectral correlations. The proposed method applies weighted interpolation in color-difference components for adaptive high frequency estimation across color components. Experimental results indicate that the proposed method produces sharper edges with fewer artifacts. It improves the existing methods in both visual and PSNR evaluations at the expense of a moderate increase in computational cost.

We also observe that zipper-effect artifacts are mainly caused by isotropic interpolation in green component or indiscriminately replacing high-frequency information from other color components. Based on the insights, we have enhanced the AP method. Using a selective subband high frequency estimation scheme and an improved initial step, our enhanced AP method effectively eliminates zipper-effect artifacts.

Comparing to some most lately proposed algorithms [38,39,41,43], our proposed method produces better PSNR results. Furthermore, it was noted by Chung [44] as the method producing the best results in all his benchmarking algorithms and employed as the refinement

step in his method [44].

Chapter 3

Demosaicking by Accurate Reconstruction of High-Frequency Information¹

3.1 Introduction

More insights about demosaicking can be obtained from Fourier domain analysis. Sub-sampling of a full-color image to obtain CFA samples results in periodic replication of Fourier spectrum of the original image. According to Whittaker-Shannon sampling theorem, 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 [9, 10]. The demosaicking process is equivalent to recovering the original image spectrum, while suppressing the aliasing artifacts. To extract the original image spectrum, bilinear and bicubic

¹This chapter results from a collaboration with Mr Lian Naixiang. My work mainly focused on theoretical analysis, algorithm development, design and running of experiments.

interpolation techniques remove the aliasing by eliminating the overlapped high-frequency content. This elimination is equivalent to low-pass filtering [22, 26], which yields blurred edges, especially in the more sparsely sampled red and blue components. A good demosaicking technique should remove the aliasing artifacts while at the same time preserving as much high-frequency information as possible.

To better preserve high-frequency information in demosaicking, the key is to exploit the strong correlations across color components. As summarized in Section 2.2.2, two typical techniques for utilizing inter-color correlation are color-difference component interpolation [32] and high-frequency estimation across color components [19]. They operate by reconstructing a full-resolution image component containing the high-frequency information, and then using it to estimate the missing color values. Our study in Chapter 2 demonstrates that color-difference interpolation is preferred to strike a balance between demosaicking performance and computational efficiency. Furthermore, our analysis shows that applying edge-adaptive techniques in color-difference interpolation is necessary for suppressing zipper-effect artifacts.

Recently, Alleysson *et al.* [43] showed that the frequency spectrum of a CFA image can be represented as the combination of that of luminance at baseband and that of chrominance modulated to high spatial frequency. Here luminance is a spatial map of the image intensity and chrominance is the image chromatic (color-difference) information. This distribution is analogous to the component multiplexing used in analog composite video standard such as NTSC and PAL [15]. These color video systems introduce various aliasing distortions such as cross-color and cross-luminance artifacts [15, 16, 18]. Similarly Alleysson *et al.* [43] formulated four kinds of possible demosaicking artifacts – blur, grid, watercolor and false color. They concluded that blur and grid zipper-effect artifacts are due to underestimation and overestimation of luminance component, due to too narrow-band and wide-band low-

pass filtering respectively. Underestimation eliminates desired high-frequency information while overestimation includes high-frequency content of chrominance. On the other hand, water color and false color are caused by underestimating and overestimating the chrominance components. Their study also suggests that inaccurate recovery of high-frequency information in either luminance or chrominance is prone to demosaicking errors. It also indicates that proper preservation and reconstruction of high-frequency information is an important task for CFA demosaicking.

Based on the observation that the spectrum of luminance is well separated from that of chrominance in the frequency domain of a Bayer CFA image, Alleysson *et al.* [43] recently proposed an interesting frequency selection (FS) approach to preserve high-frequency information. They employed an appropriately designed low-pass filter to extract a full-resolution luminance plane, and then used it as a reference to reconstruct the missing color values. This process is consistent with Lemma 1, with luminance replacing the green component as a reference. Since an image's luminance component is a composition of three color components, the high-frequency information is preserved because of the similarity across different color components. Hence it is reasonable to use luminance as a reference. However, the filter proposed in [43] unnecessarily eliminates high-frequency information along the horizontal and vertical axes, which is important for image quality as we will demonstrate in Section 3.2. Furthermore, the isotropic filtering technique in color-difference components results in visible zipper-effect artifacts. More details of Alleysson's frequency selection (FS) demosaicking will be provided in Section 3.3.

For better interpretation of the frequency composition and distribution of CFA images, we first present a simple analysis on the frequency representation of Bayer CFA samples. Our analysis shows that the important high-frequency information along the two axes can be 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 [43] is reasonably good but not optimal. Based on the analysis conducted, we propose in Section 3.4 an efficient low-pass filter for the green samples to better preserve the high-frequency information. The high-frequency information at red/blue pixel positions is then estimated by adaptive filtering of color-difference components (Section 3.5). Combining several techniques for preserving and reconstructing 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 3.7. We also show that the proposed method has a lower computational complexity and rectify a fallacious assumption from FS demosaicking [43] in Section 3.8.

3.2 Observations About High-Frequency Content

In this section, we discuss an important observation about high-frequency information, which serves as a guideline to better preservation techniques. The 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, as explained below.

In natural images the energy spectrum is primarily concentrated in a low-frequency region, along the horizontal and vertical axes [10]. A typical example is shown in Figure 3.1. The following experiments further demonstrate that high-frequency components along the axes are more important for preserving image quality than those at the diagonal corners.

Shown in Figure 3.2 are the images which are obtained by removing some frequency components. We first removed the frequency coefficients outside the central circle [see Figure 3.2(a)], preserving 99.9%, 99.8% and 99.7% of the total energy; see results in Fig-

Figure 3.1: An image and its spectrum.

ure 3.2(b)-(d). Compare these with the images in Figure 3.2(f)-(h), which were obtained by removing the frequency coefficients outside a region that is more elongated along the two axes [see Figure 3.2(e)]. The latter set of images are visually more pleasing than the former, even though the total preserved energy is the same. For example the image in Figure 3.2(h) with 99.7% preserved energy appears similar to the image with 99.8% preserved energy in Figure 3.2(c).

This experiment implies that high-frequency information along the horizontal and vertical axes has a 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 [46–48].

As we will show in the next section, the high frequencies of the green component along the horizontal and vertical axes are free from aliasing in the Bayer CFA pattern, making it possible to better preserve them in the demosaicked images.

Figure 3.2: (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 [45], which is not the concern of this chapter.

3.3 Frequency Analysis of Bayer CFA Pattern

For three sub-sampled components of Bayer CFA image [Figure 2.1(b)-(d)], the sampling operations can be expressed via multiplication by a comb function $D_C(m, n) = D(m - a_C, n - b_C) = \sum_{p=0}^{\frac{M-2}{2}} \sum_{q=0}^{\frac{N-2}{2}} \delta(m - a_C - 2p, n - b_C - 2q)$, where offsets $a_C, b_C \in \{0, 1\}$ depend on the color component $C \in \{R, G, B\}$. We can rewrite $D_C(m, n)$ in terms of cosine modulation [43], for example for $a_C, b_C = 0$,

$$D(m, n) = \frac{[1 + \cos(\pi m)][1 + \cos(\pi n)]}{4} \quad (3.1)$$

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

$$D(u, v) = \Delta(u)^T \begin{bmatrix} \frac{1}{16} & \frac{1}{8} & \frac{1}{16} \\ \frac{1}{8} & \frac{1}{4} & \frac{1}{8} \\ \frac{1}{16} & \frac{1}{8} & \frac{1}{16} \end{bmatrix} \Delta(v) \quad (3.2)$$

where $\Delta(u) = \left[\delta(u + \frac{1}{2}) \quad \delta(u) \quad \delta(u - \frac{1}{2}) \right]^T$. As expected, the sampling operation replicates periodically the Fourier transform of the component.

We can easily extend $D(u, v)$ to the three sampling functions. The expression of $D_{\mathcal{C}}(u, v)$ for $\mathcal{C} \in \{R, G, B\}$ just has different phase with $D(u, v)$. In addition, the middle side terms in $D_{\mathcal{G}}(u, v)$ are cancelled due to the inverse phase in $D(m, n)$ and $D(m-1, n-1)$. Let $\mathcal{C}_f(u, v)$ and $\mathcal{C}_s(u, v)$ be the Fourier transforms of the full-resolution and sub-sampled color component $\mathcal{C}$. We can obtain $\mathcal{C}_s(u, v)$ by convolving $D_{\mathcal{C}}(u, v)$ with $\mathcal{C}_f(u, v)$:

$$R_s(u, v) = \left(\Delta(u)^T \begin{bmatrix} -\frac{1}{16} & \frac{1}{8} & -\frac{1}{16} \\ -\frac{1}{8} & \frac{1}{4} & -\frac{1}{8} \\ -\frac{1}{16} & \frac{1}{8} & -\frac{1}{16} \end{bmatrix} \Delta(v) \right) * R_f(u, v) \quad (3.3)$$

$$B_s(u, v) = \left(\Delta(u)^T \begin{bmatrix} -\frac{1}{16} & -\frac{1}{8} & -\frac{1}{16} \\ \frac{1}{8} & \frac{1}{4} & \frac{1}{8} \\ -\frac{1}{16} & -\frac{1}{8} & -\frac{1}{16} \end{bmatrix} \Delta(v) \right) * B_f(u, v) \quad (3.4)$$

$$G_s(u, v) = \left(\Delta(u)^T \begin{bmatrix} \frac{1}{8} & 0 & \frac{1}{8} \\ 0 & \frac{1}{2} & 0 \\ \frac{1}{8} & 0 & \frac{1}{8} \end{bmatrix} \Delta(v) \right) * G_f(u, v) \quad (3.5)$$

Here $*$ denotes the convolution. 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 3.3. This means that the high-frequency information along the horizontal and vertical axes in the sub-sampled 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 explains why the green component 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 3.3: Fourier transforms of (a) quincuncial sampled green component, and (b) 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 (3.3), (3.4) and (3.5) that its Fourier transform is

$$A(u, v) = \mathbf{1}^T \begin{bmatrix} \frac{1}{16} [2G_f - R_f - B_f] (u + \frac{1}{2}, v + \frac{1}{2}) & \frac{1}{8} [R_f - B_f] (u + \frac{1}{2}, v) & \frac{1}{16} [2G_f - R_f - B_f] (u + \frac{1}{2}, v - \frac{1}{2}) \\ -\frac{1}{8} [R_f - B_f] (u, v + \frac{1}{2}) & \frac{1}{4} [2G_f + R_f + B_f] (u, v) & -\frac{1}{8} [R_f - B_f] (u, v - \frac{1}{2}) \\ \frac{1}{16} [2G_f - R_f - B_f] (u - \frac{1}{2}, v + \frac{1}{2}) & \frac{1}{8} [R_f - B_f] (u - \frac{1}{2}, v) & \frac{1}{16} [2G_f - R_f - B_f] (u - \frac{1}{2}, v - \frac{1}{2}) \end{bmatrix} \mathbf{1} \quad (3.6)$$

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

Figure 3.4 shows the energy distribution in the CFA spectrum, following (3.6). Note that

Figure 3.4: (a) Fourier transform and (b) a typical spectrum of a CFA image.

$L = \frac{1}{4}(2G + R + B)$ 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, (3.6) can be equivalently expressed as:

$$A(u, v) = \mathbf{1}^T \begin{bmatrix} C_2 & C_1 & C_2 \\ C_1 & L & C_1 \\ C_2 & C_1 & C_2 \end{bmatrix} \mathbf{1}$$

. The smooth color-difference components (see Section 2.2.2) have a more compact spectrum which have less overlap with the luminance spectrum (Figure 3.4), as compared with the overlap between spectrum replicas in subsampled components (Figure 3.3). This implies that the luminance component preserves more high-frequency information than other color components, and possesses the advantage as a reference plane against green component.

Based on this property of CFA spectrum, Alleysson *et al.* [43] recently proposed a frequency selection (FS) approach for demosaicking, extracting the full-resolution luminance L by low-pass filtering the CFA image. The estimated luminance is then used as a reference to predict the RGB components (Lemma 1). Figure 3.5 shows an example of the filter used in [43], 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 (Fig-

Figure 3.5: (a) The passband and (b) Fourier transform of the low-pass filter in [43].

ure 3.3), it is clear that the Alleysson's filter can better preserve high frequencies due to less overlap in the CFA spectrum.

However, the Alleysson's filter falls short in several respects. First, since Alleysson *et al.* designed the filter based on some knowledge of the original image [43], the design becomes difficult when the original image is not available. Second, the filter discards more high-frequency information along the horizontal and vertical directions, compared to the filter for the subsampled green component [see Figure 3.3(b)]. This can lead to degradation in image quality, as discussed in Section 3.2.

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

3.4 Better Luminance Filtering at Green Sample Positions

According to the CFA spectrum $A(u, v)$ in (3.6), 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 cancelled out each other,

$$A(u, v) * D_G(u, v) = G_s(u, v) \quad (3.7)$$

The resultant spectrum has no replicas at the sides, as shown in Figure 3.3(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

$$A(u, v) = \mathbf{1}^T \begin{bmatrix} 0 & C_1 & 0 \\ C_1 & L & C_1 \\ 0 & C_1 & 0 \end{bmatrix} \mathbf{1}$$

be the spectrum with the corner components $C2$ removed, then

$$A'(u, v) * D_G(u, v) = \frac{1}{4}[2G + R + B](u, v) * D_G(u, v) = L(u, v) * D_G(u, v) \quad (3.8)$$

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 3.3(a), 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 3.1.

An alternative solution is to simply modify the Alleysson's filter with stopband $r1 = 0$ (see Fig 3.5(a)), as shown in Figure 3.6(a). This substantially reduces the MSE (see the third column in Table 3.1). We decided to redesign the Alleysson's filter to reduce the com-

Table 3.1: MSE performance of the estimated green pixel luminance by different filters.

Image No.	Quincuncial filter (11×11)	Alleysson's filter (11×11)	Modified Alleysson's filter ($r1 = 0; 11 \times 11$)	Proposed filter (5×5)
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

putational cost and dependency on the prior image information. The original Alleysson's filter has size 11×11 , while our designed filter has size 5×5 ; see the spectrum and pulse response in Figure 3.6(b)(c). Our examination shows that the MSE changes slightly when the size is larger than 5. The passband of the redesigned filter is fixed, and hence 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 3.1). This can be explained by relatively little luminance energy in the spectrum's corners, and hence having less effect by adaptively adjusting the passband position in this direction. Computationally, it requires only 5 multiplications and 20 additions for each pixel, compared to 21 additions and 68 multiplications for the Alleysson's 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 do not cancel out at these locations. To address this problem we propose using an adaptive filtering scheme, as described in the following section.

Figure 3.6: Proposed low-pass filter for green pixel luminance: (a) passband, (b) spectrum representation, and (c) impulse response.

3.5 Adaptive Luminance Filtering at Red/Blue Sample Positions

According to Lemma 1, we can interpolate the estimated, subsampled version of luminance (at green-sample locations, see Section 3.4) to full resolution by using the difference images $Cr = L - R$ or $Cb = L - B$ [Figure 3.7(a)]. Similar to the solution in Section 2.3, we resolve 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. After that, 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 to update the luminance estimation, and our experiments show that two iterations are sufficient in most cases.

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

$$\hat{L} = \mathcal{I}_{bil}(L_s - \mathcal{F}_s^l) + \mathcal{F}^l \approx L^l \quad (3.9)$$

Figure 3.7: (a) Luminance instead of green in CFA, (b) reference neighboring samples, and (c) the adaptive interpolation filter.

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

We define the adaptive filter coefficients $w_k, k \in [1, 2, 3, 4]$, shown in Figure 3.7(c), to be proportional to the spatial correlation along the corresponding directions. It can adapt to different neighborhood spatial patterns. For example, the stripes in Figure 3.8(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 3.9(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 3.8(b) the corresponding adaptive filter is shown in Figure 3.9(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 in Figure 3.7(b), four weights α_i are calculated similar as (2.2), replacing G with L values. The coefficients w_i are obtained as

$$w_i = \frac{1}{1 + \alpha_i}, \quad (3.10)$$

Figure 3.8: (a) color-difference image $Cr = L - R_T$ and its spectrum of regions with (b) diagonal edge, (c) horizontal edges, and (d) vertical edges.

α_i is defined in (2.7).

Luminance at blue-sample locations can be interpolated similarly. The luminance prediction results, shown in Figure 3.10, demonstrate that the proposed adaptive filter yields more accurate luminance component compared with bilinear interpolation.

The filtering performance is improved substantially from the first to the second iteration, but further improvements are minor. Table 3.2 shows that there is an increase in MSEs in the inverse improvement occurs at the third iteration for 6 of 10 test images. Hence we perform two iterations in the red/blue luminance estimation. 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.

Figure 3.9: (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.

Figure 3.10: 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.

3.6 Summary of the Proposed Method

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

1. Apply the filter shown in Figure 3.6 to a CFA image to obtain the luminance values at green-sample locations.
2. Apply the adaptive filter in Figure 3.7 (c) with coefficients defined in (3.10) to estimate the luminance values at the red/blue-sample locations.
3. Use the estimated full-resolution luminance as a reference to bilinearly interpolate the sub-sampled color components (red, green, and blue) of CFA to reconstruct the full-color image based on Lemma 1.

Table 3.2: MSE performance of the estimated luminance at red/blue samples by different filters.

Image No.	Green samples 5×5 filter	Alleysson's 11×11 filter	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

Note that we use adaptive interpolation in Step 2 but bilinear interpolation in Step 3. Bilinear interpolation is sufficient in Step 3; visual results in Section 3.7 also show that bilinear interpolation in Step 3 does not introduce zipper-effect artifacts. This is because the well-estimated luminance preserves the high-frequency information from all three color components and leads to sufficiently smooth color-difference images. If we apply the adaptive filter in Step 3, the performance is improved by only 0.31 dB on average, as shown in Table 3.3. The slight improvement might not justify the 70% increase in computations.

3.7 Experimental Results

To evaluate the proposed method we compare it with five state-of-the-art demosaicking techniques—Alternating Projections (AP) [19], Successive Approximation (SA) [41], Primary-Consistent Soft-Decision (PCSD) [38]², Frequency Selection (FS) [43], and Adaptive Homogeneity-Directed (AHD) demosaicking [39]—as well as bilinear interpolation. These methods except bilinear interpolation are recently proposed and can obtain good re-

²The PCSD algorithm we used here includes the directional filtering that further improves the PSNR performance by an average of over 1.5 dB [38].

Table 3.3: PSNR performance (in dB) comparison of bilinear and adaptive filters 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

Figure 3.11: Image test set II (referred to as Flower and Portrait).

sults.

Two sets of test images were used in the experiments reported in this chapter. Test Set I is shown in Figure 2.10. Test Set II includes 2 raw data captured by Canon EOS 300D; Figure 3.11 shows the corresponding full color image obtained by Photoshop CS2 camera raw's plug-in [49].

The PSNR results obtained by the methods used for comparison are summarized in Table 3.4. Our method offers better results than the other methods for 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. Table 3.5 further compares the CPSNR performance in edge and smooth regions, where CPSNR (color peak signal-to-noise ratio) is defined as $\text{CPSNR} = -10 \log(\frac{1}{3} \sum_{c \in \{R, G, B\}} \text{MSE}_c)$ for RGB values normalized to $[0, 1]$. We used Canny edge detection [50] followed by thresholding is applied to partition an image into edge and smooth regions. Compared with the existing state-of-the-art methods, our method generally improves in both edge and smooth regions.

Figures 3.12 and 3.13 show the cropped regions of demosaicked results of two real CFA images in test set II. The proposed method produces visually favored results with sharper edges compared to most other methods in Figure 3.12, except the PCSD and AHD methods that yield similarly good results. For the texture region in Figure 3.13, the FS and our method obtains superior results with fewer false colors than other methods. The zoom-in observations illustrate some spurious stripes produced by the AHD method. Similar results can be found on the test set I.

Figures 3.14 – 3.17 show cropped regions of four test images from test set I and the corresponding demosaicked images. It can be observed that the AHD and our methods obtain the best visual 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 [34], the AHD method can produce perceptually favorable images with fewer artifacts, which is also confirmed by the results in Figures 3.14 – 3.17. The approach uses a type of anisotropic filtering, based on some homogeneity criteria

3.7. Experimental Results

81

Table 3.4: PSNR performance (in dB) comparison. For each image the red, green and blue planes 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	Image	Bilinear	AP	PCSD	SA	FS	AHD	Ours	
1	25.31	37.17	36.35	36.66	35.06	34.62	36.68	16	30.30	41.46	43.05	38.66	39.16	41.43	40.72	
	29.58	40.02	38.85	40.88	39.64	36.28	39.92		34.73	44.39	45.14	43.40	43.99	42.91	43.83	
	25.35	36.96	36.55	37.64	35.45	34.79	36.90		30.39	40.46	42.06	39.40	38.85	40.61	40.22	
2	31.89	38.29	37.32	35.54	34.04	36.59	37.71	17	31.63	41.48	40.64	40.71	40.06	39.23	40.96	
	36.26	40.20	42.59	40.06	39.70	41.56	43.71		34.54	43.65	42.56	43.13	43.06	40.68	43.45	
	32.36	38.73	40.78	39.07	36.57	40.19	40.10		30.86	40.13	40.41	39.93	40.03	38.65	40.69	
3	33.45	41.29	41.46	39.08	38.68	40.64	42.19	18	27.32	36.74	35.13	35.51	34.90	33.58	35.57	
	37.17	43.23	44.67	42.21	42.20	43.74	45.63		30.54	39.47	37.64	38.61	38.37	35.56	38.79	
	33.83	39.82	40.88	39.36	38.49	40.16	41.41		26.82	35.95	35.75	36.43	35.55	33.56	36.35	
4	32.61	37.70	36.31	36.17	35.21	35.74	37.16	19	26.78	39.44	38.72	38.55	35.75	37.44	38.68	
	36.53	42.15	43.31	42.64	42.99	41.74	44.67		31.74	42.58	41.41	42.33	40.46	39.39	42.30	
	32.91	41.29	42.18	40.88	40.76	40.98	42.63		26.95	39.11	39.79	39.63	36.73	37.62	39.99	
5	25.75	37.70	36.73	35.59	34.97	35.14	37.51	20	30.81	41.10	40.21	38.75	38.59	38.98	40.14	
	29.32	39.42	39.80	38.27	38.47	37.46	40.84		34.58	43.20	42.37	41.74	41.97	40.71	43.59	
	25.92	35.53	36.14	35.34	34.80	34.44	36.86		30.59	38.20	37.80	37.22	36.60	36.75	38.39	
6	26.70	38.34	39.32	37.22	36.25	37.60	37.70	21	27.63	38.85	37.86	37.83	36.93	36.52	38.34	
	31.05	41.21	41.49	41.40	40.81	39.13	40.74		31.54	41.53	40.00	41.61	40.98	37.89	41.18	
	27.00	37.07	37.99	36.67	35.67	36.51	36.85		27.53	37.27	36.85	36.98	36.22	35.48	37.38	
7	32.57	41.85	41.02	40.12	38.89	40.12	42.27	22	29.84	37.71	36.78	36.93	36.40	35.69	37.81	
	36.47	43.32	43.76	42.67	42.44	42.71	45.61		33.34	39.79	39.52	39.63	39.28	38.47	41.20	
	32.58	39.54	39.82	39.35	38.08	39.08	40.93		29.22	36.29	36.37	36.87	35.51	35.36	37.42	
8	22.53	35.36	34.34	34.44	31.47	33.23	34.54	23	34.43	41.90	39.36	38.06	36.97	40.05	38.94	
	27.4	37.99	37.48	38.74	36.63	35.36	38.21		37.98	43.38	44.11	41.83	41.27	44.15	44.93	
	22.48	34.26	34.45	35.00	31.64	33.26	34.53		34.12	39.84	39.79	40.54	37.65	39.18	40.75	
9	31.56	41.37	41.05	40.60	38.68	40.36	41.37	24	26.40	34.90	35.02	34.79	35.29	32.58	35.04	
	35.72	44.00	44.21	43.33	42.34	43.06	45.17		29.38	37.27	37.12	37.59	37.83	34.96	37.51	
	31.38	40.81	42.52	41.36	39.74	40.70	43.07		25.29	32.59	33.00	32.80	32.74	31.53	32.83	
10	31.84	41.44	40.46	40.12	39.28	39.53	41.17	25	25.9	32.48	33.34	32.14	31.78	33.28	32.50	
	35.37	44.30	44.46	43.86	43.55	42.92	45.54		30.28	36.75	38.25	36.48	36.60	38.05	37.44	
	31.23	40.63	41.50	40.56	40.14	39.85	42.06		26.01	32.37	33.48	32.43	31.81	33.06	32.69	
11	28.17	38.66	38.02	37.60	36.45	36.66	38.28	26	29.65	35.37	34.37	35.28	34.62	33.40	35.69	
	32.22	40.93	40.96	41.70	41.05	39.03	41.81		32.72	39.05	37.27	39.46	39.06	35.68	39.09	
	28.34	38.36	38.85	38.69	37.51	37.48	38.98		29.58	36.33	35.03	36.51	35.86	34.33	36.34	
12	32.67	42.08	41.64	41.07	39.05	40.63	41.75	27	22.47	30.77	31.93	30.99	29.76	30.81	31.75	
	36.82	44.30	45.41	44.71	43.62	44.01	45.77		26.10	33.35	34.40	33.32	33.03	32.92	34.48	
	32.35	41.17	42.10	41.39	39.51	40.99	41.86		22.32	30.21	30.90	30.20	29.23	29.76	30.83	
13	23.09	34.40	33.43	34.70	34.57	31.54	33.68	28	25.70	35.57	36.15	36.00	34.38	35.41	36.19	
	26.49	36.79	34.95	37.90	37.72	32.41	35.98		29.76	38.49	38.83	39.11	38.22	37.73	39.50	
	23.00	33.01	32.43	33.35	33.30	30.62	32.74		25.94	34.98	35.48	35.33	33.97	34.77	35.84	
14	28.15	35.81	34.40	33.33	32.44	34.17	36.10	29	32.75	39.43	39.62	39.89	37.84	39.63	40.38	
	32.01	37.75	38.32	36.74	36.88	38.02	40.52		36.34	42.78	43.41	43.14	41.64	43.87	44.52	
	28.60	34.12	35.29	34.39	33.17	34.95	36.22		33.49	40.39	40.68	40.71	39.04	40.31	41.63	
15	31.05	38.02	36.28	35.34	34.85	35.52	37.19	30	27.01	28.38	28.38	28.01	26.75	28.01	29.21	
	35.36	40.09	41.84	40.58	40.55	40.69	43.13		29.46	30.72	31.01	30.23	30.03	31.42	32.59	
	32.26	38.70	39.67	38.83	37.45	38.61	40.24		24.09	26.50	26.47	26.19	25.17	26.46	27.13	
								Average	28.87	37.83	37.29	36.66	35.64	36.27	37.61	
									32.69	40.40	40.51	40.24	39.81	39.08	41.37	
									28.76	37.02	37.50	37.10	35.91	36.33	37.81

Figure 3.12: Demosaicked results of a cropped region from Flower.

Figure 3.13: Demosaicked results of a cropped region from Portrait.

Figure 3.14: Original cropped region from Image 3 and the results obtained by different demosaicking methods.

Figure 3.15: Original cropped region from Image 9 and the results obtained by different demosaicking methods.

Figure 3.16: Original cropped region from Image 27 and the results obtained by different demosaicking methods.

Table 3.5: CPSNR performance (in dB) comparison in the edge (1st row) and smooth (2nd row) image regions. The best PSNR of each row is shown in bold and underlined.

Image	Bilinear	AP	PCSD	SA	FS	AHD	Ours	Image	Bilinear	AP	PCSD	SA	FS	AHD	Ours
1	23.56 30.80	36.20 39.57	35.60 39.12	<u>36.66</u> <u>39.85</u>	34.63 38.61	33.56 37.39	36.12 39.56	16	28.43 33.51	40.03 44.00	<u>41.51</u> <u>45.62</u>	38.94 41.31	38.33 42.65	39.62 44.44	39.31 44.46
2	28.09 38.02	35.62 40.11	37.28 41.89	34.80 41.10	33.79 38.42	36.34 41.45	<u>37.91</u> <u>42.52</u>	17	28.14 38.59	38.33 44.38	38.15 44.59	38.09 44.60	37.86 44.40	36.34 43.33	<u>38.55</u> <u>45.33</u>
3	28.01 38.85	37.25 44.65	38.09 45.48	35.88 43.78	35.68 42.61	37.13 45.01	<u>38.86</u> <u>46.11</u>	18	25.31 32.89	34.70 39.74	34.03 39.31	<u>34.79</u> 39.54	34.19 38.76	32.07 37.56	34.66 <u>39.97</u>
4	29.09 38.38	37.27 41.81	36.92 42.16	36.49 41.69	36.40 40.16	35.97 41.56	<u>37.91</u> <u>43.06</u>	19	24.88 33.92	37.27 41.87	37.63 <u>42.57</u>	37.80 42.40	35.07 39.85	35.67 41.23	<u>38.21</u> 42.31
5	23.67 31.72	35.71 39.74	35.49 39.55	34.57 38.17	34.15 37.72	33.47 38.29	<u>36.21</u> <u>40.64</u>	20	24.65 39.08	35.73 43.05	35.88 43.20	34.61 43.31	34.60 42.10	34.45 42.45	<u>36.36</u> <u>43.94</u>
6	24.39 33.43	36.25 41.59	<u>37.12</u> <u>42.56</u>	35.62 41.50	34.87 40.07	35.20 41.31	35.80 41.53	21	24.24 34.93	<u>35.87</u> 41.81	34.91 41.45	35.31 41.67	34.78 40.34	33.27 40.21	35.61 <u>42.03</u>
7	27.95 39.01	38.23 43.70	38.24 43.80	37.38 43.23	36.59 41.74	37.12 43.40	<u>39.61</u> <u>45.50</u>	22	27.50 35.77	35.73 40.63	35.43 40.29	35.69 40.64	34.92 39.56	34.37 39.25	<u>36.63</u> <u>41.48</u>
8	20.59 27.77	33.22 37.28	33.11 <u>37.44</u>	<u>33.90</u> 37.44	30.30 35.33	31.71 36.14	33.61 37.10	23	27.82 41.52	34.49 44.65	35.99 44.60	35.00 44.36	33.91 41.69	36.35 43.93	<u>36.11</u> <u>45.95</u>
9	26.73 39.02	38.36 44.01	39.37 44.52	38.07 44.49	36.47 42.84	37.99 43.60	<u>39.83</u> <u>45.29</u>	24	23.51 32.73	31.67 39.56	31.83 39.71	31.78 39.46	<u>32.04</u> 39.05	29.90 37.81	31.72 <u>39.83</u>
10	27.24 38.69	38.61 43.91	38.97 44.33	37.97 44.42	37.53 43.47	37.40 43.46	<u>39.74</u> <u>45.23</u>	25	24.70 28.70	30.86 35.84	<u>31.82</u> <u>37.17</u>	30.69 35.78	30.28 35.40	31.74 36.67	31.20 35.78
11	25.71 35.04	36.95 42.24	36.63 42.54	36.61 42.24	35.68 40.78	34.99 41.49	<u>37.08</u> <u>42.73</u>	26	27.21 32.67	33.52 38.82	33.21 36.47	33.32 <u>39.01</u>	32.82 38.25	32.62 35.16	<u>34.13</u> 38.10
12	28.59 38.93	40.12 44.22	<u>40.72</u> 44.77	40.15 44.01	38.08 42.62	39.43 43.93	40.58 <u>45.22</u>	27	19.88 28.54	28.71 33.82	29.45 <u>34.98</u>	28.69 33.91	27.89 32.70	28.49 33.35	<u>29.75</u> 34.21
13	21.80 28.19	32.85 37.18	31.91 36.19	<u>33.38</u> 37.56	33.20 <u>37.68</u>	29.84 34.30	32.26 36.80	28	21.94 34.47	31.86 41.03	32.17 41.61	32.09 41.61	30.74 40.05	31.30 41.08	<u>32.49</u> <u>42.13</u>
14	26.47 32.21	33.96 39.10	33.62 38.89	32.87 36.90	32.04 36.03	33.33 38.64	<u>35.34</u> <u>40.47</u>	29	28.70 41.25	36.06 45.90	36.08 46.52	36.19 46.46	34.56 43.99	36.05 46.37	<u>36.94</u> <u>47.85</u>
15	27.08 35.45	35.48 41.94	35.53 42.18	34.75 40.79	33.83 40.58	34.68 41.13	<u>36.64</u> <u>42.55</u>	30	23.48 30.00	25.89 30.94	25.90 31.01	25.53 30.55	24.82 29.09	26.10 30.44	<u>27.06</u> <u>32.00</u>
Average									25.64 34.80	35.23 40.90	35.42 41.15	34.92 40.73	34.00 39.55	34.22 40.15	<u>35.87</u> <u>41.66</u>

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 need to be preserved in both directions (see Figure 3.18). 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 Table 3.4) and has much lower computational cost, as discussed in the following section.

Figure 3.17: Original cropped region from Image 27 and the results obtained by different demosaicking methods.

3.8 Discussion

To further show the advantage 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.* [43].

3.8.1 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 pixel locations requires a total of $10MN$ additions, $2.5MN$ multiplications, and $8MN$ bit-shift operations. Adaptive filtering at red and blue pixels needs $12.5MN$ 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

Figure 3.18: Comparison of the AHD and proposed demosaicking method in texture regions.

Table 3.6: Comparison of demosaicking complexity.

Method	No. of computational operations			
	Addition	Absolute	Bit-shift	Multiplication
AP	392.5MN	2MN	3.5MN	384MN
SA	32-48MN	2MN	3.5MN	12.75-21.25MN
PCSD	149MN	60MN	5MN	8MN
FS	76MN	-	48MN	21MN
AHD ³	184MN	12MN	-	49MN
Proposed	40.5MN	3MN	11MN	10.5MN

is summarized in Table 3.6.

Table 3.6 also shows the computational complexities of the five state-of-the-art methods. The complexity figures of the AP [19] and AHD [39] methods are taken directly from the corresponding papers. Although Li [41] 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 [38, 43].

It follows from Table 3.6 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 terms of visual quality, requires nearly five times the number of additions and multiplications.

According to objective measures in Table 2.3 and Table 3.4, the proposed method in Chapter 2 produces slightly better PSNR values on average. The complexity measures in Table 2.4 and Table 3.6 indicate the proposed method in Chapter 3 requires slightly lower

³We consider the comparison operation as addition one and count two iterations of median filtering as suggested by the authors.

Figure 3.19: Modified Bayer CFA patterns with (a) red and (b) blue in quincunx lattice.

computational cost. The two proposed methods are comparable under combined consideration of both performance and computational cost. Another merit of Chapter 3 is the analysis on properties of high frequency based on frequency spectrum, which is applicable to the following chapters.

3.8.2 Modified CFA patterns

Based on the frequency selection (FS) demosaicking method as described in Section 3.3, 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 3.19). The improved PSNR performance is 0.5 dB on average, as shown in the left columns of Table 3.7, which is the same as Table I in [43]. The explanation given in [43] 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.

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 [51]. Actually the higher red-green and blue-green correlations suggest that green samples share more infor-

Table 3.7: Comparison of PSNR performance (in dB) by applying FS approach [43] 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 [43]			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

mation 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 $C1$ (see Figure 3.5), but also discards some high frequencies along the horizontal and vertical axes, which are preserved in the Bayer CFA, as we discussed in Section 3.3.

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 3.7 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 the best PSNR results on average.

3.9 Conclusion

In this chapter, we have propose an adaptive filtering demosaicking method for better reconstruction of high-frequency information. First, we show by examples that high-frequency information along the horizontal and vertical axes seems more important for image quality than that in the spectrum's corners. We have also analyzed the spectrum of CFA image using sampling theory, indicating the luminance preserves more high frequency than other color components due to the smaller bandwidth of modulated color-difference components.

Furthermore, we observed that extra high-frequency information can be preserved in green locations.

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 various image contents and can better preserve high frequencies. Furthermore, due to the similarity of high-frequency information across three color components, the extracted full-resolution luminance can be used to estimate the high-frequency information of all other color components. 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 lower computational cost.

Chapter 4

Demosaicking and Compression in DSC Image Processing¹

4.1 Introduction

To push the DSC technologies to their limits, optimizing the image processing chain, rather than individual process component, is necessary for enhancing the operational/energy efficiency while maintaining the image quality. The conventional DSC image processing chain is shown in Figure 4.1(a). The CFA data from the sensor is first demosaicked and color adjusted (white balance, color correction, and gamma correction), and finally compressed for storage or transmission [7, 8].

One downside of conventional DSC processing is that CFA demosaicking does not increase the information content (entropy) of the original image, but introduces mainly redundancy [1] and consumes substantial resources of the camera. As such, one can consider an

¹This chapter results from a collaboration with Mr Lian Naixiang. I was mainly involved in model development as well as design and running of experiments.

alternative processing chain by reversing the demosaicking and compression process [1–6], as shown in Figure 4.1(b). Here the output of the CFA sensor is compressed before converting it to a full color image. This allows demosaicking and color adjustments to be done directly in the end device, rather than inside the camera.

The main advantage of the alternative processing chain is that the number of CFA samples is only 1/3 of that in the full-color image, thus requiring less computational resource, storage capacity, and/or transmission bandwidth. Recent work [1–6] has also indicated that the alternative processing chain can achieve better image quality than the conventional one under low compression ratios. Furthermore, the processing time (and hence the power consumption) can be reduced by off-loading the demosaicking process from the camera to the end device, such as a PC, PDA or printer with capability of direct printing from cameras. 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 [52]. It also allows different demosaicking methods to be applied to suit the needs of applications. For example, PC image viewers can use the most advanced and complex demosaicking methods to suit high resolution PC displays, because computational cost is not an issue in this case [53]. On the other hand, if the image is to be viewed on a PDA screen, simpler demosaicking methods usually suffice.

A disadvantage of the alternative chain is the potential complex implementation—the compression must be done directly on CFA images, and hence might need to be customized to the CFA pattern (i.e., non-rectangular green sample lattice) [4–6]. Some of the benefits, such as quality advantage over the conventional chain, are also uncertain. The existing studies [1–6] are mainly empirical, making 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.

Existing empirical studies [1–6] 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 4.2. Since one main goal of this chapter was to explain the existing empirical studies, we follow the conventional simplified framework (Figure 4.2).²

In this chapter we analyze the simplified conventional and alternative processing chains (Figure 4.2) by detailed modeling of their reconstruction errors. We first introduce the previous work and motivation in Section 4.2. Next we study the interaction between compression and demosaicking errors in Section 4.3. 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 model proposed, 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 [54] to 2D full-color and CFA images in Section 4.4. The models derived allow prediction of the compression error variances

²To justify the usefulness of our analysis based on the simplified chains, we have also considered the two processing chains in Figure 4.1(c) and (d), 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.

Figure 4.1: 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; (c) feasible conventional image processing chain; (d) feasible alternative image processing chain.

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 com-

Figure 4.2: Simplified block diagrams of DSC image processing chains: (a) conventional, (b) alternative.

pression and demosaicking errors, we compare the performances of the conventional and alternative processing chains for several compression and demosaicking methods in Section 4.5. 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) [55], this means that the alternative processing chain can be used in DSCs without loss of image quality.

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 [4, 5]. Our analysis in Section 4.6 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 4.3.1).

4.2 Previous Work and Motivation

4.2.1 Alternative processing methods

In the alternative processing chain the demosaicking and compression are reversed, as shown in Figure 4.1. As a result, the CFA samples are compressed directly, without being converted to a full-color image. Standard grayscale compression techniques can be applied directly to CFA images by discarding pixel color labels. 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 [1]. Performance can be slightly improved by using JPEG2000 compression, which is capable of achieving an acceptable performance even in the presence of artificial discontinuities [3]. It has also been suggested to compress CFA color components separately [1, 2]. However, exploiting inter-color correlations can substantially improve the compression performance.

Existing image compression standards use color transformation to decorrelate the color components [12, 13]. Similar transformation can be used to remove artificial discontinuities in CFA images, as was shown by Lee and Ortega [4], Parrein *et al.* [5], and Koh *et al.* [6]. 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 4.3. 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 4.4(a).

Each chrominance plane (U or V) can be compressed using standard techniques. But the quincunx lattice of the luminance plane requires preprocessing. Koh *et al.* [6] 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

Figure 4.3: Color transformation of a 2×2 Bayer CFA unit. Figure 4.4: Y plane and its 45° rotation.

between Y_1 and Y_2 , Lee and Ortega [4], and Parrein *et al.* [5] proposed to rotate the quincunx lattice by 45° , converting it into a diamond shaped rectangular lattice, as shown in Figure 4.4(b).

4.2.2 Motivation and notation

Koh *et al.* [6] used JPEG compression in their alternative processing chain and found that it can outperform the conventional processing chain at low ($< 10:1$) and high ($> 80:1$) compression ratios. Lee and Ortega [4] observed superiority of alternative processing with JPEG compression at low ($< 10:1$) compression ratios. Parrein *et al.* [5] used JPEG2000 and reached the same conclusion for compression ratios $< 15:1$. All these conclusions are based on empirical studies, making their results applicable to only existing compression and demo-

Figure 4.5: Cropped region of Bike image: (a) After bilinear demosaicking; (b) After bilinear demosaicking and JPEG (10:1) compression.

saicking 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 advantage 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 4.1 and illustrated in Figure 4.5. This would not happen if the errors were independent, since compression always leads to decrease in image quality. The models developed in Section 4.3.1 could help explain this phenomenon.

Table 4.1: PSNR performance (in dB) of the conventional processing chain with bilinear demosaicking and JPEG/JPEG2000 compression.

Image	After demosaicking	After demosaicking and compression			
		JPEG(10:1)	JPEG(30:1)	JPEG2000(10:1)	JPEG2000(30:1)
Bike	23.44	25.18	24.31	23.27	22.93
Lighthouse	28.01	30.04	29.17	27.87	27.65
Sails	31.69	33.59	32.97	31.58	31.38

Figure 4.6: Block diagrams of error models.

4.3 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 4.6. Let x be the original (desired) full-color image, $\bar{x}$ be the CFA image, and

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

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

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

where δx and $\delta \bar{x}$ are the compression errors of x and $\bar{x}$, while Δx is the demosaicking error. We assume the compression error to be stationary and let $\xi_s^2 \triangleq E[(\delta x_i)^2]$ and $\xi_o^2 \triangleq E[(\delta \bar{x}_i)^2]$ be the compression error variances for the full-color and CFA images respectively, where the expectation is taken over pixel locations.

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.

4.3.1 Conventional: demosaicking-compression

From Figure 4.6(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

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

Clearly, for a given compression method and bitrate, $E\{[C(x_i + \Delta x_i) - C(x_i)]^2\}$ is a function 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 depends on the compression technique and the bitrate used. Similarly, $E\{[C(x_i + \Delta x_i) - C(x_i)]\delta x\} \approx a_c E(\Delta x_i \delta x_i) = \rho_a a_c \xi_s \zeta$, where ρ_a is the correlation coefficient between two errors. Combining these approximations yields the following expression for the total error variance of the conventional processing chain:

$$\Omega^2 \approx \xi_s^2 + a_c^2 \zeta^2 + 2\rho_a a_c \xi_s \zeta \quad (4.1)$$

The model (4.1) is qualitatively supported by the experimental results shown in Figures 4.7 and 4.8. We plot the total error variance Ω^2 as a function of the compression error

Figure 4.7: Relationship between Ω^2 and ξ_s^2 for 15 test images. Each curve corresponds to one test image.

variance ξ_s^2 in Figure 4.7, and as a function of demosaicking error variance ζ^2 in Figure 4.8. 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 4.7, the observed curves are linear for sufficiently large ξ_s^2 , with a slope close to 1. This is expected, since according to (4.1), $\Omega^2 \approx \xi_s^2$ for large $\xi_s^2 \gg \zeta^2$. When ξ_s^2 is small, the other two terms in (4.1) 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 4.8 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 (4.1).

Figure 4.8: Relationship between Ω^2 and ζ^2 for 15 test images. Each curve corresponds to one test image.

The curves in Figure 4.8 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 (4.1), and the results are shown in Table 4.2. 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 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 a wide range of compression ratios 5:1 - 15:1 (JPEG) and 5:1 - 30:1 (JPEG2000), as shown in Table 4.2. We have also found that ρ_a is small, with amplitude not exceeding 0.1 on average, as shown in Table 4.2. Hence inside this range of

Table 4.2: Estimated values of a_c^2 and ρ_a in the model of conventional processing chain [(4.1)].

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

Figure 4.9: Predicted total error Ω^2 of conventional chain using simplified model (4.2).

compression ratios we can simplify the model (4.1) to the following:

$$\Omega^2 \approx \xi_s^2 + a_c^2 \zeta^2 \quad (4.2)$$

where $a_c^2 = 0.5$ for JPEG compression and $a_c^2 = 1$ for JPEG2000. Figure 4.9 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 4.9), 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 (4.2) may overestimate the total error of conventional chain,

it is sufficient to explain the phenomenon in Table 4.1. 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 4.1), since $a_c^2 \approx 1$ (Table 4.2).

4.3.2 Alternative: compression-demosaicking

From Figure 4.6(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

$$\Psi^2 = E \{ [D(\bar{x}_i + \delta\bar{x}_i) - D(\bar{x}_i)]^2 \} + E(\Delta^2 x_i) + 2E \{ [D(\bar{x}_i + \delta\bar{x}_i) - D(\bar{x}_i)] \Delta x_i \} \quad (4.3)$$

Clearly, for a given demosaicking method, $E \{ [D(\bar{x}_i + \delta\bar{x}_i) - D(\bar{x}_i)]^2 \}$ is a function of the compression error variance of CFA image, $\xi_o^2 \triangleq E(\delta^2 \bar{x}_i)$. 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 compression error and demosaicking error, and the parameter

Figure 4.10: Relationship between Ψ_o^2 and ζ^2 for 15 test images. Each curve corresponds to one test image.

$b_d^2 = E\left\{\left[\frac{\partial D(x_i)}{\partial x_i}\right]^2\right\}$ 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:

$$\Psi^2 \approx \zeta^2 + b_d^2 \xi_o^2 + 2\rho_b b_d \zeta \xi_o \quad (4.4)$$

To validate the error model (4.4), we applied alternative processing to 15 test images and the results are shown in Figures 4.10 and 4.11. The total error variance Ψ^2 is plotted as a function of the demosaicking error variance ζ^2 in Figure 4.10, and as a function of the compression error variance ξ_o^2 in Figure 4.11. Similar to Section 4.3.1, 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 4.10 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 (4.4). When the

Figure 4.11: Relationship between Ψ_o^2 and ξ_o^2 for 15 test images. Each curve corresponds to one test image.

compression error is comparable to (or larger than) the demosaicking error, the other two terms become significant and break the linearity. In Figure 4.11 the curves are also linear for large compression errors and small demosaicking errors, which agrees with the error model (4.4) because in this case $\Psi^2 \approx b^2 \xi_o^2$. For large demosaicking errors (e.g. those from bilinear demosaicking), linearity breaks due to the increasing influence of the cross term in (4.4).

We can use the curves in Figure 4.11 to estimate the parameters b_d^2 and ρ_b , shown in Table 4.3, 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

Table 4.3: Estimated values of b_d^2 and ρ_b in the error model of alternative processing chain [(4.4)].

		Bilinear	Hamilton [28]	AP [19]	IECI [56]
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

frequencies and hence have little effect on the preceding compression error $b_d^2 \approx 1$.

Similar to ρ_a , parameter ρ_b is also small, not exceeding 0.1 in amplitude on average, as shown in Table 4.3. Hence we can simplify the model (4.4) to the following:

$$\Psi^2 \approx \zeta^2 + b_d^2 \xi_o^2 \quad (4.5)$$

where $b_d^2 = 0.88$ for bilinear demosaicking and $b_d^2 = 1.1$ for other advanced ones. Figure 4.12 shows that the prediction from the simple model presents a very good fit to the experimental results.

Figure 4.12: Predicted the error of alternative chain using simple model (4.5).

4.3.3 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 (4.2) and (4.5), we obtain the following expression for the difference of total error variances

$$\Omega^2 - \Psi^2 = (\xi_s^2 - b_d^2 \xi_o^2) + (a_c^2 - 1)\zeta^2 \quad (4.6)$$

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 4.2), leading to cancelation of demosaicking error term.

According to (4.6), 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 4.5, 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.

4.4 Compression Error Model

Most popular image compression techniques, such as JPEG and JPEG2000, decompose an image into a set of subbands (or frequencies), and use variable bitrate allocation to quantize these subbands. The number of bits allocated to each subband depends on the portion of the total signal energy in this subband. For example, JPEG compression assigns more bits to lower frequencies since they contain larger portion of signal energy. The energy distribution among the subbands depends on image statistics, such as inter-pixel correlations. Hence, we can estimate the compression error by modeling the quantization error and image statistics.

Katto and Yasuda [54] derived the expression for the error variance ξ_I^2 of wavelet based compression³ corresponding to the optimal bitrate allocation. The results of their work show that if R_I is the total compression bitrate, σ_I^2 the signal variance, N the total number of wavelet coefficients, N_k the number of wavelet coefficients in the k th subband, and $\alpha_k = \frac{N_k}{N}$, then the quantization error for the image I is:

$$\xi_I^2 = \epsilon^2 \sigma_I^2 2^{-2R_I} \cdot \prod_{k=0}^{K-1} (A_k B_k)^{\alpha_k} \quad (4.7)$$

where ϵ is a constant depending on the signal, 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., $\xi_I^2 = \sum_{k=0}^{K-1} B_k \sigma_{qk}^2$. The product of $A_k B_k$ in (4.7) is closely related to the coding gain [57, 58], defined as

$$\mathcal{G}_I = \frac{1}{\prod_{k=0}^{K-1} (A_k B_k)^{\alpha_k}} \quad (4.8)$$

³The 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.

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 [57, 58] as:

$$A_k = \sum_i \sum_j h_k(i) h_k(j) \rho_{|i-j|} \quad (4.9)$$

$$B_k = \sum_i g_k^2(i) \quad (4.10)$$

The model (4.7) can be extended to images modeled by 2D Markov random processes [57–62]. In our analysis, we use a second order stationary Markov random process as the image model [9]. 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:

$$\begin{aligned} \sigma_I^2 &= E\{I^2(m, n)\} \\ \rho_{i,j} &= \frac{E\{I(m, n)I(m+i, n+j)\}}{\sigma_I^2} = \rho_0 \sqrt{i^2+j^2} \end{aligned} \quad (4.11)$$

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

The derivation of (4.7) is based on modeling the k th subband quantization error as $\sigma_{qk}^2 = \epsilon^2 \sigma_k^2 2^{-2R_k}$ [63], where R_k is the subband bitrate. However, this quantization model is valid only at high bitrates R_k [64]. To show this, in Figure 4.13 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 (4.7) to lower bitrates by allowing β to be a function of bitrate:

$$\xi_I^2 = \epsilon^2 \sigma_I^2 2^{-2\beta(R_I)R_I} G_I^{-1} \quad (4.12)$$

Figure 4.13: Average logarithm of compression error variance *versus* bitrate.

Based on the generalized error model (4.12), we can now derive the compression error models for full-color and CFA images.

4.4.1 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 (4.12), 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 quantization error of all color components is

$$\xi_s^2 = \frac{1}{3} \epsilon^2 \mathcal{G}_I^{-1} \sum_{\mathcal{C}} \sigma_c^2 2^{-2\beta(R_c)R_c} \quad (4.13)$$

A special case is the 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., $R_c =$

Table 4.4: 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

$\frac{R}{3}$, minimizes the overall compression error. Then (4.13) becomes

$$\xi_s^2 = \epsilon^2 \sigma_I^2 2^{-2\beta(\frac{R}{3})\frac{R}{3}} \mathcal{G}_I^{-1} \quad (4.14)$$

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 [12, 13]. 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

$$\frac{E\{S_1(m, n)S_2(m + i, n + j)\}}{\sigma_I^2} = \rho_{i,j} \nu_{s_1, s_2} \quad (4.15)$$

This assumption is sufficiently precise, as evidenced by the experimental results in Table 4.4.

Let v_c^T be the row of T_s that convert R,G,B to Y,U,V, $\mathcal{C} \in \{Y, U, V\}$. It follows from (4.15) that the inter-pixel correlation in a component $\mathcal{C}$ is the same as in RGB color compo-

nents:

$$\frac{E\{\mathcal{C}(m, n)\mathcal{C}(m + i, n + j)\}}{\sigma_c^2} = \frac{\rho_{i,j} v_c^T E\{S(m, n)S(m, n)^T\} v_c}{\sigma_c^2} = \rho_{i,j} \quad (4.16)$$

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 (4.12) 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 (4.12) 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 (4.9) and (4.10). The overall quantization error becomes

$$\xi_s^2 = \epsilon^2 \sigma_I^2 2^{-2\beta(\frac{R}{3})\frac{R}{3}} \prod_{\mathcal{C} \in \{Y, U, V\}} (\mathcal{A}_c \mathcal{B}_c)^{\frac{1}{3}} \prod_{k=0}^{K-1} (A_k B_k)^{\alpha_k} = \epsilon^2 \sigma_I^2 2^{-2\beta(\frac{R}{3})\frac{R}{3}} \mathcal{G}_s^{-1} \quad (4.17)$$

where $\mathcal{G}_s$ is the new coding gain. This is similar to (4.14), 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 (4.17) becomes (4.14).

4.4.2 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 (4.12) we obtain

$$\xi_o^2 = \epsilon^2 \sigma_I^2 2^{-2\beta(R)R} \prod_{k=0}^{K-1} (A_{ok} B_k)^{\alpha_k} = \epsilon^2 \sigma_I^2 2^{-2\beta(R)R} \mathcal{G}_o^{-1} \quad (4.18)$$

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 4.2.1. 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

$$\xi_t^2 = \epsilon^2 \sigma_I^2 2^{-2\beta(R)R} \prod_{\mathcal{C} \in \{Y_1, Y_2, U, V\}} \left(\mathcal{A}_c \mathcal{B}_c \prod_{k=0}^{K-1} (A_{ck} B_k)^{\alpha_k} \right)^{\frac{1}{4}} = \epsilon^2 \sigma_I^2 2^{-2\beta(R)R} \mathcal{G}_t^{-1} \quad (4.19)$$

where $\mathcal{G}_t$ is the corresponding coding gain. Different from that of a full-color image (4.16), the inter-pixel correlations in a CFA image will change under color transformation. Hence parameter A_{ck} in (4.19) depends on the color component $\mathcal{C}$, which explains the difference between the expressions of coding gains in (4.17) and (4.19).

The correlated components Y_1 and Y_2 can be more efficiently compressed as one image Y [4–6]. Since the samples of the combined image are arranged on a quincunx lattice (Figure 4.4), existing work has suggested to rotate Y by 45° , converting a quincunx lattice to a rectangular lattice with diamond shaped support [4, 5]. The formula (4.19) is still applicable if we recalculate parameter A_{ck} based on the diagonal inter-pixel correlations in a quincunx lattice.

4.4.3 Evaluation of the compression error models

Recall from Section 4.3.3 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 (4.17) and (4.18), we obtain

$$\frac{\xi_s^2}{\xi_o^2} = 2^{2\beta_r(R)R} \frac{\mathcal{G}_o}{\mathcal{G}_s} \quad (4.20)$$

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 4.14, 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 4.13) 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 [55], $\beta(\frac{R}{3}) > 1$ and this explains the observed reduced slope values.

Figure 4.14 also shows that the break-point bitrate, corresponding to $\frac{\xi_s^2}{\xi_o^2} = b_d^2 \approx 1$ (Table 4.3), 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 (4.20) as a constant, which can be estimated by fitting (4.20) to actual data at high bitrates. The curves of the predicted log-ratios are shown in Figure 4.14. Our predicted break-point for JPEG2000 is $R_b = 2.2$ bpp, which practically coincides with the actual value. For JPEG, the predicted value is $R_b = 3.6$ bpp, which is slightly smaller than the actual $R_b = 3.8$ bpp. 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 bitrates. Figure 4.14 also shows that the ratio $\frac{\xi_s^2}{\xi_o^2}$ in JPEG increases at bitrate $R < 0.8$ bpp. This happens because JPEG's $\beta(R)$ becomes too large at very low bitrates,

Figure 4.14: The ratio between the compression error variances of the full-color and CFA images.

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 [6]. However, such high compression ratios ($> 80:1$) lead to very poor image quality, and is out of the interest and scope of this chapter. This fact will be used in comparison of the two processing chains in Section 4.5.

4.5 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.

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 4.3.3. 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 .

Table 4.5: Comparison of processing chains (conventional vs. alternative).

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

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 4.5 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 (4.20), $\frac{\xi_s^2}{\xi_o^2}$ 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 4.5 agree well with the experimental results shown in Figure 4.15. In this figure we compare the signal-to-noise ratio (SNR) between 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 4.15.

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 (4.17) or (4.18). 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 (4.2) and (4.5) and plotted as a function of bitrate (Figure 4.15). The results show that for JPEG with AP or IECI demosaicking method $R_b \approx 4.5$ bpp, while 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 (4.19). The corresponding curves are shown in Figure 4.15 (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

4.5. Performance Comparison of Two Processing Chains

120

Figure 4.15: Predicted *versus* exact performance comparison between conventional and alternative processing chains.

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 [6]. 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 [5]⁴.

Figure 4.16: Performance comparison between the conventional processing chain using bi-linear and the alternative one using IECI.

In the alternative processing chain the demosaicking operation can be moved to a PC, al-

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

Figure 4.17: Comparison of the images reconstructed from a real CFA image by using *bilinear* demosaicking and JPEG2000 compression (10:1).

lowing for more advanced demosaicking methods to be used. Doing this will further increase the break-point compression ratios. Figure 4.16 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 4.17 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 4.18(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 4.18(e)].

It should be noted that even though there is a small difference between the predicted and actual error values, which is mainly due to the simplifications 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 4.3.

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.

4.6 Suitable Reference Images for Performance Evaluation

Some recent performance evaluation studies of the alternative and conventional processing chains [4, 5] 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

$$\bar{\Omega}^2 = \gamma^2 \xi_s^2 \quad (4.21)$$

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 4.3.2 to obtain the error variance

$$\bar{\Psi}_o^2 = E \{ (D[C(\bar{x})] - D(\bar{x}))^2 \} = b_d^2 \xi_o^2 \quad (4.22)$$

where b_d^2 is the same parameter as in (4.4). Performance comparison is then based on the sign of

$$\bar{\Omega}^2 - \bar{\Psi}_o^2 = \gamma^2 \xi_s^2 - b_d^2 \xi_o^2 \quad (4.23)$$

This is similar to $\xi_s^2 - b_d^2 \xi_o^2$, which is the first term in (4.6). However, the second term in (4.6), corresponding to the demosaicking error, is missing.

Using (4.23) instead of (4.6) 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 (4.6) and the main difference between (4.23) and (4.6) is in the parameter γ^2 , which may lead to a lower R_b [see Figure 4.19(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 4.5. For example, the large errors of bilinear demosaicking lead to superiority of the conventional processing chain at all compression ratios. But it follows from (4.23) that the alternative processing chain should perform better than the conventional one at low compression ratios, as shown in Figure 4.19(a)(b).

In summary, using demosaicked image as a reference leads to correct conclusions when JPEG2000 is combined with advanced demosaicking methods, because in this case $\gamma^2 \approx 1$. This explains why the results in [5] 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 [4].

4.7 Conclusions

Alternative processing chain offers advantage in DSC processing by off-loading the demosaicking operation to an end device such as a PC or PDA. However, for it to be widely adopted, the alternative processing should not lead to notable degradation in image quality. Recently reported empirical results [4–6] 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 outperform 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 [55]. 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.

(a) Test image

(b) Conventional, Bilinear

(c) Alternative, Bilinear

(d) Conventional, IECI

(e) Alternative, IECI

Figure 4.18: Comparison of the reconstructed images after demosaicking and JPEG2000 compression (10:1) of CFA data downsampled from a test image.

Figure 4.19: Relative performance of alternative and conventional processing chains for different reference images.

Chapter 5

Demosaicking and Sharpening in DSC

5.1 Introduction

To suppress the aliasing caused by CFA sampling, DSC uses an optical low-pass filter over sensor as anti-aliasing filter (AA filter). In DSC with CFA, AA filter is typically implemented using two layers of doubly-refractive crystal, which spreads each optical point into a cluster of four points [65]. AA filter works by cutting off the image frequency components higher than the Nyquist frequency, at the cost of trading-off image sharpness. To compensate for the blurring caused, usually a sharpening process is applied inside the DSC towards the end of the processing chain [8, 55].

The classical linear unsharp masking technique is a commonly used method for sharpening. In conventional unsharp masking, given an input image x , the sharpened image $S(x)$ is obtained by adding a highpass filtered and scaled version $\lambda g(x)$ to the original image as follows [9]:

$$S(x) = x + \lambda g(x), \quad (5.1)$$

where $g(x)$ denotes the extracted high-frequency component and λ is a positive scaling factor that controls the level of sharpness enhancement. A commonly-used highpass filter g is a linear function corresponding to the Laplacian mask as shown below:

$$\begin{bmatrix} h_{-1,-1} & h_{-1,0} & h_{-1,1} \\ h_{0,-1} & h_{0,0} & h_{0,1} \\ h_{1,-1} & h_{1,0} & h_{1,1} \end{bmatrix} = \begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix} / 8, \quad (5.2)$$

Many schemes have been suggested to improve the performance of unsharp masking [66–68]. These methods generally classify the image pixels into different classes, based on the local statistical property, and apply different filtering for each class of pixels.

The conventional DSC processing chain – demosaicking followed by sharpening (henceforth refer to as ‘Conventional’ method) – has one major drawback: the subsequent sharpening operation could amplify the demosaicking errors. This problem could be severe when a low-performance demosaicking method is used. Two typical examples are shown in Figure 5.1. It can be observed that the sharpened images contain more artifacts than the demosaicked images. More examples can be found in Figure 5.6 and Figure 5.7. In this chapter, we shall refer to this problem as the *demosaicking-sharpening problem*.

There are several methods proposed for solving the demosaicking-sharpening problem. In [69, 70], Komatsu *et al.* proposed to simultaneously remove the blurring in the middle of demosaicking. They modeled the problem as the inverse of the image blurring process caused by the optical lowpass filter, and employed the Landweber-type iterative approach [69] and total-variation-based super-resolution method [70] to solve the inverse problem. However, they stated that their methods are meant for a hypothetical white-red-blue (WRB) CFA (which replaces G pixels of the Bayer CFA with white pixels), and are not applicable to the popular Bayer CFA. Since this chapter focuses on the Bayer CFA, we do not

include their results for discussion or comparison.

Lukac and Plataniotis [71] proposed an alternative process chain – sharpening the raw CFA data prior to demosaicking (henceforth refer to as ‘CFAS’ method). The main advantage is that less computational resource is required for the sharpening process since the number of CFA samples is only 1/3 of the full-color image. We have analyzed this method (details shall be described in Section 5.2) and concluded that sharpening before demosaicking enhances the high-frequency information, which degrades the subsequent demosaicking performance. As illustrated in Figure 5.1, CFAS method produces noticeable visual artifacts, similar to the conventional method.

In this chapter, we propose two solutions to the demosaicking-sharpening problem. We start by modeling the reconstruction errors of the conventional and CFAS processing chains, and then analyzing the interaction between sharpening and demosaicking errors in Section 5.2. Based on our analysis, we then derive two design guidelines: avoid propagating or enlarging errors when extracting the high-frequency information, and adding high-frequency information after demosaicking to avoid degrading the demosaicking performance. Following the guidelines, we propose our methods to the problem in Section 5.3. Experimental results show that our proposed methods outperforms the existing methods, especially in cases of large demosaicking errors. The chapter is concluded in Section 5.4.

5.2 Error models

The difference between conventional and CFAS processing chains lies in the reversal of sharpening S and demosaicking D operations, as illustrated in Figure 5.2. Let x be the original (desired) full-color image, $\bar{x}$ be the raw CFA image, and the demosaicked image be $D(\bar{x}) = x + \Delta x$ with demosaicking error on CFA data Δx . We can write the result of the

Figure 5.1: Conventional versus CFAS (Bilinear and AP refer to the demosaicking algorithms used).

conventional processing chain as

$$S[D(\bar{x})] = S(x + \Delta x) \quad (5.3)$$

Let $\Delta S(x)$ be the demosaicking error on the sharpened CFA data. We can write the result of CFAS processing chain as

$$D[S(\bar{x})] = S(x) + \Delta S(x), \quad (5.4)$$

We treat the demosaicking errors in each pixel as a stationary random variable, and represent the demosaicking error variance on the raw CFA data and on the sharpened CFA data as $\zeta_r^2 \triangleq E[(\Delta x_i)^2]$ and $\zeta_a^2 \triangleq E\{[\Delta S(x_i)]^2\}$, respectively, where i designates individual pixels. In the following subsections, we shall evaluate the overall error variances of the conventional and CFAS processing chains.

Figure 5.2: Block diagrams of error models.

5.2.1 Conventional chain: demosaicking-sharpening

Given that $S(x)$ is the benchmark image, the error in the conventional method is the difference between the sharpened versions of the demosaicked and original images:

$$S[D(\bar{x})] - S(x) = S(x + \Delta x) - S(x)$$

Hence, the overall error variance is given by

$$\Omega^2 = E\{[S(x + \Delta x) - S(x)]^2\}. \quad (5.5)$$

Clearly, for a given sharpening technique, the overall error variance Ω^2 depends on $\zeta_r^2 \triangleq E[(\Delta x_i)^2]$, i.e., the demosaicking error variance of raw CFA samples. Let $f(\zeta_r^2) = E\{[S(x + \Delta x) - S(x)]^2\}$, then $f(0) = 0$. Assuming that the demosaicking error is small, we can expand $f(\zeta_r^2)$ using Taylor series and preserve only the linear term, i.e. $f(\zeta_r^2) \approx \alpha_f^2 \zeta_r^2$, where parameter $\alpha_f^2 = E\{[\frac{\partial S(x_i)}{\partial x_i}]^2\}$ in general depends on the sharpening technique. This approximation leads to the following expression for the total error variance of the conventional processing chain:

$$\Omega^2 \approx \alpha_f^2 \zeta_r^2. \quad (5.6)$$

The model (5.6) is qualitatively supported by the experimental results shown in Figure 5.3. We plot the total error variance Ω^2 as a function of the demosaicking error variance ζ_r^2 in Figure 5.3, where each point corresponds to one of the 15 test images, and the variances are estimated using the Mean Square Error (MSE) over the entire image. The relationship is discussed for various demosaicking algorithms: Bilinear (BI), AP [19], ECI [32], IECI [56], PCSD [38], AHD [39], SA [41], and FS [43]. For most of the demosaicking algorithms, a linear relationship is clearly followed.

Figure 5.3: Relationship between Ω^2 and ζ_r^2 using different demosaicking techniques for the demosaicking-sharpening chain.

The curves in Figure 5.3 can be used to estimate the parameter α_f^2 . We use standard Matlab function “polyfit” to fit the curves to model (5.6), and the results are shown in Table 5.2. In summary, our model shows that the overall error of the demosaicking-sharpening chain is proportional to the demosaicking error, with proportional constant $\alpha_f^2 > 1$. In other words, the larger the demosaicking error, the larger is the overall error; and the overall error is always greater than the demosaicking error. This clearly implies that the subsequent sharpening process amplifies the demosaicking error.

5.2.2 CFAS chain: sharpening- demosaicking

From (5.4), the error in the CFAS chain can be expressed as:

$$D[S(\bar{x})] - S(x) = \Delta S(x).$$

Hence, the overall error variance Ψ^2 is equal to the demosaicking error variance on the

sharpened CFA data, i.e., $\Psi^2 = \zeta_a^2$. As a counterpart of (5.6), Ψ^2 is formulated as a linear function of the demosaicking error on raw CFA data ζ_r^2 as follows:

$$\Psi^2 = \beta_f^2 \zeta_r^2. \quad (5.7)$$

Since $\Psi^2 = \zeta_a^2$, (5.7) can be expressed as $\zeta_a^2 = \beta_f^2 \zeta_r^2$.

We applied the CFAS chain to 15 test images, and estimate the β_f^2 values for various demosaicking algorithms. The results are listed in Table 5.2. As β_f^2 is generally larger than 1, this implies that the sharpening process before demosaicking degrades the whole performance. We will explain the reason from the model of demosaicking error in the following.

In [72], Haralabidis and Pilinis studied the demosaicking error expressions of three linear demosaicking algorithms, BI, SHTI [22], and ECI [32], and reported that the demosaicked image variance is a linear function of the original image variance. Based on their study and the principle of demosaicking algorithms, we can write the demosaicking error variance as:

$$\zeta^2 = h(\nu, \rho) \sigma_o^2, \quad (5.8)$$

where ζ^2 and σ_o^2 are variances of demosaicked and original images, ρ and ν are the inter-pixel and inter-color correlations, and $h(\nu, \rho)$ is a monotonically decreasing function of variables ρ and ν . The model in (5.8) shows that the demosaicking error is proportional to the image variance and inversely proportional to the inter-pixel and inter-color correlations.

The sharpening process adds high-frequency content to the images and modifies their statistical characteristics, and it changes the final demosaicking error variance. Table 5.1 shows the image variance and correlations after sharpening compared with the original images. It clearly shows that the sharpening process enlarges the image variance, and reduces the intra-color and inter-color correlations. As a result, it enlarges the demosaicking error

according to (5.8).

Table 5.1: Image variances and correlations after sharpening process.

	Image variance	Image correlation	
		Intra-color (ρ)	Inter-color (ν)
Original	2865.7	0.9490	0.9852
Sharpened	3252.4	0.8842	0.9693

Table 5.2: Estimated values of α_f^2 and β_f^2 .

	BI	AP	ECI	IECI	PCSD	AHD	SA	FS
α_f^2	3.52	8.86	3.41	3.22	3.60	3.40	3.29	3.24
β_f^2	3.93	13.31	13.14	19.63	12.74	13.03	9.98	9.16

It is important to note that all α_f^2 and β_f^2 values are larger than 1. This implies that sharpening applied either before or after demosaicking leads to a larger overall error. Based on the analysis presented in Section 5.2, we obtain two guidelines for solving the demosaicking-sharpening problem: (1) Avoid enlarging errors when extracting the high-frequency information for sharpening, and (2) Adding the sharpening high-frequency component after the demosaicking process so as to avoid degrading the demosaicking performance.

5.3 Proposed Solutions

5.3.1 Proposed method I

Our first proposed method extracts the high-frequency information from the CFA samples, and then adds it to the three color components after demosaicking. That is, the sharpening operation is carried out in two stages: extracting the high-frequency information and adding the high-frequency information, and the demosaicking operation is inserted in between these two stages, as illustrated in Figure 5.4.

By extracting the high-frequency information from the original CFA image samples, we can avoid enlarging the demosaicking error. Furthermore, by adding the high-frequency

information after demosaicking, we can avoid degrading the demosaicking performance. The problem is that from the CFA samples, we can only extract the high-frequency information from one color component in each pixel location. We solve this problem by adding the same high-frequency information back to three color components. This works as the high-frequency information of the three color components (R , G and B) are similar, as illustrated in Section 2.2.2.

Suppose that the color component $\mathcal{C} \in \{R, G, B\}$ consists of low- and high-frequency information, i.e., $\mathcal{C} = \mathcal{C}^l + \mathcal{C}^h$. The sharpening process can be described as $\mathcal{S}(\mathcal{C}) = \mathcal{C} + \alpha\mathcal{C}^h$, where α is a proper scalar. Assume that the high-frequency component is identical across three color components and $\hat{\mathcal{F}}$ denotes the demosaicked value of missing color $\mathcal{F}$, the desired sharpened value of $\mathcal{F}$ can be estimated as $\widetilde{\mathcal{S}(\mathcal{F})} = \hat{\mathcal{F}} + \alpha\mathcal{C}^h$.

Figure 5.4: Block diagram of the proposed method I.

In order to extract the high-frequency information from the sub-sampled CFA color planes, we need to design a band-pass filter based on the CFA spectrum. As discussed in Section 3.3, the luminance component preserves more high-frequency information than the primary color components (i.e., red, green, and blue). Therefore, we propose to extract the high-frequency information from the luminance component from the CFA spectrum directly by using a band-pass filter. The desired shape of the filter is shown in Figure 5.5(a). Two possible implementations for the desired filter are shown in Figure 5.5(b) (11×11 filter) and (d) (5×5 filter). We make the following observations from our experimental results: (a) filters with smaller sizes are preferred, because a larger size leads to a larger possibility of

edge-crossing and artifacts, (b) the simplest 5×5 implementation in Figure 5.5(e) produces reasonably good results compared with other more delicate filters. Hence, we have chosen the filter in Figure 5.5(e) for our experiments, so as to balance between the performance and computational cost. Note that applying this filter is equivalent to applying the Laplacian mask (given in (5.2)) to each of the color component after removing the missing pixels.

Figure 5.5: (a) Pass-band of proposed band-pass filter for extracting high-frequency components, two implementations: (b)(d) spectrum representation, and (c)(e) impulse response.

Given the band-pass filter H , for a green pixel, the proposed procedures are as follows:

Figure 5.6: Result of the proposed method I and other methods under comparison.

1. Extract the high-frequency information from the luminance component as below:

$$L^h(m, n) = \sum_{(k, l) \in \Lambda} \{h_{k, l} CFA(m + k, n + l)\}, \quad (5.9)$$

where Λ is the support of the filter; and h denotes the element of H .

2. Obtain the demosaicked color components: $\tilde{C}(m, n)$ where $C \in R, G, B$.
3. Derive the sharpened color values as: $S(C)(m, n) = \tilde{C}(m, n) + L^h(m, n)$.

Figure 5.6 and Figure 5.7 show the results of method I. Comparing with the results produced by CFAS and conventional methods, the proposed method does not enlarge demo-

Figure 5.7: Result of the proposed method I and other methods under comparison.

saicking artifacts. Furthermore, the proposed method produces fewer false-color artifacts even comparing with the demosaicking results. This is because adding the same high-frequency information to the three color components makes the color values more consistent, and thus reduces false colors. Moreover, the proposed method processes the same number of samples as in sharpening step of the CFAS method, i.e., $\frac{1}{3}$ of that processed in the conventional method.

5.3.2 Proposed method II

In this section, we propose another method to fully exploit the available high-frequency information in the CFA samples. As discussed in Section 3.3, the high-frequency information along the horizontal and vertical axes in the sub-sampled green component will not be as much aliased as the other components. This aliasing-free high-frequency information also exists in the luminance signal at the green pixel locations. However, in the proposed method I, we filter out this part of information by using filters as shown in Figure 5.5.

Figure 5.8: Block diagram of proposed II solution.

Hence, we propose another solution which performs demosaicking and sharpening simultaneously to make better use of the available high-frequency information. We first recover the full-resolution luminance component with better preserved high-frequency information. We then bilinearly interpolate the chrominance components and sharpening the luminance component. The proposed method is illustrated in Figure 5.8. It consists of the following steps:

1. Extract the luminance component at green pixel locations using filter $H1$ given in Figure 3.6(c):

$$L_g(m, n) = \sum_{(k,l) \in \Lambda} \{h1_{k,l} CFA(m+k, n+l)\}, \quad (5.10)$$

where Λ is the support of the filter, $h1$ denotes the element of $H1$, and CFA denotes the CFA image.

2. Extract the luminance component for red sample locations using filter $H2$ given in Figure 3.7(c):

$$\begin{aligned} \hat{R}_g &= \mathcal{I}_{bil} R_s \\ Cr_g &= L_g - \hat{R}_g \\ L_r(m, n) &= \sum_{(k,l) \in \Lambda} \{h2_{k,l} Cr(m+k, n+l)\} + R(m, n), \end{aligned} \quad (5.11)$$

where $\mathcal{I}_{bil}$ denotes bilinear interpolation, R_s is the sub-sampled version of R component, and Cr designates the color-difference image (i.e., chrominance component).

Iterate the process once for better estimation.

Apply the same process to the blue sample locations.

3. Sharpen the recovered luminance component using the unsharp masking method.
4. Interpolate the chrominance components using bilinear method.
5. Restore the R , G , and B components by adding the interpolated chrominance to the luminance component.

The two filters used were described in Section 3.4 and Section 3.5, which have the property that the resultant luminance keeps more high-frequency information than the general demosaicked color components.

The following lemma explains why this approach works.

Lemma 2. *Let $\mathcal{L}$ be a full-resolution luminance component. The demosaicked and sharpened value of any full-resolution color component $\mathcal{C} \in \{R, G, B\}$ can be obtained from its sub-sampled version $\mathcal{C}_s$ using*

$$S(\mathcal{C}) \approx \mathcal{I}(\mathcal{C}_s - \mathcal{F}_s) + S(\mathcal{L}) \quad (5.12)$$

where $\mathcal{L}_s$ is a sub-sampled version of $\mathcal{L}$, and $\mathcal{I}$ denotes a proper interpolation process.

Proof: Let $S(\mathcal{L}) = (\mathcal{L}^l + \mathcal{L}^h) + \mathcal{L}^h = \mathcal{L}^l + 2\mathcal{L}^h$, where $\mathcal{L}^l$ and $\mathcal{L}^h$ are the low-and high-frequency components of $\mathcal{L}$, respectively. Let $\mathcal{C} = \mathcal{C}^l + \mathcal{C}^h$. Assume that the inter-color high-frequency components are similar. We have $\mathcal{L}^h \approx \mathcal{C}^h$. As a result, the difference image $\mathcal{C} - \mathcal{L} \approx \mathcal{C}^l - \mathcal{L}^l$ contains mainly the low-frequency information. Hence, it can be accurately interpolated from its sub-sampled versions, i.e., $\mathcal{C}^l - \mathcal{L}^l \approx \mathcal{I}(\mathcal{C}_s^l - \mathcal{L}_s^l) \approx \mathcal{I}(\mathcal{C}_s - \mathcal{L}_s)$. It follows that

$$\mathcal{I}(\mathcal{C}_s - \mathcal{L}_s) + S(\mathcal{L}) \approx \mathcal{C}^l - \mathcal{L}^l + \mathcal{L}^l + 2\mathcal{L}^h \approx S(\mathcal{C}) \quad (5.13)$$

■

Figure 5.9 and Figure 5.10 present the experimental results of our proposed method compared with 2 existing methods. Adaptive demosaicking algorithm in [73] is used for the conventional and CFAS chain. Figure 5.9 shows that the our proposed method II produces fewer false-color artifacts than that of CFAS. Moreover, the CFAS result as shown in Figure 5.10 has over-sharpened edges that are thicker than the natural ones. From the experiments, we

Figure 5.9: Result of the proposed method II and other methods under comparison.

conclude that our proposed method II is better than CFAS and close to the conventional. The computational complexity of CFAS and proposed method II are similar, which is about $\frac{1}{3}$ of the conventional chain.

Figure 5.11 presents the performance comparison of two proposed methods with existing methods. The proposed method I aims to optimize the processing chain of sharpening and demosaicking. It is applicable for processing chain with any demosaicking methods. The proposed method II performs sharpening and AFD demosaicking technique simultaneously. The proposed method I improves the performance more obviously in the cases using low-end demosaicking techniques, while the improvement is little when high-performance demosaicking technique is applied. Both two proposed methods require $\frac{1}{3}$ computational cost in sharpening operation.

Figure 5.10: Result of the proposed method II and other methods under comparison.

Figure 5.11: Result of the proposed methods I and II and other methods under comparison.

5.4 Conclusion

In this chapter, we have provided a better insight into the existing processing chains of demosaicking-sharpening problem by developing analytical models for the reconstruction error variances. Based on the analysis on the interaction between the demosaicking and sharpening errors, we deduce two guidelines for addressing the demosaicking-sharpening problem. First, we avoid propagating or enlarging errors when extracting the high-frequency information for sharpening; and it is better to extract high-frequency information directly from the original CFA samples when the demosaicked color values contain large errors. Second, we add the high-frequency information after demosaicking process to avoid degrading demosaicking performance. Two methods have been proposed based on the guidelines. Experimental results show that our proposed methods can outperform the existing ones.

Chapter 6

Heuristic Solution of Rainbow Effect Removal

6.1 Introduction

In analog color TV broadcasting, a *composite* video signal is transmitted, which combines a luminance component Y and a chrominance (or chroma) component C [18, 74]. During decoding, visual artifacts appear if some of the C signal remains in the reconstructed Y and vice versa [17, 18, 75]. Remaining C in Y causes dots crawling up the vertical edge of the image, known as *chroma crawl* or *dot crawl*. Remaining Y in C causes rainbow-like color strips superimposed on what should be a monochromatic image, and the artifact is known as *rainbow effect*, the focus of this chapter.

Rainbow-effect presents itself as interlaced color stripes, usually in purple-green or orange-blue, and appears in regions of high luminance frequency and intensity, such as areas with white-strip pattern. Figure 6.1 shows a video frame (from a NTSC composite video source) contaminated with rainbow effect, where purple-green interlaced color strips are vis-

ible on the surface of the white tennis racket. Rainbow-effect is highly visible because the interlaced color pair sits on the opposite ends of the Munsell's color wheel, resulting in high color contrast. In motion pictures, this artifact shows up as rolling color strips and is visually annoying.

Figure 6.1: An example of rainbow-effect artifact on a captured video frame: (a) the artifact frame, (b) the enlarged artifact region.

There exist some methods for removing rainbow-effect artifacts from the captured video frames by using specialized digital filters during post-processing. For example, an AviSynth plug-in called GuavaComb filter was proposed in [17]. GuavaComb filter exploits the property that the erroneous chroma components between two neighboring frames are inverse in phase, and operates by simply averaging the chroma of the two frames. Our tests on GuavaComb filter show that it has several limitations: it only removes rainbow-effect artifact in static scenes and does not work well when large motion exists. It requires neighboring frames for processing, thus, is not feasible for single-frame restoration. Another method [76] for removing rainbow-effect artifacts has been implemented as a VirtualDub plug-in called Smart Smoother IQ (SMIQ) filter. SMIQ filter applies spatial filtering in chrominance components to remove the abrupt changes. Our tests, however, show that it cannot completely remove all the artifacts and it introduces blurred edges in uncontaminated regions.

Applying spatial or temporal filter (such as SMIQ and GuavaComb filters) to the entire

video frame to remove rainbow effect does not yield satisfactory results. Temporal filtering does not work when large motion exists, while indiscriminate spatial filtering leads to blurred edges. It is observed that the rainbow-effect only contaminates the chrominance components in local regions. Thus it is desirable to keep the intact regions of the images out of processing, as the restoration will introduce distortions to these regions.

In this chapter, we propose a novel method to detect and remove rainbow-effect artifacts. Our detection method exploits statistical features in the luminance and chrominance components to identify the artifact regions. Two types of statistical features are used: (a) moments of error prediction, and (b) third moment of chrominance characteristic function. Once the artifact regions are identified, we suppress the rainbow-effect artifacts by removing large wavelet coefficients in the high-frequency subbands of the regions. Experimental results on both captured video frames and simulated images show that our method outperforms the existing methods; it not only removes the rainbow-effect artifacts effectively, but also retains the image quality notably.

This chapter is organized as follows. Section 6.2 describes the procedure for detecting the artifact regions, while Section 6.3 provides the operations for removing the rainbow-effect artifacts. Section 6.4 presents the experimental results and Section 6.5 concludes the chapter.

6.2 Artifact Detection

In this section, we propose a novel method to identify and isolate the rainbow-effect contaminated region.

We start by segmenting the video frame into non-overlapping homogenous regions. The purpose is to group the contaminated pixels into connected regions, which are well separated from the uncontaminated regions. We adopt Deng and Manjunath's unsupervised technique

for segmenting color images into regions [77]. This technique is capable of segmenting a color image into regions of homogeneous color textures, with scales adjustable by parameters. It operates by first quantizing the colors into several representative classes, and then classifying the image pixels into regions according to their spatial distribution. The initial segmentation results are then merged based on their color similarity. The final segmentation result is heavily influenced by the color quantization threshold – too large a threshold resulting in merged neighboring regions with similar color while too small a threshold degrading the regions' homogeneity. From our analysis, rainbow-effect contaminated region possesses high-frequency contents and is rich in colors. Coarse segmentation is hence preferred. By adjusting the thresholds for color quantization and region merging, we are able to group the contaminated pixels into connected regions well separated from the uncontaminated areas, as illustrated in Figure 6.2. In our experiments, the thresholds for color quantization and region merging are set as 150 and 0.1, respectively.

Figure 6.2: Example of segmentation result: (a) segmented regions and (b) the artifact region.

Next, we classify the segmented regions into artifact (i.e., contaminated) or intact (i.e., uncontaminated) regions. This is accomplished by identifying one or more discriminating features of the two classes. The following two subsections will describe the statistical features used in our proposed method.

6.2.1 First and second moments of prediction errors

It is observed that rainbow-effect artifact usually occurs in areas of high luminance frequencies (thus smaller correlation between neighboring Y values) and the resultant chrominance artifact signals are rainbow-like (thus smaller correlation in neighboring I and Q values). On the contrary, the components Y , I , and Q in a homogenous intact region generally change slightly and smoothly from one pixel to the next. We studied the correlation between neighboring pixels of the Y , I , and Q component signals. Table 6.1 lists the average correlation coefficients of the neighboring pixels in the artifact and the intact regions from five captured video frames. It shows that intact regions possess a higher correlation than the artifact regions. This is a candidate feature for discriminating artifact regions from intact regions.

Table 6.1: Average correlation coefficients of neighboring pixels of the component signals in artifact and intact regions.

	Intact Region			Artifact Region		
	ρ of Y	ρ of I	ρ of Q	ρ of Y	ρ of I	ρ of Q
1	0.93	0.97	0.93	0.79	0.78	0.74
2	0.98	0.99	0.98	0.84	0.95	0.94
3	0.96	0.98	0.97	0.92	0.95	0.92
4	0.93	0.97	0.93	0.75	0.91	0.90
5	0.92	0.96	0.95	0.64	0.94	0.92
Abs. AVG	0.94	0.97	0.95	0.78	0.90	0.88

While the correlation of neighboring pixels could be a possible discriminating feature for the artifact and intact regions, the challenge lies in how to calculate the correlation for these odd-shaped segmented regions. Instead of using the correlation directly, we propose to predict the component values of a pixel based on its neighboring pixels, and use the prediction errors as an indication of correlation.

Consider a pixel at location (m, n) ; its luminance value $Y(m, n)$ can be predicted from

its eight neighbors using a linear predictor as follows:

$$\begin{aligned}
 \tilde{Y}(m, n) = & w_1 Y(m-1, n-1) + w_2 Y(m, n-1) \\
 & + w_3 Y(m+1, n-1) + w_4 Y(m-1, n) \\
 & + w_5 Y(m+1, n) + w_6 Y(m-1, n+1) \\
 & + w_7 Y(m, n+1) + w_8 Y(m+1, n+1),
 \end{aligned} \tag{6.1}$$

where w_k 's are some appropriate weights.

This linear predictor can be expressed more compactly in a matrix form as

$$\tilde{\mathbf{Y}} = P\mathbf{w}, \tag{6.2}$$

where $\mathbf{w} = (w_1, w_2, \dots, w_8)^T$ is a column vector of the weights; vector $\tilde{\mathbf{Y}}$ contains the magnitudes of $\tilde{Y}(m, n)$ strung out into a column vector; and the rows of the matrix P correspond to the neighboring pixel values. The weights can be obtained using the least squares method [78] as follows

$$\mathbf{w} = (P^T P)^{-1} P^T \tilde{\mathbf{Y}}. \tag{6.3}$$

The errors between the actual Y values and the predicted ones are given by

$$\mathbf{e} = |\mathbf{Y} - P\mathbf{w}|. \tag{6.4}$$

The prediction errors for the two chrominance components I and Q can be similarly computed.

We then obtain the first order and the second order statistics, i.e., the mean and the variance, from the prediction errors of the Y , I and Q components, and combine them into a 6-D

feature vector for each of the segmented regions.

In order to perform classification, we reduce the dimension of the feature vector space via principal component analysis (PCA) [79]. Let $\mathbf{x}_i \in R^6$, $i = 1, \dots, N$ denotes the 6-D feature vector of the i th segmented region, where N is the total number of regions. The mean of the feature vector is computed as $\mu = \frac{1}{N} \sum_{i=1}^N \mathbf{x}_i$. The feature vectors are then packed into a $6 \times N$ matrix $\mathbf{M} = [\mathbf{x}_1 - \mu, \mathbf{x}_2 - \mu, \dots, \mathbf{x}_N - \mu]$. In general, $N > 6$. Hence, we can compute the 6×6 covariance matrix as $\mathbf{C} = \mathbf{M}\mathbf{M}^T$. The principal components $\mathbf{e}_j$ can be obtained as the eigenvectors of the covariance matrix. Projecting the feature vectors into the first principal component as $\mathbf{p} = \mathbf{e}_1\mathbf{x}$, we obtain a $1 \times N$ vector $\mathbf{p}$ where each value represents the discriminating feature of a region.

Figure 6.3 shows the distribution of the feature values of 4 test video frames. The artifact regions generally possess larger feature values than most of the intact regions. A few intact regions also produce large feature values and could be classified as artifact regions. Considering the trade-off between effectiveness and computational cost, we decide to adopt this feature as our primary classification criterion, and deal with the misclassification of intact regions later.

Since the projected values vary for different frames, the Fisher linear discrimination (FLD) technique is used to adaptively thresholding the regions. Given that there are two classes: artifact and intact, the threshold is chosen as the one maximizing Fisher discriminant [80] as follows:

$$J(T) = \frac{|\mu_1 - \mu_2|^2}{S_1^2 + S_2^2}, \quad (6.5)$$

where μ_1 , μ_2 and S_1^2 , S_2^2 denote the mean and the variance of the two classes, respectively. It is well-known that this threshold maximizes the inter-class means while minimizes the intra-class variances. In our implementation, T can vary from the mean to the maximum

value of the projected vector $\mathbf{p}$. The specific T that maximizes $J(T)$ is then selected as the threshold. The classification results are shown in Figure 6.3. In (a) and (b), the artifact and intact regions are clearly separated. In (c) and (d), a few intact regions are misclassified as artifact regions. All the artifact regions, however, are correctly detected.

Figure 6.3: Classification results using PCA projection and FLD thresholding of 4 test video frames.

We shall now deal with the few misclassified intact regions. Further analysis reveals that these regions contain high frequency contents in both luminance and chrominance. This finding agrees with the selection of the feature vector (i.e., discriminating classes based on the predictability of one pixel by its neighboring pixels). To exclude these as artifact regions,

we exploit other characteristics of the artifact regions, as to be described in Section 6.2.2.

It should be noted that incursion of a few additional intact regions in the artifact-removal process does not pose a big problem (some side effects introduced in these regions). The reverse, however, would result in obvious annoying artifacts as they are not processed and removed. Thus, the second criterion is optional and can be skipped to tradeoff performance against the computational cost.

6.2.2 Third moment of the chrominance

It is observed that the histogram of chrominance in the affected region is generally symmetrical about its mean due to the periodic pattern. Figure 6.4 shows an example of the chrominance histograms of an artifact region and a typical high-frequency intact region.

Figure 6.4: Chrominance histogram of (a) an artifact region, and (b) a typical intact region.

We shall use the third order moment to measure the skewness (or the symmetry) of the chrominance histogram. Let $h(r_k) = n_k, k = 1, 2, \dots, L$ denote the histogram of the chrominance component in a region, where r_k is the k th level in the chrominance intensity range, L is the maximum level, and n_k is the number of pixels in the region whose chrominance level is r_k . Normalized histogram can be obtained simply by dividing all the $h(r_k)$ by the total number of pixels in the region, denoted as N , i.e., $p(r_k) = \frac{h(r_k)}{N}$. Let m denote the mean

value. The l th moment is given by

$$\mu_l = \sum_{i=1}^L (r_i - m)^l p(r_i). \quad (6.6)$$

The third order moment measures the skewness of the histogram: zero indicates symmetrical; a positive value indicates skewing to the right (about the mean), and a negative value indicates skewing to the left. Table 6.2 compares the skewness measures of artifact regions and intact regions. It shows that the absolute skewness values of intact regions are generally larger than those of artifact regions. In our implementation, we classify a region as intact if the absolute skewness is greater than 0.5 for both of the two chrominance components. Note that our study shows that this discriminating feature is not as effective as the prediction errors described in the previous section. It, thus, cannot be used as the primary discriminating feature but could be used as a supplementary feature to verify the artifact regions detected by prediction errors.

Table 6.2: Third moment of chrominance histogram for artifact and intact regions.

	Intact Region		Artifact Region	
	μ_3 of I	μ_3 of Q	μ_3 of I	μ_3 of Q
1	0.26	-0.53	0.27	-0.17
2	1.18	0.60	-0.03	-0.21
3	0.71	0.48	0.34	-0.09
4	-0.50	0.42	-0.04	0.01
5	0.40	-0.63	-0.12	0.15
Abs. AVG	0.61	0.53	0.16	0.13

6.3 Artifact Removal

Once the rainbow-effect contaminated regions are identified and isolated, we can proceed to remove the artifacts from the contaminated chrominance components in these regions. From the generating principle of rainbow-effect artifact we know that it contaminates only the chrominance components in the local regions. Therefore, we shall restore only the chromi-

nance values in the detected regions and leave the luminance component intact to avoid further degradation.

As indicated in Section 2.2.2, the high-frequency information of a color image is similar across color components. As a result, the high-frequency contents in color differences $R - G$ and $B - G$ diminish. Since the chrominance components I and Q are derived from the color differences, we expect similar diminishing high-frequency contents. On the other hand, the erroneous terms causing rainbow-effect artifacts usually contain large coefficients in the high-frequency subbands as illustrated in Figure 6.5.

Figure 6.5: Wavelet decomposition of a test image (rainbow-effect artifact region are circled).

A typical rainbow-effect contaminated region is overshadowed by vertically interlaced color stripes. As a result, the HL subband of the artifact region is dominated by artifact coefficients. The wavelet coefficients in the HL and the HH subbands are generally small and the energy is mainly concentrated in the LH subband at coarse scale.

To preserve the useful edge information in the LH subband at coarse scale, we adopt a wavelet thresholding method [81] to remove the artifacts in this subband. In wavelet denoising, wavelet coefficients are roughly classified into two categories, small or large [81]. Small coefficients correspond to smooth regions of the image, while large coefficients occur in the edge regions. The main idea of wavelet denoising is to modify the coefficients according to this classification. Since the luminance component is intact in the rainbow-effect contaminated regions, it can serve as a reference for classification. We exploit the cross-scale correlations among the detail wavelet coefficients for this purpose.

Assume that W_s^{LH} is the wavelet coefficient of the luminance in the LH subband at scale s , the state maps can be generated by directly multiplying cross-scale wavelet coefficients as follows:

$$E_s(m, n) = W_s^{LH}(m, n)W_{s+1}^{LH}(m, n). \quad (6.7)$$

Note that the multiplication is done on the wavelet coefficients without down-sampling.

The state of each pixel is decided by thresholding the map E_s :

$$q(m, n) = \begin{cases} 1 & \text{if } |E_s(m, n)| > T \\ 0 & \text{otherwise} \end{cases} \quad (6.8)$$

We use a model for wavelet coefficient with mixing components that are Gaussian [82] with the probability density function as:

$$f(\tilde{x}|q) = \frac{1}{\sqrt{2\pi\sigma_q^2}} \exp - \frac{\tilde{x}^2}{2\sigma_q^2}, \quad (6.9)$$

where $\tilde{x}$ denotes the noisy wavelet coefficient.

Given a state q , the minimum mean square error (MMSE) estimate of the wavelet coeffi-

cient x is given by:

$$\hat{x} = \frac{\hat{\sigma}_q^2}{\hat{\sigma}_q^2 + \hat{\sigma}_n^2} \tilde{x}. \quad (6.10)$$

where $\hat{\sigma}_q^2$, $q = 0, 1$ are estimates of the signal variances, and $\hat{\sigma}_n^2$ is an estimate of the artifact variance. In our implementation, $\hat{\sigma}_q^2$ is estimated with the square of local mean value $E(\tilde{x})$, while $\hat{\sigma}_n^2$ is estimated by $[\tilde{x} - E(\tilde{x})]^2$.

For the HL and HH subbands, we set the detail wavelet coefficients to zero in the artifact region.

In summary, our proposed rainbow-effect detection and removal method consists of the following steps:

1. Segmentation

Use Deng and Manjunath's unsupervised segmentation method [77] to segment the video frame into non-overlapping homogenous regions, so as to group the contaminated pixels into connected regions.

2. Artifact Region Detection

- a) For each region, construct a 6-D feature vector consisting of the means and variances of the prediction errors of Y , I , and Q component signals, using (6.1) – (6.4).
- b) Project the 6-D feature vectors onto a 1-D subspace defined by the first PCA projection axis. The projection value of each region is thresholded to decide the region class. The threshold is set as the value maximizing the Fisher discriminant defined in (6.5).
- c) For each of the identified artifact regions in step (b), calculate the third moment μ_3 of the chrominance components according to (6.6). Set the regions with $|\mu_3| > 0.5$ for both I and Q as intact regions.

3. Artifact Removal

The LH wavelet subbands for the chrominance components are restored via wavelet thresholding. The state of each pixel is determined based on the cross-scale correlation with the intact luminance component. For the other high-frequency subbands HL and HH, the coefficients are set to zero.

6.4 Experimental Results

In this section, we present the rainbow-effect detection and removal results of our proposed method and that of two existing filtering methods. The schemes used for comparison include the traditional 2-D Wiener denoising filter ¹ and the SMIQ filter [76] ².

Figure 6.6 and Figure 6.7 show the detected artifact regions using our proposed artifact-detection method, and the restoration results of our proposed method compared with the 2-D Wiener filter and the SMIQ filter on two video frames obtained from composite video source. Figure 6.8 and Figure 6.9 show the results on two simulated video frames (obtained by artificially introducing rainbow-effect artifacts on the original digital images shown). Firstly, our artifact-detection method is effective in identifying and isolating rainbow-effect contaminated regions. Secondly, the restoration results show that the 2D Wiener filter does not work well in removing rainbow-effect artifacts, and the SMIQ filter produces residue. All in all, our proposed method is effective for the captured video frames as well as the simulated ones.

Table 6.3 lists the peak signal-to-noise ratio (PSNR) values of the simulated frames and the restored frames (compared with the original images) using our proposed method, 2-D Wiener filter and SMIQ filter. Although all the methods show an improved PSNR compared

¹We use the MATLAB function 'wiener2'.

²We use the default optimal parameters: diameter = 11 and strength = 200.

with the degraded version, our proposed method clearly outperforms the others.

Figure 6.6: Results for a captured video frame: (a) the captured frame, (b) the detected artifact region, and (c)–(e) restoration results.

6.5 Conclusion

In this chapter, we propose a novel method to restore rainbow-effect contaminated video frames. Based on the observed distinct characteristics from artifact regions, we propose a method to detect the contaminated regions and then remove the artifacts. Our detection method exploits features in the component signals (luminance and chrominance) to discriminate artifact regions from intact regions. Two statistical features are used: (1) the first and the second moments of prediction errors, and (2) the third moment of the chrominance

Figure 6.7: Results for a captured video frame: (a) the captured frame, (b) the detected artifact region, and (c)–(e) restoration results.

histograms. Once the artifact regions are identified and isolated, we suppress the artifacts locally via wavelet thresholding. Experiments conducted on captured video frames as well as simulated frames show that our method is effective in detecting artifact region and restoring the contaminated region. Our method also improves the image quality quantitatively on the simulated frames in terms of PSNR. Comparing with the traditional 2-D Wiener filter and the SMIQ filter, our method is clearly superior.

Figure 6.8: Results for a simulated video frame: (a) the original image, (b) the simulated frame, (c) the detected artifact region, and (d)–(f) restoration results.

Table 6.3: Comparison of PSNR results.

Image		Whole Frame			Artifact Region		
		R	G	B	R	G	B
Barbara	Degraded	33.08	37.32	29.69	31.31	34.98	26.45
	2D Wiener	34.07	38.29	30.80	32.05	35.52	27.28
	SMIQ	31.87	38.93	32.90	32.93	38.79	32.86
	Proposed	34.05	38.79	31.68	34.09	39.12	33.52
Starburst	Degraded	26.96	29.25	23.11	21.83	24.30	18.28
	2D Wiener	28.28	29.90	25.17	22.25	24.53	18.82
	SMIQ	29.86	30.76	27.66	24.09	25.81	20.97
	Proposed	30.16	30.85	27.84	27.12	27.58	25.66

Figure 6.9: Results of a simulated video frame: (a) the original image, (b) the simulated frame, (c) the detected artifact region, and (d)–(f) restoration results.

Chapter 7

Analysis, Detection, and Suppression of Rainbow-Effect

7.1 Introduction

Existing methods for removing rainbow-effect are mostly heuristic. To develop a principled solution, there is a need to formulate the cause of rainbow-effect artifacts in greater details. We are not aware of any theoretical or systematic work that analyzes and removes this intriguing artifacts from the signal processing standpoint.

In this chapter, we begin by deriving the formulas from the first principle governing the rainbow-effect artifact. According to these formulas, we noted that rainbow-effect artifact contaminates a local region. Hence, it is desirable to separate the contaminated and intact regions, as global restoration will inevitably introduce distortion to the intact region. Based on this motivation and the formulas, we propose a novel method to detect the contaminated regions and subsequently remove the artifacts in these regions. Our detection method exploits discriminating features in the component signals to identify the artifact regions. Once

the artifact regions are localized, we suppress the artifacts by modifying the large fine-scale wavelet coefficients of the chrominance components. We further extend the frame-based restoration technique to video sequences by exploiting inter-frame correlation. Experimental results on both captured and simulated video frames show that our method outperforms all the existing methods. It not only removes the rainbow-effect artifacts effectively, but also notably improves the visual quality.

This chapter is organized as follows. In Section 7.2, we derive the formulas governing the rainbow effect. Section 7.3 describes the procedure for detecting the artifact regions. Section 7.4 presents the operations for removing the artifacts in video frame. Section 7.5 extends the frame-based removal processes to video sequences. Section 7.6 presents the experimental results and Section 7.7 concludes the chapter.

7.2 Formulation of Rainbow-Effect Artifact

In analog color TV broadcasting, the color video components (as well as the audio component) are multiplexed and modulated into a single *composite* signal for transmission [18, 74]. This process involves a few steps. Firstly, the three primary colors R , G and B are transformed into one luminance component Y and two chrominance components U and V as follows:

$$\begin{aligned} Y &= 0.299R + 0.587G + 0.114B \\ U &= B - Y \\ V &= R - Y \end{aligned} \tag{7.1}$$

PAL system uses Y , U , V components directly. NTSC system uses Y , I , Q instead, where

the two chrominance components I and Q are obtained by combining scaled and rotated U and V . This serves to align I to orange-cyan axis, and Q to the green-purple axis [15]:

$$\begin{aligned} I &= -0.545U + 0.839V \\ Q &= 0.839U + 0.545V \end{aligned} \quad (7.2)$$

Next, the two chrominance components are modulated onto a color subcarrier of frequency f_c and combined into a single chroma signal $C(t)$ using the so-called quadrature amplitude modulation (QAM) as follows¹:

$$C(t) = I(t)\cos(f_ct) + Q(t)\sin(f_ct) \quad (7.3)$$

The baseband composite video signal, $S(t)$, is obtained by summing the luminance $Y(t)$ and the chroma $C(t)$:

$$S(t) = Y(t) + C(t). \quad (7.4)$$

Finally, the composite signal is modulated onto a picture carrier of frequency f_y using vestigial sideband modulation (VSB). The frequency of the picture carrier f_y depends on the broadcasting channel used, which is much larger than the color subcarrier f_c .

It is important to note that the composite video signal is *one-dimensional*. This 1-D signal is raster-scanned to form the 2-D images on the TV monitors. To analyze the rainbow-effect artifact, it is sufficient to investigate the composite signal and its components along a horizontal scan line, as rainbow-effect artifact is caused by the interference between the

¹On PAL system, the phase of the V signal is reversed from line to line. The difference is ignored here since we are only concerned with signals in a scan line.

signal components. In the following analysis, $S(t)$, $Y(t)$, $C(t)$, $I(t)$ and $Q(t)$ denote signals of a horizontal scan line.

At the receiver, the composite signal $S(t)$ is first demodulated to the baseband. The luminance $Y(t)$ and the chroma $C(t)$ are then separated. The decoder, at times, cannot achieve a perfect separation, leaving parts of the $Y(t)$ signal in $C(t)$ and vice versa. Assume that $\Delta Y(t)$ denotes the residual $Y(t)$ signal in chroma $C(t)$, the decoded chroma signal $\tilde{C}(t)$ can be expressed as

$$\tilde{C}(t) = \Delta Y(t) + I(t)\cos(f_c t) + Q(t)\sin(f_c t) \quad (7.5)$$

To extract $I(t)$, $\tilde{C}(t)$ is multiplied by $2\cos(f_c t)$, which gives:

$$2\tilde{C}(t)\cos(f_c t) = I(t) + I(t)\cos(2f_c t) + Q(t)\sin(2f_c t) + 2\Delta Y(t)\cos(f_c t) \quad (7.6)$$

A lowpass filter is then applied to the resultant signal to remove the high-frequency terms and recover $I(t)$. The pass band of the lowpass filter is typically set to the bandwidth of the chrominance components, i.e., 1.5 MHz. Since $f_c = 3.58$ MHz in NTSC and $f_c = 4.43$ MHz in PAL, the middle two terms can be readily removed by the lowpass filter. The last term, however, requires further analysis. Since our interest is in the digitalized videos, we approximate the continuous signal $\Delta Y(t)$ using N-point Discrete Cosine Transform (DCT) to carry out frequency analysis. The last term can therefore be expressed as:

$$2\Delta Y(t)\cos(f_c t) = \sum_{n=0}^{N-1} [\Delta Y_n \cos(f_n + f_c)t + \Delta Y_n \cos(f_n - f_c)t] \quad (7.7)$$

where f_n denotes the n -th frequency component and ΔY_n denotes its magnitude.

The above expression shows that if $|f_n - f_c|$ is smaller than the pass-band of the lowpass

filter (1.5 MHz) and the magnitude of ΔY_n is sufficiently large, the term $\Delta Y_n \cos(f_n - f_c)t$ will remain in the recovered $I(t)$ signal, resulting in visual artifacts. Since the value of the color subcarrier frequency f_c is fixed and defined in the NTSC/PAL specification, we can deduce that $2.08 \text{ MHz} < f_n < 5.08 \text{ MHz}$ (NTSC) and $2.93 \text{ MHz} < f_n < 5.93 \text{ MHz}$ (PAL) are the ranges of luminance frequencies that may trigger visual artifacts. As $Y(t)$ is meant to be band-limited to 4.2 MHz (NTSC) and 5.5 MHz (PAL), the above ranges correspond to the high-frequency luminance. Furthermore, since the magnitude of ΔY_n has to be sufficiently large to leave residue in chrominance, we can infer that rainbow-effect artifacts are more noticeable in areas of large luminance intensity. Combining the findings, we conclude that the rainbow-effect artifacts are most apparent in regions of *high luminance intensity and frequency*, such as areas with white strips. The location where rainbow-effect artifacts are observed in Figure 6.1 clearly validates this conclusion. The characteristics of the triggering luminance are summarized in the following property:

Property 1: The triggering luminance (a) falls in the high frequencies range of $[2.08, 5.08] \text{ MHz}$ (for NTSC) and $[2.93, 5.93] \text{ MHz}$ (for PAL), and (b) possesses large magnitude. ■

The recovered $I(t)$ signal is given by:

$$\tilde{I}(t) = I(t) + \sum_{f_k \in \Omega} [\Delta Y_k \cos(f_k - f_c)t] \quad (7.8)$$

where Ω is the set of *dominant* luminance frequencies in the range of $[2.08, 5.08] \text{ MHz}$ for NTSC and $[2.93, 5.93] \text{ MHz}$ for PAL.

Similarly, the recovered $Q(t)$ signal is given as follows:

$$\tilde{Q}(t) = Q(t) + \sum_{f_k \in \Omega} [\Delta Y_k \sin(f_k - f_c)t] \quad (7.9)$$

Since the erroneous term appear to be in the shape of damped sinusoids, the contaminated chrominance oscillates within its color range and peaks at the opposite ends of the color pair: orange–cyan for $I(t)$ and green–purple for $Q(t)$. This explains the *rainbow-like* visual artifacts. If the adjacent scan lines are similar and the rainbow-effect repeats at about the same locations, the resultant visual artifacts take the form of *vertical color strip* as illustrated in Figure 7.1(a). Figure 7.1(b) shows the sinusoidal pattern of a chrominance component along a horizontal scan line.

Figure 7.1: Rainbow-effect artifacts: (a) vertical rainbow-like color strips, (b) the chrominance I values along the scan line marked in (a).

The frequency of the chrominance artifact f_a is given by:

$$f_a = |f_k - f_c|, f_k \in \Omega \quad (7.10)$$

From (7.8) and (7.9), we further deduce that the mean value of the chrominance artifacts over a local area is zero, as the erroneous term is sinusoidal.

The following property summarizes the characteristics of the chrominance artifact.

Property 2: The spatial-frequency of the chrominance artifact is related to that of the triggering luminance according to (7.10). The chrominance artifact changes periodically along the scan line, and has zero mean over a local area. ■

To verify the formulation, we applied these formulas on selected digital images with the aim to artificially introduce rainbow-effect artifact (the detailed simulation procedures are described in Appendix). We selected a white-strip pattern (possesses high luminance frequency and intensity, and zero chrominance) as shown in Figure 7.2(a) and artificially introduced rainbow-effect artifact. Rainbow-like color strips similar to the captured video frames are observed in the simulated result (Figure 7.2(b)). Figure 7.2(c) and (d) show the I chrominance values along a scan line. The original image has a zero chrominance, whereas the contaminated one exhibits a sinusoidal pattern.

Figure 7.2: Artificially introducing rainbow-effect artifact onto a digital image: (a) original image region, (b) simulated artifact region; (c) and (d) are the chrominance I values along the scan line marked in (a) and (b), respectively.

Figure 7.3 shows the frequency distribution (obtained via DCT) of the luminance Y and

chrominance I signals, along a scan line, in the rainbow-effect contamination region of a captured video frame and a simulated frame. For the captured video frame shown in Figure 7.3 (a), we observe that the luminance signal peaks at 2.84 MHz , which falls into the rainbow-effect triggering range of $[2.08, 5.08] \text{ MHz}$. Using (7.10), we derive the resultant chrominance-artifact frequency of 0.64 MHz . The observed chrominance-artifact frequency indeed peaks at 0.65 MHz . For the simulated frame shown in Figure 7.3 (b), the luminance signal peaks at 2.36 MHz and 2.23 MHz . The corresponding chrominance-artifact frequencies derived are 1.22 and 1.35 MHz respectively. The observed chrominance-artifact frequency peaks 1.3 MHz – a likely combination of the two above frequencies.

Figure 7.3: Frequency distribution of the luminance Y and chrominance I signals of (a) the captured video frame shown in Figure 7.1, and (b) the simulated frame shown in Figure 7.2.

Armed with a clear understanding of rainbow-effect artifacts, we can proceed to identify and isolate the artifact regions and devise means to remove the artifacts.

7.3 Detection of Artifact Regions

In this section, we present a novel method to identify and isolate rainbow-effect contaminated regions based on our formulation.

We start by segmenting the video frame into non-overlapping homogenous regions with Deng and Manjunath's unsupervised technique for segmentation [77] (see Chapter 6). Next, we exploit the properties summarized in Section 7.2 to classify the segmented regions into artifact (i.e., contaminated) and intact (i.e., uncontaminated) regions.

We know from Property 1 that the rainbow-effect artifact is triggered by dominant luminance in the high-frequency range of $[2.08, 5.08]$ MHz (for NTSC) and $[2.93, 5.93]$ MHz (for PAL). Hence, it is logical to exploit the strength of the luminance signal in this range for detecting rainbow-effect artifacts. We choose to perform frequency analysis using wavelet decomposition (instead of DCT or DFT) for its capability of multi-resolution analysis with localization in both spatial and frequency domains. As mentioned earlier, the signals of interests are confined to the scan lines. Thus, 1-D discrete wavelet transform (DWT) suffices. The 1-D DWT represents a signal s in terms of a set of shifted and dilated wavelet functions $\psi_{j,l}$ and scaling function $\phi_{N,l}$ [83]. When these shifted and dilated functions form an orthonormal basis, an arbitrary signal s can be decomposed as

$$s = \sum_l u_{N,l} \cdot \phi_{N,l} + \sum_{j,l} \omega_{j,l} \cdot \psi_{j,l} \quad (7.11)$$

where j denotes the scale; l denotes the position; N is the maximum number of the scales in the filter bank; $\omega_{j,l}$ denotes the wavelet coefficient; and $u_{N,l}$ denotes the scaling coefficient. When $N = j = 1$, the DWT is known as a one-scale decomposition, with the scaling coefficients representing the lower-half frequencies and the wavelet coefficients representing the higher-half. For example, assuming that the maximum frequency is F_m MHz, then

the resultant scale coefficients denote $0 - \frac{F_m}{2}$ MHz and the wavelet coefficients represent $\frac{F_m}{2} - F_m$ MHz. Successive decomposition on the scale coefficients produces subbands in $0 - \frac{F_m}{4}$ MHz and $\frac{F_m}{4} - \frac{F_m}{2}$ MHz.

Following the steps shown in Figure 7.4, we decompose the luminance signal using wavelet packet transform [84] to locate the frequency range of $[2.08, 5.08]^2$. The popular Daubechies wavelet (Daubechies-4 with mother wavelet filter $\psi = [0.4830, 0.8365, 0.2241, -0.1294]$) is used [83]. The following six frequency subbands of interest are obtained: $Y_1 \in [2.10, 2.63]$ MHz, $Y_2 \in [2.63, 3.15]$ MHz, $Y_3 \in [3.15, 3.68]$ MHz, $Y_4 \in [3.68, 4.20]$ MHz, $Y_5 \in [4.20, 4.73]$ MHz, and $Y_6 \in [4.73, 5.25]$ MHz. Figure 7.5 shows the luminance signal in these six subbands for the marked scan lines of the captured video frame shown in Figure 7.1 and the simulated video frame shown in Figure 7.2(b). Given that $Y(t)$ is supposed to be band-limited to 4.2 MHz in NTSC (and 5.5 MHz in PAL) before adding into the composite signal, the coefficient values in subbands above 4.2 MHz (i.e., Y_5 and Y_6) are small.

Figure 7.4: Decomposition structure of non-decimated wavelet packet transform for the NTSC luminance signal.

²The steps shown are meant for NTSC system, with upper-frequency bound of 2×4.2 MHz. PAL signal can be worked out similarly with upper-frequency bound of 2×5.5 MHz.

Figure 7.5: Luminance signal in the six subbands for the marked scan line of: (a) the captured video frame shown in Figure 7.1 and (b) the simulated video frame shown in Figure 7.2(b).

Next, we need to find a computational-efficient mean to compute the signal “strength” of the luminance subbands. Since the fine-scale wavelet subbands have zero mean, the “strength” can be represented by their variances. As the wavelet coefficients in frequency subbands above 4.2 MHz (Y_5 and Y_6) are generally small, we therefore collect the variances from subbands Y_1 , Y_2 , Y_3 , Y_4 and form a feature vector of 4 coefficients for each of the segmented region.

To perform classification, we could reduce the dimension of the feature vector space via principal component analysis (PCA) similarly as Section 6.2.1. For computational efficiency, we exploit the L2 norm of the feature vectors as the representative feature instead. Let $\mathbf{x}_i = [x_1, x_2, x_3, x_4]$, then the square of the L2 norm is given by $\|\mathbf{x}\|^2 = \sum_{j=1}^4 x_j^2$.

Figure 7.6 shows the distribution of the feature values for four test captured video frames. The feature values of the artifact regions are distinguishingly larger than those of the intact regions. Since the range of feature value varies among frames, the Fisher linear discrimination (FLD) technique is used to adaptively threshold the regions. Figure 7.6 shows the clas-

Figure 7.6: Distribution of feature values for four test video frames captured from composite video sources.

sification results on four test captured video frames, the artifact and intact regions are clearly separated. For low-complexity coarse classification, a fixed threshold of 75 percentile could be used, based on our empirical study.

Experimental results of the proposed detection method are shown in Figure 7.11 of Section 7.6. The results clearly show that our detection method is effective in identifying and isolating rainbow-effect contaminated regions.

7.4 Suppression of Rainbow-Effect Artifacts

Based on our analysis presented in Section 7.2, we know that the chrominance-artifact frequency falls in the range of $[0, 1.5] \text{ MHz}$. To address the artifacts in both spatial and frequency domains, we perform the artifact removal via wavelet transform. Figure 7.7 shows the wavelet packet decomposition steps for the chrominance signals and the three subbands of interest: $I_1, Q_1 \in [0, 0.53] \text{ MHz}$, $I_2, Q_2 \in [0.53, 1.05] \text{ MHz}$, $I_3, Q_3 \in [1.05, 1.58] \text{ MHz}$.

As an example, Figure 7.8 shows the chrominance signal in these subbands for the marked scan lines of the captured video frame shown in Figure 7.1 and the simulated video frame shown in Figure 7.2(b) (the corresponding luminance signal is shown in Figure 7.5). The artifact is apparent in the I_2 subband for Figure 7.8(a) and I_3 subband for Figure 7.8(b), which correspond to the triggering luminance in Y_2 and Y_1 subbands in Figure 7.5(a) and Figure 7.5(b), respectively, according to (7.10). Table 7.1 summarizes the relationship between triggering luminance subbands and the resultant chrominance-artifact subbands, based on (7.10).

Figure 7.7: Decomposition structure of non-decimated wavelet packet transform for the NTSC chrominance signals.

Next, we devise a mean to remove the artifact from the contaminated chrominance com-

Figure 7.8: Artifact chrominance signals in the three subbands for the marked scan line of: (a) the captured video frame shown in Figure 7.1, and (b) the simulated frame shown in Figure 7.2(b).

Table 7.1: Triggering luminance subband and its resultant chrominance-artifact subbands.

luminance	chrominance	
Y_1	I_3	Q_3
Y_2	I_2	Q_2
Y_3	I_1	Q_1
Y_4	I_1	Q_1

ponents, while maintaining the “original” chrominance contents. Based on Property 2, we understand that the chrominance artifact is sinusoidal, and thus possesses a zero-mean over a local area. In other words, the chrominance artifact does not affect the local mean. Hence, it is logical to replace the contaminated wavelet coefficients by its local mean, which would remove the artifact but maintain the “original” chrominance contents.

We further exploit the property that chrominance components possess diminutive high-frequency contents in our artifact removal process. In other words, the wavelet coefficients in the high-frequency chrominance subbands I_2, Q_2 and I_3, Q_3 are generally small and the energy is mainly concentrated in the low-frequency I_1, Q_1 subbands as illustrated in Figure 7.9. On the other hand, the erroneous terms corresponding to rainbow-effect artifacts

produce large wavelet coefficients in all chrominance subbands (illustrated in Figure 7.9 too).

Figure 7.9: Histograms of the wavelet coefficients for the original and the artifact chrominance in subbands: (a) I_1 , (b) I_2 , and (c) I_3 .

To remove the artifacts but preserve the “original” contents in the low-frequency chrominance subbands I_1 and Q_1 , we adopt the wavelet thresholding technique [81, 85]. We classify the wavelet coefficients into two categories: original and contaminated. Since the artifacts are caused by the residual luminance, luminance information could be served as a reference for classification. Assume that W_i^Y is a column vector consisting of wavelet coefficients of the part of a scan line within a contaminated region of the luminance subbands Y_i ($i = 1, 2, 3, 4$); and W_i^I and W_i^Q are the corresponding chrominance-artifact subbands, according to Table 7.1 (an example is shown in Figure 7.5 and Figure 7.8). In each of the W_i^Y local cycle, we compute the peak magnitude and its corresponding frequency value as follows: The points of intersection between W_i^Y and zero axis are marked as $\mathbf{a} = [a_1, a_2, a_3, \dots, a_m]$, where m is the total number of intersection points. The peak magnitudes are recorded as $\mathbf{q} = [q_1, q_2, q_3, \dots, q_m]$ occurring at $\mathbf{b} = [b_1, b_2, b_3, \dots, b_m]$. A peak at b_j is classified as a triggering point if its absolute magnitude is larger than a threshold, i.e., $|q_j| > T_i$ ($j = 1, 2, \dots, m$). In our implementation, T_i is set to the average of the absolute $q_1, q_2, q_3, \dots, q_m$. For each peak at b_j , its local frequency is calculated as $f_{b_j} = \frac{8.4}{a_{j+1} - a_j}$. Each luminance triggering point yields a sinusoid chrominance artifact. The cyclic range of the

resultant artifact signal can be predicted as $A = \frac{8.4}{f_a}$, where f_a is the predicted chrominance frequency according to (7.10) (by replacing f_y with a particular f_{b_j}). Given a triggering point at b_k , the chrominance coefficients in the local range of $[b_k - A, b_k + A]$ are evaluated against a threshold T_{ci} to decide its status: original or contaminated. In other words, supposing that W_b denotes the magnitude of chrominance coefficients at location b , where $b \in [b_k - A, b_k + A]$, it is classified as contaminated if $|W_b| > T_{ci}$. In our implementation, T_{ci} is set to the mean value of the chrominance magnitudes in the artifact region. For an identified contaminated chrominance wavelet coefficient W_b , the “original” wavelet coefficient is estimated by its local mean $\hat{W}_b$ computed using an averaging filter as follows.

$$\hat{W}_b = \frac{1}{2 * A} \sum_{i=b_k-A}^{b_k+A} W_i. \quad (7.12)$$

For the high-frequency chrominance subbands I_2, Q_2, I_3 , and Q_3 , as the “original” coefficient values are minimal, we simply set all the wavelet coefficients to their local mean value of the entire artifact region, without destroying the visual quality.

In summary, our proposed rainbow-effect detection and removal algorithm consists of the following steps:

1. Segmentation

Use Deng and Manjunath’s unsupervised segmentation method [77] to segment the video frame into non-overlapping homogenous regions.

2. Artifact-Region Detection

For each region, construct a 4-D feature vector comprising the variances of the four luminance wavelet subbands Y_1, Y_2, Y_3 and Y_4 , and compute the L2 norm of the feature vector which serves as the discriminating feature value. The feature value of each

region is thresholded to decide the region class. The threshold is set to one that maximizes the Fisher discriminant.

3. Artifact Removal

Restore the wavelet coefficients of chrominance in $[0, 0.53]$ MHz subband by wavelet thresholding. The triggering points in luminance are first determined by evaluating their magnitudes. Within the predicted range around the identified triggering points, the chrominance coefficients are evaluated by magnitude thresholding. Those classified as contaminated are restored with their estimated local mean. The wavelet coefficients in $[0.53, 1.05]$ MHz and $[1.05, 1.58]$ MHz subbands are replaced by the local mean value of the entire artifact region.

7.5 Removal of Rainbow-Effect Artifacts in Video Sequences

The frame-based detection-removal algorithm proposed in the previous section can be extended to video sequences by applying it to each of the video frames independently (henceforth referred to as the *intra-frame* restoration method). Nonetheless, the restoration performance could be improved by exploiting the inter-frame correlations. In a video sequence, the image contents of the neighboring frames change little, thus, the luminance components are similar. However, the chroma switches sign from one frame to the next frame [18] in composite video transmission. For instance, suppose that the composite video signal for the current video frame is given as in (7.4). Then, in the preceding (or the successive) frame, the signal is given by:

$$S(t) = Y(t) - I\cos(f_c t) - Q\sin(f_c t) \quad (7.13)$$

Following the analysis described in Section 7.2, the recovered chrominance components I and Q are given by:

$$\tilde{I}(t) = I(t) - \sum_{k \in \Omega} [\Delta Y_k \cos(f_k - f_c)t] \quad (7.14)$$

$$\tilde{Q}(t) = Q(t) - \sum_{k \in \Omega} [\Delta Y_k \sin(f_k - f_c)t] \quad (7.15)$$

The above expressions show that the chrominance artifacts at the same spatial location in two neighboring frames should have similar values but inverse in sign. Based on this property, GuavaComb filter [17] (as mentioned in Section 6.1) removes the artifacts by averaging the chroma of two neighboring frames (henceforth referred to as the *chroma-averaging* method). Chroma-averaging method works well on *static* scenes but fails in scenes with large motion. Figure 7.14(d) shows the result of averaging the chroma of two captured video frames in a dynamic scene contaminated by rainbow-effect artifact. It can be seen that chroma-averaging method fails to completely remove the artifacts due to the presence of large motion.

We notice that the chrominance correlation between neighboring frames can be used to distinguish artifact regions from intact regions. Table 7.2 compares the average inter-frame correlation in the chrominance components (I and Q), and their wavelet subbands (I_1, I_2, I_3, Q_1, Q_2 , and Q_3) in the artifact and the intact regions. It can be seen that the correlation is much lower in the artifact regions due to the erroneous changes in chrominance. We therefore propose to predict the artifact regions in the successive frame based on the chrominance correlation.

However, due to the misalignment between the regions from two neighboring frames, which results in a decrease in correlation, some intact regions could be mistakenly predicted as artifact regions. The alignment can be improved by estimating the motion vector between

Table 7.2: Average inter-frame correlations of chrominance components and their subbands in the artifact and intact regions.

Inter-frame Correlation	Artifact Region				Intact Region			
	I	I_1	I_2	I_3	Q	Q_1	Q_2	Q_3
k & k+1	0.04	0.36	0.17	0.43	0.94	0.95	0.74	0.47
k & k+2	0.29	0.55	0.24	0.17	0.91	0.93	0.64	0.35

the two frames. The kinds of motion present in video sequences include global translation and global rotation due to camera motion, camera zoom, object translation, object rotation, *etc.* [86]. For simplicity, we shall consider only the global translation. Let F be the luminance of the current frame; and F_k be the luminance of its neighboring frame related to F by a global translation (v_1, v_2) . We adopt a simple image registration approach [87] to estimate (v_1, v_2) as follows:

$$(\hat{v}_1, \hat{v}_2) = \underset{i,j}{\operatorname{argmin}} E\{[F_k(m+i, n+j) - F(m, n)]^2\} \quad (7.16)$$

and align the frame F_k to F by

$$F'(m, n) = F_k(m + \hat{v}_1, n + \hat{v}_2) \quad (7.17)$$

If the frames match well, i.e., having a small mean square error $E\{[F_k(m+i, n+j) - F(m, n)]^2\}$, the frames are considered as containing similar contents. The correlation parameters of each aligned region pair are then calculated. The regions with average correlation value of less than 0.5 are classified as artifact regions.

After identifying the artifact regions, we propose an inter-frame restoration method by combining our intra-frame technique and chroma-averaging method. It operates by applying the intra-frame restoration method within the detected artifact regions, and chroma-averaging in the rest of the regions. We observed that although the chroma-averaging restoration method is not effective when large motion exists, it can remove small and random artifacts,

which are not integrated into a region.

7.6 Experimental Results

In this section, we present the results of our rainbow-effect detection and removal algorithm, and compare them with the existing methods. The benchmark schemes for comparison include GuavaComb filter [17] and Smart Smoother IQ (SMIQ) filter³ [76]. We also include the restoration results using 2D Wiener filter⁴ – a traditional denoising technique, for comparison.

We shall first present the single-frame (or intra-frame) detection and restoration results, and then proceed to present results for inter-frame restoration for video sequences. Since the GuavaComb filter requires neighboring frames for processing, its results will be presented later.

Figure 7.10: Test video frames. (a) captured video frames. (b) simulated video frames.

Figure 7.10 shows several rainbow-effect contaminated video frames. In Figure 7.10 (a),

³For SMIQ filter, we use the default optimal parameters, i.e., diameter=11 and strength=200.

⁴We use the MATLAB function 'wiener2'.

Figure 7.11: Detected artifact masks of the test video frames in Figure 7.10.

the video frames are captured (digitized) from composite video sources. In Figure 7.10 (b), the images are generated by artificially introducing rainbow effect on selected digital images. Figure 7.11 shows the detected artifact regions using our proposed detection method. Clearly, our proposed detection method is effective in identifying and isolating artifact regions in captured as well as simulated frames.

Figure 7.12 compares the restored images using (a) 2D Wiener filter, (b) SMIQ filter and (c) our proposed detection-restoration method for the captured video frames shown in Figure 7.10(a). Visual inspection reveals that our method can restore more contamination pixels than the other methods, and it does not introduce degradation (such as blur edges) in the uncontaminated areas.

To evaluate the restoration performance *quantitatively*, we run the restoration on simulated video frames shown in Figure 7.10(b). We then compute the peak-signal-to-noise ratio (PSNR) of the restored frame against the original digital image. Table 7.3 compares the PSNR values (in dB) of the degraded simulated frames (caused by the introduction of rainbow-effect artifact), and the corresponding restored frames using 2D Wiener filter, SMIQ filter and our proposed method. All the three restoration methods show an improvement in

Figure 7.12: Restoration results on captured video frames using: (a) 2D Wiener filter, (b) SMIQ filter, and (c) our proposed method.

Figure 7.13: Restoration results on simulated video frames. (a) original frames, (b)-(d) restored frames using 2D Wiener filter, SMIQ filter, and our proposed method, respectively.

PSNR, compared with the degraded simulated frames. Nonetheless, our method clearly outperforms the others. The restoration results are shown in Figure 7.13.

Table 7.3: Comparison of PSNR performance.

Image		Whole Image			Artifact Region		
		R	G	B	R	G	B
Window	Degraded	33.50	38.25	30.45	28.05	32.58	24.82
	2D Wiener	33.77	38.54	30.91	28.45	32.91	25.35
	SMIQ	33.04	38.46	31.65	28.41	34.52	27.41
	Proposed	34.13	41.95	33.11	29.13	37.62	28.24
Stripe	Degraded	32.71	35.43	26.21	27.93	32.12	22.89
	2D Wiener	33.71	36.25	27.07	29.16	33.54	24.26
	SMIQ	37.44	38.65	30.94	34.66	37.23	31.44
	Proposed	38.61	38.93	30.08	38.80	43.11	33.80
Barbara	Degraded	33.08	37.32	29.69	31.31	34.98	26.45
	2D Wiener	34.07	38.29	30.80	32.05	35.52	27.28
	SMIQ	31.87	38.93	32.90	32.93	38.79	32.86
	Proposed	34.51	38.87	31.80	35.97	39.95	33.85
Starburst	Degraded	26.96	29.25	23.11	21.83	24.30	18.28
	2D Wiener	28.28	29.90	25.17	22.25	24.53	18.82
	SMIQ	29.86	30.76	27.66	24.09	25.81	20.97
	Proposed	30.24	31.16	28.25	27.19	27.78	25.55

Figure 7.14 compares the restoration results for captured video sequences of three methods: chroma-averaging GuavaComb filter, and our proposed intra-frame and inter-frame restoration methods. It can be seen that our proposed inter-frame method removes the rainbow-effect artifacts more effectively than chroma-averaging and intra-frame method.

7.7 Conclusion

In this chapter, we propose a novel method to restore rainbow-effect contaminated videos. We begin by deriving the formulas governing rainbow-effect artifact from the first principle. Based on the formulation, we exploit features in the residual luminance and the chrominance artifact to detect the rainbow-effect contaminated region. Once the artifact regions are isolated, we remove the rainbow-effect artifacts by modifying large wavelet coefficients in chrominance subbands. We then extend our intra-frame restoration method to inter-frame, for restoring video sequences, by exploiting the inter-frame chrominance correlation. Exper-

Figure 7.14: Removing rainbow-effect artifacts in video sequences. (a)-(b) Two successive captured video frames (the artifact region is highlighted with circle). (c) Result of inter-frame detection. Restoration results of (d) chroma-averaging, (e) proposed intra-frame removal, and (f) proposed inter-frame removal method.

iments conducted on captured video sequences from composite video sources demonstrate that our method can effectively remove rainbow-effect artifacts. Results on simulated video frames show that our method can improve the image quality in terms of PSNR.

Comparing with the proposed method in Chapter 6, this method has two advantages: 1) it can be customized to be more computational efficient because the exploited artifact detection are based on line signal information instead of the whole artifact region, 2) the artifact removal is more accurate with identifying the target frequency range.

Appendix: Simulating Rainbow Effect on Digital Images

To quantitatively evaluate the performance of the various rainbow-effect artifact removal techniques, we need the original (uncontaminated) images as the basis for calculating PSNR. However, in most cases (such as captured videos from composite video sources) these images are not available. An alternative solution is to introduce rainbow-effect artifacts on digital images via simulation. In this section, we shall describe our simulating procedures for NTSC system [18, 74].

1. **Color Transformation** Transform the image from RGB color space into YIQ color space, , using (7.1) and (7.2). The image width is normalized to 452, which is the minimum pixel numbers per line [16, 18]. In the following discussion, the term signal refers to a single horizontal scan line.
2. **Band-limiting the NTSC Primaries** The Y , I , and Q signals are band-limited before modulation. According to [16], the sampling rate of the NTSC primaries with 452 samples per line is 8.4 MHz . The maximum signal frequency is half of the sampling rate, i.e., 4.2 MHz . We band-limit the signals according

to [18]. The Y , I , Q signals are first interpolated to twice of the length, i.e., 904 pixels per line. Y is filtered to a 4.2 MHz cutoff using a 9-pixel-wide filter $h_y = (0.0295, -0.0524, -0.118, 0.472, 1, 0.472, -0.118, -0.0524, 0.0295)$. I and Q are filtered to a 1.5 MHz cutoff using a 29-pixel-wide filter $h_c = (0.00474, 0.01739, 0.0295, 0.02743, 0, -0.0524, -0.1123, -0.1467, -0.118, 0, 0.2074, 0.4720, 0.73489, 0.92867, 1, \dots)$, where h_c is symmetrical, and we only show half of the filter coefficients. These two filters are adopted from [18]. After band-limiting, the signals are down-sampled to the original size of 452 samples per line.

3. **Modulation** After band-limiting, the Y , I , and Q signals are modulated and combined into one single composite signal. Assuming that the picture carrier frequency $f_y = 20\text{ MHz}$, the chrominance carrier frequency is then $f_y + f_c = 23.58\text{ MHz}$. The signals are interpolated to meet the maximum sampling rate of $2 \times (f_y + f_c)$. Since the original sampling rate of the image is 8.4 MHz, the signals need to be interpolated for $\frac{2 \times 23.58}{8.4}$ times (set as 6 in our implementation). (7.13) becomes:

$$S(j) = Y'(j)\cos\left(\frac{20M}{23.58M} * \frac{\pi}{2} * j\right) + I'(j)\cos\left(\frac{\pi}{2} * j\right) + Q'(j)\sin\left(\frac{\pi}{2} * j\right) \quad (7.18)$$

where j is the pixel position; Y' , I' , and Q' are the interpolated versions of Y , I , and Q .

4. **Demodulation** To extract the chrominance component I , the composite video signal in (7.18) is multiplied with $2\cos(\frac{\pi}{2} * j)$. We apply low-pass filtering directly on the resultant signal instead of the separated out chroma C signal as in (7.5). Thus, (7.6)

becomes:

$$\begin{aligned}
 2S(j)\cos\left(\frac{\pi}{2} * j\right) &= I'(j) + I'(j)\cos(\pi * j) + Q'(j)\sin(\pi * j) \\
 &+ Y'(j)\cos\left(\left(\frac{20M}{23.58M} + 1\right) * \frac{\pi}{2} * j\right) \\
 &+ Y'(j)\cos\left(\left(\frac{20M}{23.58M} - 1\right) * \frac{\pi}{2} * j\right) \quad (7.19)
 \end{aligned}$$

A low-pass filter is applied to recover the I' signal. The low-pass filter is the interpolated version of h_c , and the interpolation parameter is set as 6 in our implementation. Q' can be similarly extracted.

5. **Reconstruction** The recovered $\tilde{I}'$ and $\tilde{Q}'$ are down-sampled to the original size of 452 pixels per line. Finally, the down-sampled $\tilde{I}$ and $\tilde{Q}$ signals are combined with the original luminance component Y to form the YIQ triplet, which is then converted back to RGB color space.

Chapter 8

Conclusion and Future Work

This thesis addresses error analysis and suppression in three applications – error suppression in CFA demosaicking, error analysis in DSC processing chain, and error detection and removal for composite video frames.

We first studied the error suppression techniques for CFA demosaicking. Our examination shows that demosaicking artifacts are mainly caused by isotropic interpolation or inconsistency between color components. To address this issue, we have proposed a method with weighted interpolation in color-difference components to adaptively estimate the missing high frequencies across color components. Moreover, we have enhanced an existing method using better edge-adaptive interpolation and selective high-frequency estimation across color components. Subsequently, we have investigated the CFA demosaicking problem from the view of Fourier domain analysis. From this perspective, CFA demosaicking is equivalent to recovering the original image spectrum, and the demosaicking artifacts are mainly caused by improper separation between the aliased image spectra. By analyzing the spectrum of CFA image using sampling theory, we have observed that the luminance component preserves more high-frequency information than other color components due to the smaller

bandwidth of modulated color-difference planes. Furthermore, we have observed that extra high-frequency information is preserved in the green pixel locations. Based on these observations, we have designed a demosaicking method which first extracts the luminance component by using adaptive filtering techniques and then reconstructs the color components based on the high correlation between color components. Experimental results confirm that the proposed methods outperform the existing methods both visually and in terms of PSNR.

One possible future research direction of our work on CFA demosaicking is to apply our observations and proposed techniques to reconstructing CFA video sequence. Conceivably, the temporal correlation of the color signal provides a wealth of information, which could be exploited in the recovery process to enhance the video quality. Another possible application is super-resolution, which reconstructs one or a set of high-resolution images from a sequence of low-resolution ones. Although it may not be straightforward to apply the CFA demosaicking algorithms to super-resolution reconstruction [88], the schemes of exploiting color image correlations and extracting component spectrum could be explored for the purpose of constructing super-resolution.

In the second part of this thesis, we have analyzed the error suppression in DSC processing chain. A statistical model has been proposed to analyze the error interaction in the DSC image processing chain. This model helps us to better understand the interaction between the key processes, and provides clues of better cooperative solutions. In particular, we analyzed two specific cases in DSC, one involving demosaicking and compression and the other comprising demosaicking and sharpening. For the problem of demosaicking and compression, the models developed not only confirm the empirical results reported in literature, but also accurately predict the break-point compression ratio, below which an alternative processing chain can outperform the conventional one. For the problem of demosaicking and sharpening, we have obtained two design guidelines from the error model analysis: 1)

avoid enlarging errors when extracting the high-frequency information for sharpening, and 2) adding the high-frequency component after demosaicking process to avoid degrading demosaicking performance. We have proposed two solutions, based on these guidelines, and the experimental results show that our proposed algorithms can outperform existing methods.

One possible future research direction on this topic is to design an efficient compression technique customized for CFA samples. As compression is even more important for video due to large amount of data, one can extend the study on image compression to video compression with the inclusion of video demosaicking. Another possible research is apply the error analysis to other visual signal processing chains.

Finally, we have investigated the error detection and removal in digitized composite videos. The main focus is on rainbow-effect artifacts, which are caused by residual high-frequency luminance information in the extracted chrominance components after demodulation. Two important characteristics are observed: 1) the artifacts usually occur in areas of high luminance frequencies, and 2) the resultant chrominance artifacts possess approximately symmetric histogram. Based on the observed characteristics, we have proposed a scheme to detect the artifact regions and then remove the artifacts effectively. Furthermore, we have investigated the source of rainbow-effect artifacts and derived the governing formulas accordingly. Based on the formulation, we exploit high-frequency energy of luminance in specific frequency range to detect the regions affected by rainbow-effect artifacts. Once the artifact regions are isolated, we remove the artifacts by modifying large wavelet coefficients in chrominance subbands. We have then extended our intra-frame restoration method to restoring video sequences by exploiting the inter-frame chrominance correlation. Experiments conducted on captured video sequences from composite video sources demonstrate that our method can effectively remove rainbow-effect artifacts. Results on simulated video frames show that our method can also improve the image quality in terms of the PSNR.

As for the future work on error detection and removal in digitized composite video frames, one can aim to develop more effective techniques by incorporating the inter-frame correlation to restore other similar artifact, e.g., color bleeding artifacts from composite video sequences.

Author's Publications

Journal Papers:

1. Lanlan Chang and Yap-Peng Tan, "Effective Use of Spatial and Spectral Correlations for Color Filter Array Demosaicking", *IEEE Transactions on Consumer Electronics*, vol. 50, pp. 355-365, Feb. 2004.
2. Lanlan Chang and Yap-Peng Tan, "Hybrid Color Filter Array Demosaicking for Effective Artifact Suppression", *Journal of Electronic Imaging*, vol. 15, Mar. 2006.
3. Nai-Xiang Lian, Lanlan Chang, Vitali Zagorodnov and Yap-Peng Tan, "Reversing Demosaicking and Compression in Color Filter Array Image Processing: Performance Analysis and Modeling", *IEEE Transactions on Image Processing*, vol. 15, pp. 3261-3278, Nov. 2006.
4. Nai-Xiang Lian, Lanlan Chang, Vitali Zagorodnov and Yap-Peng Tan, "Adaptive Filtering for Color Filter Array Demosaicking", *Under Press in IEEE Transactions on Image Processing*, 2007.
5. Lanlan Chang and Yap-Peng Tan, "Color Filter Array demosaicking in Real Digital Camera", *Under Preparation*.
6. Lanlan Chang and Yap-Peng Tan, "Analysis, Detection and Removal of Rainbow Effect", *Under Preparation*.
7. Lanlan Chang and Yap-Peng Tan, "Rainbow Effect Detection and Removal", *Under Preparation*.

Conference Papers:

1. Lanlan Chang and Yap-Peng Tan, "Adaptive Color Filter Array Demosaicking with Artifact Suppression ", *IEEE International Symposium on Circuits and Systems (IS-CAS 2004)*, May 23-26, 2004, Vancouver, Canada.
2. Lanlan Chang and Yap-Peng Tan, "Combined Use of Spatial and Spectral Correlations for Enhanced Color Filter Array Demosaicking", *IEEE International Conference on Multimedia & Expo (ICME 2004)*, June 27-30, 2004, Taipei, Taiwan.
3. Nai-Xiang Lian, Lanlan Chang and Yap-Peng Tan, "Improved Color Filter Array Demosaicking by Accurate Luminance Estimation", *IEEE International Conference on Image Processing (ICIP2005)*, Sept. 11-14, 2005, Genova, Italy.
4. Haoyang Li, Lanlan Chang and Yap-Peng Tan, "Performance Factors for Evaluating Color Filter Array Demosaicking Algorithms", *IEEE International Conference on Information, Communications and Signal Processing (ICICS 2005)*, Dec. 6-9, 2005, Bangkok, Thailand.
5. Nai-Xiang Lian, Lanlan Chang, Vitali Zagorodnov and Yap-Peng Tan, "Reversing Demosaicking and Compression of Color Filter Array Images", *IEEE International Conference on Image Processing (ICIP2006)*, Oct. 8-11, 2006, Atlanta, GA, USA.
6. Lanlan Chang, Yap-Peng Tan, and Hock-Chuan Chua "Detection and Removal of Aliasing Artifacts", *IEEE International Conference on Image Processing (ICIP2007)*, Sept. 16-19, 2007, San Antonio, Texas, USA.

Bibliography

- [1] Y. T. Tsai, "Color image compression for single-chip cameras," *IEEE Transaction on Electron Devices*, vol. 38, no. 5, 1991.
- [2] T. Toi and M. Ohita, "A subband coding technique for image processing in single CCD cameras with Bayer color filter arrays," *IEEE Transactions on Consumer Electronics*, vol. 45, no. 1, pp. 176–180, 1999.
- [3] N. Zhang and X. Wu, "Lossless compression of color mosaic images," *IEEE International Conference on Image Processing*, vol. 1, pp. 517 – 520, 2004.
- [4] S.-Y. Lee and A. Ortega, "A novel approach of image compression in digital cameras with a Bayer color filter," *IEEE International Conference on Image Processing*, vol. 3, pp. 482–485, 2001.
- [5] B. Parrein, M. Tarin, and P. Horain, "Demosaicking and JPEG2000 compression of microscopy images," *IEEE International Conference on Image Processing*, vol. 1, pp. 521 – 524, 2004.
- [6] C. C. Koh, J. Mukherjee, and S. K. Mitra, "New efficient methods of image compression in digital cameras with color filter array," *IEEE Transactions on Consumer Electronics*, vol. 49, p. 4, 2003.
- [7] 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.
- [8] 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, 2005.
- [9] MATLAB Version 6.5, *Image Processing Toolbox User's Guide*. The MathWorks, Inc., 2002.
- [10] J. S. Lim, *Two-Dimensional Signal and Image Processing*. Prentice-Hall PTR, 1989.
- [11] G. C. Holst, *CCD Arrays Cameras and Displays*. JCD Publishing and SPIE - The international Society for Optical Engineering, 1998.

- [12] V. Bhaskaran and K. Konstantinides, *Image and Video Compression Standards*. Kluwer Academic Publishers, 1995.
- [13] D. S. Taubman and M. W. Marcellin, "JPEG2000: Standard for interactive imaging," *Processings of the IEEE*, vol. 90, no. 8, pp. 1336–1357, 2002.
- [14] D. Taubman, "High performance scalable image compression with EBCOT," *IEEE Transactions on image Processing*, vol. 9, no. 7, pp. 1158–1170, 1997.
- [15] Y. Wang, J. Ostermann, and Y.-Q. Zhang, *Video Processing and Communications*. Prentice Hall, Inc., 2002.
- [16] A. M. Tekalp, *Digital Video Processing*. Prentice Hall PTR, Prentice Hall. Inc., 1995.
- [17] Wilbert, "Removing rainbow effects and dot crawls," <http://big5.doom9.org/capture/rainbows.htm>.
- [18] J. F. Blinn, "NTSC: nice technology, super color," *IEEE Computer Graphics & Applications*, vol. 13, no. 2, pp. 17–23, 1993.
- [19] B. K. Gunturk, Y. Altunbasak, and R. M. Mersereau, "Color plane interpolation using alternating projections," *IEEE Transactions on Image Processing*, vol. 11, no. 9, 2002.
- [20] B. Bayer, "Color imaging array," *United States Patent*, no. 3,971,065, 1976.
- [21] D. Cok, "Signal processing method and apparatus for sampled image signals," *United States Patent*, no. 4,630,307, 1986.
- [22] 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.
- [23] H. Hibbard, "Apparatus and method for adaptively interpolating a full color image utilizing luminance gradients," *United States Patent*, no. 5,382,976, 1996.
- [24] T. W. Freeman, "Median filter for reconstructing missing color samples," *United States Patent*, no. 4,724,395, 1988.
- [25] 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.
- [26] 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.

- [27] J. Adams, K. Parulski, and K. Spaulding, "Color processing in digital cameras," *Proc. of IEEE*, vol. 18, no. 6, pp. 20–30, 1998.
- [28] J. Hamilton and J. Adams, "Adaptive color plane interpolation in single sensor color electronic camera," *United States Patent*, no. 5,629,734, 1997.
- [29] R. Kimmel, "Demosaicing: image reconstruction from color CCD samples," *IEEE Transactions Image Processing*, vol. 7, no. 3, pp. 1221–1228, 1999.
- [30] 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.
- [31] X. Li and M. T. Orchard, "New edge-directed interpolation," *IEEE Transactions on Image Processing*, vol. 10, no. 10, 2001.
- [32] S. C. Pei and I. K. Tam, "Effective color interpolation in ccd color filter arrays using signal correlation," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no. 6, pp. 503 – 513, 2003.
- [33] R. Rajeev, E. S. Wesley, L. B. Griff, and S. William, "Demosaicking methods for Bayer color arrays," *Journal of Electronic Imaging*, vol. 11, no. 3, pp. 306–315, 2002.
- [34] B. K. Gunturk, J. 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, 2005.
- [35] N.-X. Lian, V. Zagorodnov, and Y. P. Tan, "Edge-preserving image denoising via optimal color space projection," *IEEE Transactions on Image Processing*, vol. 15, no. 9, pp. 2575–2587, 2006.
- [36] F. Dretske, *Seeing and Knowing*. Chicago: University of Chicago Press, 1969.
- [37] R. Sekuler and R. Blake, *Perception*. McGraw Hill, 4th ed., 2002.
- [38] X. Wu and N. Zhang, "Primary-consistent soft-decision color demosaicking for digital cameras," *IEEE Transactions on Image Processing*, vol. 13, pp. 1263–1274, 2004.
- [39] K. Hirakawa and T. W. Parks, "Adaptive homogeneity-directed demosaicing algorithm," *IEEE Transactions on Image Processing*, vol. 14, no. 3, pp. 360–369, 2005.
- [40] 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, 2001.

- [41] X. Li, "Demosaicing by successive approximation," *IEEE Transactions on Image Processing*, vol. 14, no. 3, pp. 370–379, 2005.
- [42] L. Chang and Y.-P. Tan, "Hybrid color filter array demosaicking for effective artifact suppression," *Journal of Electronic Imaging*, vol. 15, no. 1, 2006.
- [43] 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.
- [44] K.-H. Chung and Y.-H. Chan, "Color demosaicing using variance of color differences," *IEEE Transactions on Image Processing*, vol. 15, no. 10, pp. 2944–2955, 2006.
- [45] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*. Pearson Education. Inc., 2002.
- [46] F. W. Campbell and J. Kulikowski, "Orientational selectivity of the human visual system," *Journal of Physiology*, vol. 187, no. 2, pp. 437–445, 1966.
- [47] 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, 2003.
- [48] M. Vetterli and J. Kovacevic, *Wavelets and Subband Coding*. Prentice-Hall PTR, 1995.
- [49] Adobe Photoshop CS2 Camera Raw Plug-In. [Http://www.adobe.com/products/photoshop/cameraraw.html](http://www.adobe.com/products/photoshop/cameraraw.html).
- [50] J. Canny, "A computational approach to edge detection," *IEEE Transactions on PAMI*, vol. 8, no. 6, 1986.
- [51] 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.Demosaicing>.
- [52] R. Lukac and K. N. Plataniotis, "Fast video demosaicking solution for mobile phone imaging application," *IEEE Transaction on Consumer Electronics*, vol. 51, no. 2, pp. 675–681, 2005.
- [53] 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, 2005.
- [54] 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.
- [55] M. Mancuso and S. Battiato, "An introduction to the digital still camera technology," *ST Journal of System Research*, vol. 2, 2001.
- [56] L. Chang and Y.-P. Tan, "Effective use of spatial and spectral correlations for color filter array demosaicking," *IEEE Transactions on Consumer Electronics*, vol. ASSP-50, pp. 355–365, 2004.

- [57] 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.
- [58] 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.
- [59] 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, 1999.
- [60] 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, 1999.
- [61] 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, 2000.
- [62] 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, 2001.
- [63] N. S. Jayant and P. Noll, *Digital Coding of Waveforms*. Prentice-Hall Press, 1984.
- [64] R. M. Gray and D. L. Neuhoff, "Quantization," *IEEE Transaction on Information Theory*, vol. 44, no. 6, pp. 2325–2382, 1998.
- [65] B. W. Keelan, *Handbook of Image Quality: Characterization and Prediction*. CRC Press, 2002.
- [66] G. Ramponi and A. Polesel, "Rational unsharp masking technique," *Journal of Electronic Imaging*, vol. 7, no. 2, 1998.
- [67] A. Polesel and G. Ramponi, "Image enhancement via adaptive unsharp masking," *IEEE Transactions on Image Processing*, vol. 9, no. 3, pp. 505–510, 2000.
- [68] S. H. Kim and J. P. Allebach, "Optimal unsharp mask for image sharpening and noise removal," *Journal of Electronic Imaging*, vol. 14, no. 2, 2005.
- [69] T. Komatsu and T. Saito, "Demosaicking for a color image sensor with removal of blur due to an optical low-pass filter," *Proc. SPIE*, vol. 5301, pp. 334–345, 2004.
- [70] T. Komatsu and T. Saito, "Super-resolution sharpening-demosaicking method for removing image blurs caused by an optical low-pass filter," *IEEE International Conference on Image Processing*, vol. 1, pp. 845–848, 2005.

- [71] R. Lukac and K. N. Plataniotis, "Cost-effective sharpening of single-sensor camera images," *IEEE International Conference on Multimedia & Expo (ICME)*, 2006.
- [72] P. E. Haralabidis and C. Pilinis, "Linear color camera model for a skylight colorimeter with emphasis on the imaging pipeline noise performance," *Journal of Electronic Imaging*, vol. 14, no. 4, 2005.
- [73] N.-X. Lian, L. Chang, Y.-P. Tan, and V. Zagorodnov, "Adaptive filtering for color filter array demosaicking," *IEEE Transactions on Image Processing*.
- [74] Rec. ITU-R BT.1700, "Characteristics of composite video signals for conventional analogue television systems,"
- [75] Y. C. Faroudja, "NTSC and beyond," *IEEE Transactions on Consumer Electronics*, vol. 34, no. 1, pp. 166–178, 1988.
- [76] "Smart smoother IQ filter for VirtualDub," <http://www.doki.ca/filters/>.
- [77] Y. Deng and B. S. Manjunath, "Unsupervised segmentation of color - texture regions in images and video," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 8, 2001.
- [78] B. Windrow and S. Stearns, *Adaptive Signal Processing*. 1990.
- [79] K. Fukunaga, *Introduction to Statistical Pattern Recognition*. Elsevier, 1990.
- [80] R. Fisher, *The Use of Multiple Measure in Taxonomic Problems*. Annals of Eugenics, 1936.
- [81] H. C. J. Romberg and R. G. Baraniuk, "Bayesian tree-structured image modeling using wavelet-domain hidden markov models," *IEEE Transactions on Signal Processing*, vol. 10, no. 7, pp. 1056–1068, 2001.
- [82] M. S. Crouse, R. D. Nowak, and R. G. Baraniuk, "Wavelet-based statistical signal processing using hidden markov models," *IEEE Transactions on Signal Processing*, vol. 46, no. 4, pp. 886–902, 1998.
- [83] I. Daubechies, *Ten Lectures on Wavelets*. New York: SIAM, 1992.
- [84] A. C. Cristn and A. T. Walden, "Multitaper power spectrum estimation and thresholding: wavelet packets versus wavelets," *IEEE Transactions on Signal Processing*, vol. 50, pp. 2976–2986, 2002.
- [85] E. J. Balster, Y. F. Zheng, and R. L. Ewing, "Feature-based wavelet shrinkage algorithm for image denoising," *IEEE Transactions on Image Processing*, vol. 14, no. 12, pp. 2024–2039, 2005.

- [86] C. Stiller and J. Konrad, "Estimating motion in image sequences," *IEEE Signal Processing Magazine*, vol. 16, no. 4, pp. 70–91, 1999.
- [87] B. S. Reddy and B. N. Chatterji, "An fft-based technique for translation, rotation, and scale-invariance image registration," *IEEE Transactions on Image Processing*, vol. 5, no. 8, pp. 1266–1271, 1996.
- [88] S. Farsiu, D. Robinson, M. Elad, and P. Milanfar, "Advances and challenges in super-resolution," *International Journal of Imaging Systems and Technology*, vol. 14, no. 2, pp. 47–57, 2004.